Міністерство освіти і науки України
Міжнародний гуманітарний університет
Кафедра комп’ютерних наук

Русу О.П.

ШТУЧНІ НЕЙРОННІ МЕРЕЖІ

Методичні вказівки до практичних робіт
для здобувачів вищої освіти,
які навчаються за спеціальностями
галузі «Інформаційні технології»

Одеса 2025

Затверджено Вченою Радою Міжнародного гуманітарного університету.
Протокол № 5 від 20 січня 2025 р.

Русу О.П. Штучні нейронні мережі. Методичні вказівки до практичних робіт для здобувачів вищої освіти, які навчаються за спеціальностями галузі «Інформаційні технології» [Електронне видання]. Кафедра комп'ютерних наук Міжнародного гуманітарного університету. Одеса, 2025. 84 с.

Навчальна дисципліна «Штучні нейронні мережі» присвячена вивченню існуючих та перспективних штучних нейронних мереж. При вивченні дисципліни особливу увагу приділяється методам програмної реалізації штучних нейронних мереж, зокрема алгоритмам та математичному апарату, який використовується для їх реалізації. Завдання вивчення дисципліни полягає в отриманні навичок аналізу, вибору та реалізації штучних нейронних мереж з метою впровадження їх в програмному забезпеченні.

ЗМІСТ

Практична робота 1. Дослідження штучного нейрону

5.1 Теоретичні положення

Штучний нейрон (Artificial neuron, (Математичний нейрон, Нейрон Маккалоха-Піттса, Формальний нейрон) – вузол штучної нейронної мережі, що є спрощеною моделлю природного нейрона. Математично, штучний нейрон зазвичай представляють як деяку нелінійну функцію Y від лінійної комбінації всіх вхідних сигналів X_1, $X_2 \ldots X_n$ (Рисунок 5.1).

Рисунок 5.1 – Штучний нейрон

Штучний нейрон є математичною моделлю біологічного нейрона мозку. Зазвичай його зображують у вигляді кружечка із лініями, що позначають входи та вихід. Штучні нейрони об'єднують в нейронні мережі, при цьому входи штучних нейронів можуть підключатися як до входів системи (по первинних датчиків), так і до виходів інших нейронів (Рисунок 5.2).

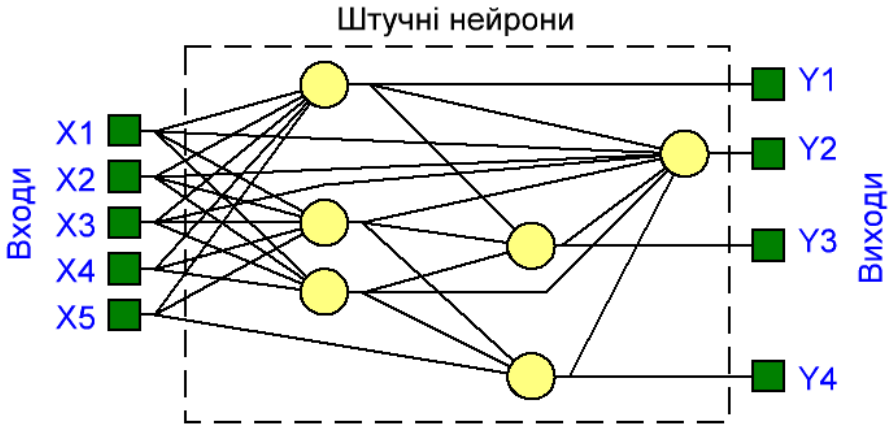


Рисунок 5.2 – Нейронна мережа

Штучний нейрон складається із двох основних структурних елементів: вагового суматора та порогового елемента (Рисунок 5.3). Ваговий суматор приймає сигнали $X_1 \ldots X_n$ із входів штучного нейрону, множить кожен із них на відповідні вагові коефіцієнти $W_1 \ldots W_n$ а потім обчислює суму отриманих добутків. Таким чином, сигнал на виході вагового суматора S обчислюється за формулою:

$$S = \sum_{i=1}^{n} X_i W_i \,. \qquad (1)$$

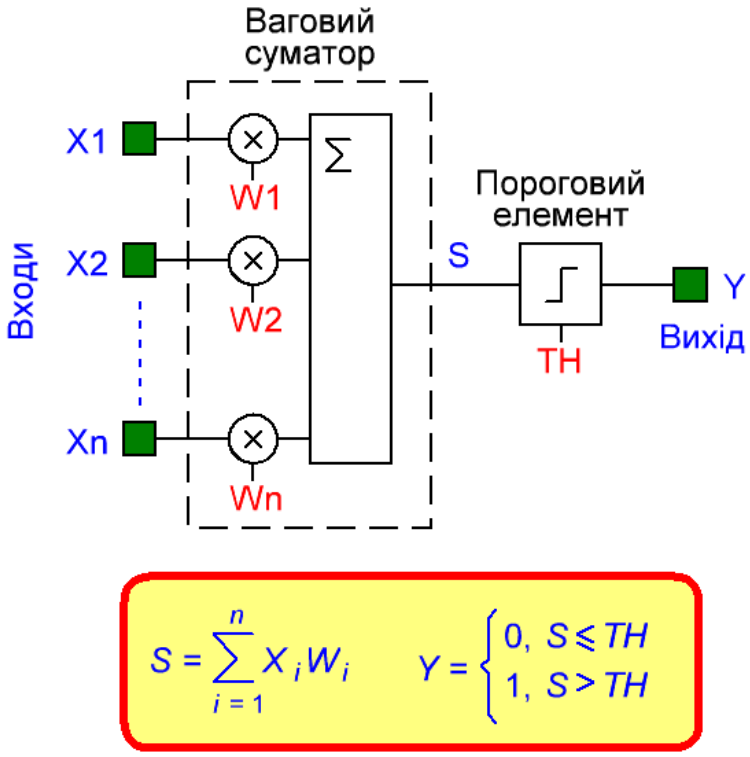


Рисунок 5.3 – Структурна схема штучного нейрона

Цей сигнал подається на вхід порогового елемента, який формує вихідний сигнал *Y*, відповідно до правила:

$$Y = \begin{cases} 0, \text{ якщо } S \le TH \\ 1, \text{ якщо } S > TH \end{cases} \qquad (2)$$

Таким чином, якщо сигнал *S* на вході порогового елемента більше за деякий поріг *TH*, то на його виході встановлюється сигнал із рівнем, який відповідає рівню логічної одиниці. В іншому випадку – коли вхідний сигнал *S* менше порогового значення *TH* – на виході штучного нейрону встановлюється сигнал із рівнем, що відповідає рівню логічного нуля. Деякі фахівці стан, коли вихідний сигнал штучного нейрону дорівнює 0, називають звичайним або незбудженим станом нейрона, а стан, у якому на його виході присутній сигнал із рівнем логічної одиниці – збудженим.

Вагові коефіцієнти $W_1 \ldots W_n$ імітують рівень синаптичних зв'язків між нейронами у реальному біологічному організмі. Чим більше значення цих коефіцієнтів, тим більша можливість переходу нейрона в збуджений стан. Проте слід також розуміти, що ймовірність переходу нейрона в збуджений стан також має зворотну залежність від порога чутливості *TH* – чим менше значення порога чутливості, тим більша імовірність переходу штучного нейрона у збуджений стан.

Логічна функція (2) називається активаційною функцією нейрона. У даному випадку, залежність вихідного сигналу Y від вхідного S має стрибкоподібний характер (Рисунок 5.4), через що подібну функції іноді називають «функцією-сходинкою». За такій функції вихідна напруга нейрона має дискретний характер, проте це не обов'язково – наразі існують нейронні мережі, у яких вихідна напруга нейронів може приймати будь яке значення, а передавальна функція може мати іншу форму, наприклад, лінійну, експоненціальну, логарифмічну або сигмаподібну.

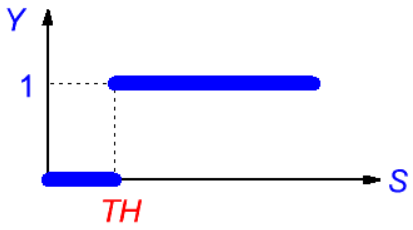


Рисунок 5.4 – Передавальна функція порогового елемента

5.2 Опис схеми дослідження

Схема дослідження містить штучний нейрон, що має два входи: *X1* та *X2*, сигнали з яких потрапляють на входи суматора через відповідні перемножувачі (Рисунок 5.5). Вихідний сигнал штучного нейрона *Y* формується як результат порівняння вихідного сигналу вагового суматора *S* із пороговим значенням *TH*.

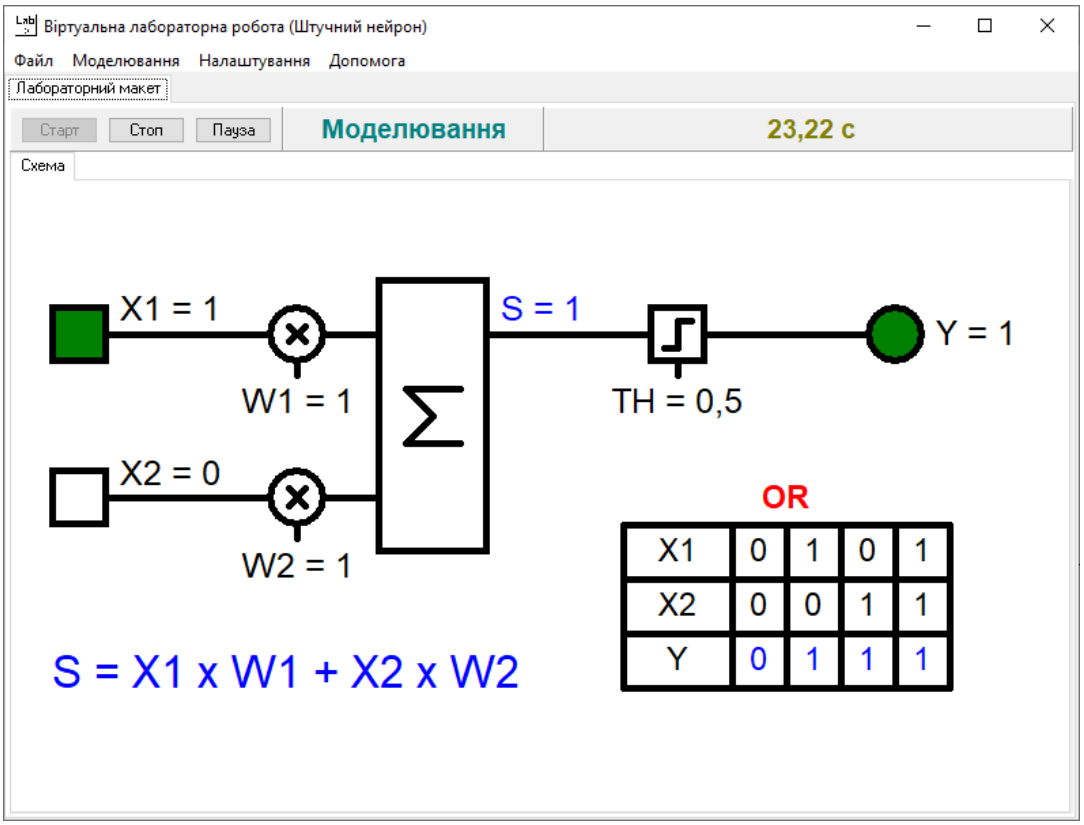


Рисунок 5.5 – Досліджувана схема

Вхідні сигнали X1 та X2 можуть приймати лише два значення: 0 або 1. Для того, щоб змінити рівень сигналу необхідно підвести до них курсор миші (вони при цьому будуть виділені червоним кольором) та один раз клацнути лівою кнопкою.

Коефіцієнти W1 та W2 можуть приймати будь-яке значення у діапазоні –10…10. Для того, щоб змінити коефіцієнт передачі перемножувачів, необхідно підвести до них курсор миші (вони при цьому будуть виділені червоним кольором) і двічі клацнути лівою кнопкою. При цьому відкриється діалогове вікно, у якому можна буде встановити необхідний параметр (Рисунок 5.6).

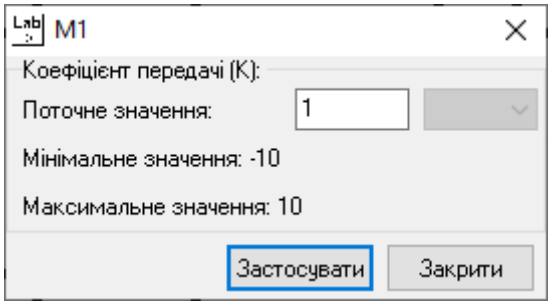


Рисунок 5.6 – Вікно редагування значення вагового коефіцієнту

Поріг збудження штучного нейрона ТН змінюється аналогічно. Для цього необхідно підвести курсор миші до порогового елементу (він при цьому буде виділений червоним кольором) та двічі клацнути лівою кнопкою. При цьому відкриється вікно, аналогічне діалоговому вікну редагування параметрів вагових коефіцієнтів (Рисунок 5.7).

Для того, щоб розпочати моделювання необхідно натиснути кнопку «Старт», при цьому макет перейде у активний стан. Призупинити або зовсім зупинити моделювання можна за допомогою кнопок, відповідно, «Пауза» та «Стоп».

У режимі моделювання, у верхній частині вікна буде сформований напис «Моделювання» та буде відображатися загальна кількість промодельованого часу. Якщо рівень сигналів на вході (X1 та X2) та виході (Y) штучного нейрона відповідає рівню логічної одиниці, то ці входи та вихід будуть підсвічуватися зеленим кольором.

В правій нижній частині робочого вікна досліджуваної схеми розташована таблиця, в якій розміщені вихідні сигнали нейрона Y для усіх можливих комбінацій вхідних сигналів X1 та X2. Якщо ця комбінація співпадає з однією із типових функції (NOT, OR, AND, XOR) або їх інверсних версій (NOR, NAND, NXOR) (Рисунок 5.7), то ця назва буде виведена у верхньому рядку таблиці.

OR		
X1	X2	Y
0	0	0
1	0	1
0	1	1
1	1	1

NOR		
X1	X2	Y
0	0	1
1	0	0
0	1	0
1	1	0

AND		
X1	X2	Y
0	0	0
1	0	0
0	1	0
1	1	1

NAND		
X1	X2	Y
0	0	1
1	0	1
0	1	1
1	1	0

NOT (X1)		
X1	X2	Y
0	0	1
1	0	0
0	1	1
1	1	0

NOT (X2)		
X1	X2	Y
0	0	1
1	0	1
0	1	0
1	1	0

XOR		
X1	X2	Y
0	0	0
1	0	1
0	1	1
1	1	0

NXOR		
X1	X2	Y
0	0	1
1	0	0
0	1	0
1	1	1

Рисунок 5.7 – Таблиці істинності для типових логічних функцій: інверсії (NOT), кон’юнкції (логічне І, AND), диз'юнкції (логічне АБО, OR), логічного додавання (розділяюче АБО, XOR), та їх інверсних версії (NAND, NOR та NXOR)

5.3 Завдання для роботи

Головною метою цієї роботи є «відчути» як працює штучний нейрон – найпростіший елемент нейронних мереж. Тому завдання є дуже простим – шляхом підбору необхідного значення коефіцієнтів W1, W2 та TH налаштувати штучний нейрон таким чином, щоб він відтворював усі типові логічні функції, які наведені на рисунку (Рисунок 5.7). Результати досліджень необхідно занести до таблиці, аналогічній таблиці (Таблиця 5.1). Для підтвердження отримання відповідної функції у звіті про виконання необхідно навести копію екрану схеми дослідження.

Таблиця 5.1 – Налаштування штучного нейрона

Функція	X1	X2	TH
NOT (X1)			
NOT (X2)			
OR			
NOR			
AND			
NAND			
XOR			
NXOR			

5.5 Вказівки до виконання та корисні поради

Під час проведення дослідження зверніть увагу, що логічні операції бувають унарні (від одного аргументу) (NOT) та бінарні (від двох аргументів) (OR, AND, XOR). Для унарних операцій слід реалізувати функцію для кожного входу (X1 та X2).

Якщо у назві логічної функції першою стоїть літера «N», то це означає лише додаткову інверсію вихідного сигналу (окрім функції NOT). Таблиця істинності для цих функцій така ж сама як і для оригінальних функції, тільки усі значення Y замінені на протилежні. Наприклад, NOR еквівалентно OR + NOT.

Один і той же результат може бути отриманий при різних значеннях коефіцієнтів W1, W2 та TH. Наприклад, функція OR може бути реалізована при значеннях коефіцієнтів: W1 = 1, W2 = 1, TH = 0,5, та W1 = 0,5, W2 = 0,7, TH = 0,4.

Не забувайте, що коефіцієнти W1, W2 та TH можуть приймати негативне значення.

Якщо коефіцієнт W1 = 0, то вихід Y ніяк не буде реагувати на зміну рівня сигналу на вході X1, тобто цей вхід начебто відключений. То ж саме стосується і каналу, пов’язаному із входом X2. Це буде корисно при реалізації унарних операцій.

Для експорту схеми можна скористатися пунктом меню «Файл → Експорт результатів → Екран схеми». При його використанні схема буде скопійована у буфер обміну як рисунок, після цього її легко можна буде вставити у звіт (наприклад, скориставшись пунктом меню Правка → Вставити).

Якщо у вас тривалий час не виходить реалізувати певну логічну функцію, то слід припинити подальші спроби – скоріш за все ви ще не все знаєте. Відмітьте це у звіті (наприклад, «не вдалось налаштувати») та продовжуйте роботу над іншою функцією.

5.6 Зміст звіту про виконання

1. Назва роботи

2. Мета роботи

3. Схема дослідження

4. Результати дослідження (Таблиця 5.1 та скріншоти для кожної логічної функції)

5. Висновки

Практична робота 2. Навчання нейронної мережі

6.1 Теоретичні положення

6.1.1 Принципи побудови нейронних мереж

Штучний нейрон (artificial neuron, математичний нейрон, нейрон Маккалока-Пілттса, формальний нейрон), запропонований американськими вченими Ворреном Маккалоком та Уолтером Пилттсом (Рисунок 6.1), є головним вузлом штучної нейронної мережі. Так само як і біологічний нейрон, математичний нейрон має кілька входів X_1, X_2…X_n та один вихід Y.

Рисунок 6.1– Штучний нейрон

Кожен штучний нейрон складається із двох основних елементів – вагового суматора та активаційного елемента (Рисунок 6.2). Ваговий суматор арифметично складає рівень сигналів на входах штучного нейрона і передає обчислену суму на вхід активаційного елемента. Активаційний елемент аналізує сигнал на своєму вході і формує вихідний сигнал, відповідно до правила, яке називається активаційною функцією.

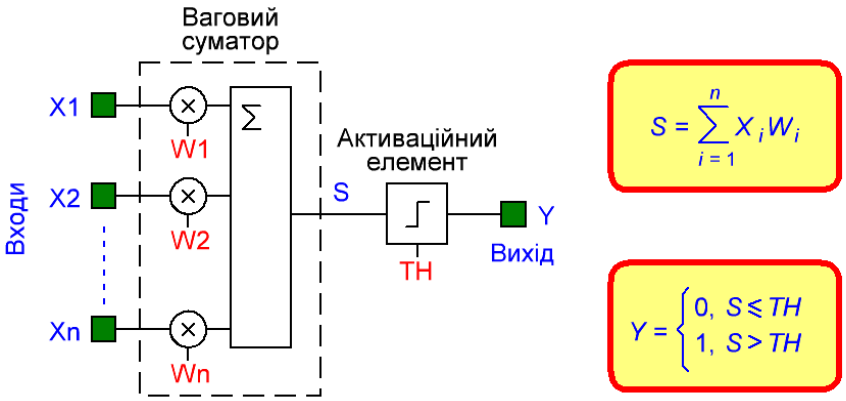


Рисунок 6.2 – Структурна схема штучного нейрона

Зверніть увагу, що вхідні сигнали надходять до суматора не безпосередньо, а через помножувачі, які обчислюють добуток вхідного сигналу на певний статичний ваговий коефіцієнт W. Ці помножувачі є еквівалентами синапсів у біологічних системах, і вони можуть суттєво змінити вплив даного входу на стан нейрону. Якщо ваговий коефіцієнт більше нуля, то сигнал на відповідному вході буде збуджувати нейрон. Причому, чим більше абсолютне значення вагового коефіцієнту, тим суттєвіший вплив цього входу на стан нейрона. Якщо ж ваговий коефіцієнт менше нуля, то цей вхід навпаки – буде гальмувати нейрон, і чим більше абсолютне значення буде мати ваговий коефіцієнт, тим більш складніше іншим сигналам буде збудити нейрон. Якщо цей коефіцієнт буде рівним нулю, то сигнал на відповідному вході не буде впливати на стан нейрона.

Функція активації, або передавальна функція штучного нейрона (Activation Function, Excitation Function, Squashing Function, Transfer Function) встановлює залежність вихідного сигналу активаційного елемента від сигналу на його вході. Теоретично, сигнал на виході нейрона може приймати будь яке значення, проте частіше за все використовують такі функції, які забезпечують рівень вихідного сигналу або в діапазоні від 0 до 1, або в діапазоні від –1 до 1. Вибір функції активації залежить від конкретного призначення нейронної мережі. Частіше за все використовують сигмоїдні функції активації, проте це не обов'язково. У деяких випадках, наприклад, коли сигнал на виході нейрона повинен мати уніполярні значення, слід використовувати функції на основі гіперболічного тангенса, у деяких випадках – функції на основі псевдовипрямлених значень вхідного сигналу. А от для реалізації деяких простих нейронних мереж інколи достатньо використовувати нейрони із простою пороговою функцією активації, графік якої нагадує сходинку (Рисунок 6.3).

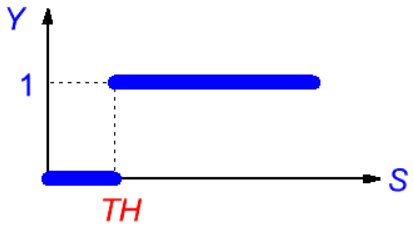


Рисунок 6.3 – Порогова активаційна функція

Штучні нейрони з'єднуються між собою, утворюючи мережу. У загальному випадку, штучна нейронна мережа може мати будь-яку кількість входів та будь-яку кількість виходів. Крім того, самі нейрони всередині мережі можуть з'єднуватися між собою як завгодно, утворюючи безкінечно велику кількість можливих комбінацій. Оскільки системи із довільною конфігурацією дуже важко як аналізувати та реалізовувати, то на практиці розробники нейронних мереж, зазвичай, штучно обмежують себе і використовують лише певну кількість стандартних нейронних мереж.

У цьому випадку, усі нейрони штучної нейронної мережі розділяються на окремі частини, які називають шарами (Рисунок 6.4). Розрізняють вхідний шар, який з'єднує мережу із джерелами вхідних сигналів, та вихідний шар, який з'єднує систему із приймачами результатів обчислень.

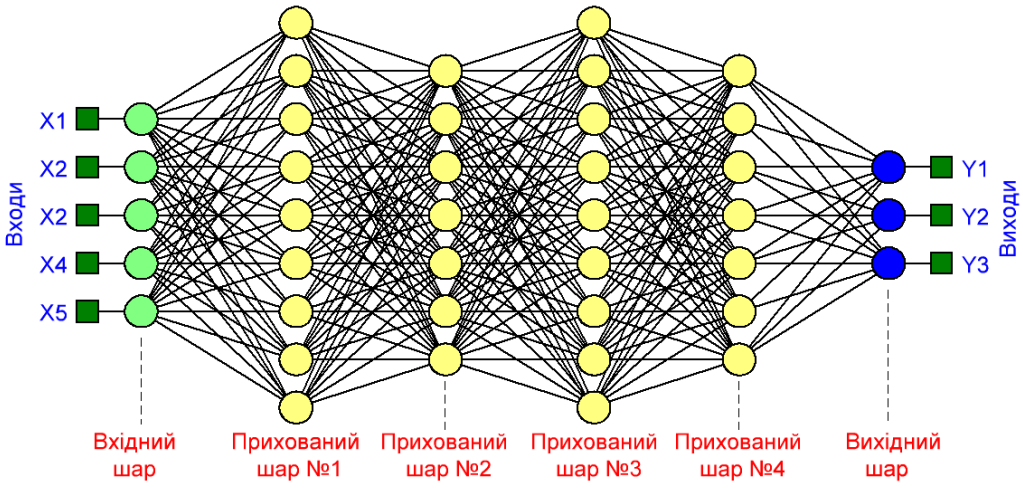


Рисунок 6.4 – Нейронна мережа

Нейрони вхідного шару, зазвичай, не виконують якогось певного логічного навантаження і виконують лише узгоджувальну функцію, що полягає у передачі та нормалізації вхідних сигналів. Нейрони вихідного шару формують вихідні сигнали і остаточно ідентифікують комбінації вхідних сигналів. Приховані шари не мають зв'язку із зовнішнім світом (тому вони і називаються прихованими) і збуджуватися при виявленні певних комбінацій (образів) вхідних сигналів. Наприклад, при розпізнаванні зображень нейрони прихованих шарів можуть збуджуватися при виявленні як простих графічних примітивів (прямих ліній, кіл або полігонів певної форми), так і більш складних образів, наприклад контурів тіла кішки, собаки або інших тварин. При цьому є одна закономірність – чим далі прихований шар знаходиться від вхідного шару, тим більший рівень його абстракції, тобто перший нейрони першого прихованого шару можуть збуджуватися при виявленні певних ліній або кіл, а нейрони другого шару – при виявленні контурів тіла.

6.1.2 Алгоритм навчання нейронних мереж

Одним із найпоширеніших застосунків нейронних мереж є системи розпізнавання образів, наприклад, системи розпізнавання тексту. Першою подібною системою стала нейронна мережа на основі одношарових персептронів, створена у 1960 році Френком Розенблатом, на комп'ютері «Марк-1» (Рисунок 6.5). Цей комп'ютер підключили до 400 сульфід-кадмієвих фотоелементів, які утворили світлочутливу матрицю із роздільною здатністю 20 на 20 елементів. Зв'язки всередині нейронної мережі налаштовувались за допомогою патч-панелі та потенціометрів. Така система могла розпізнавати деякі літери, які підносили до «ока» комп'ютера на спеціальних картках

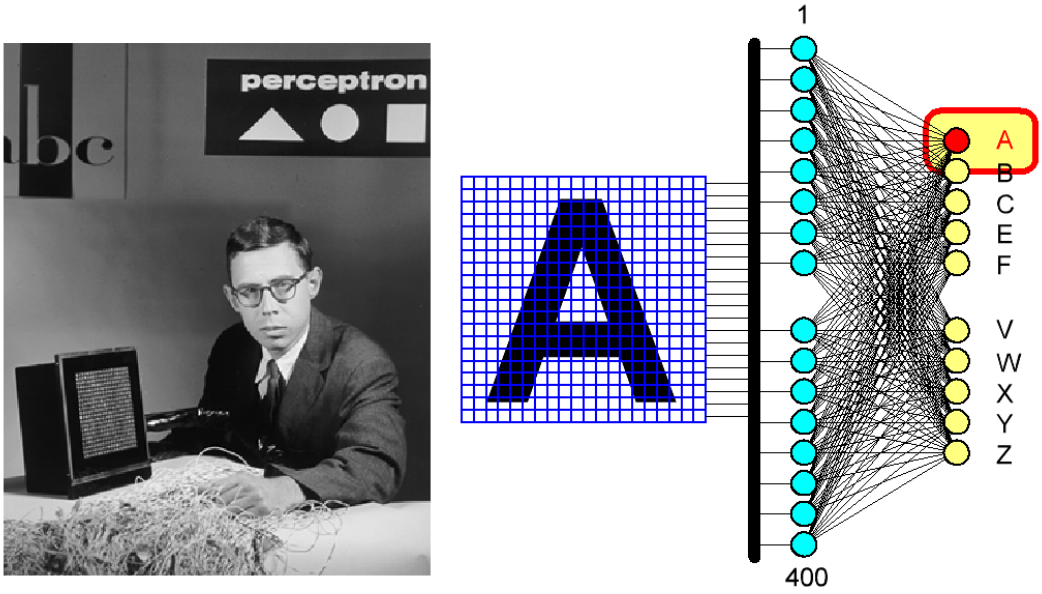


Рисунок 6.5 – Перша нейронна мережа для розпізнавання тексту (1960 р.)

Оцінимо приблизну кількість зв'язків всередині подібної мережі. Нехай мережа повинна розпізнати 28 літер латинського алфавіту, отже її вихідний шар повинен мати 28 нейронів, кожен із яких повинен мати зв'язок із кожним із 400 нейронів вхідного шару. Таким чином, всередині мережі повинно бути 28 х 400 = 11 200 зв'язків, для кожного з яких необхідно підібрати свій ваговий коефіцієнт *W*. Зрозуміло, що ручне налаштування подібної системи, якщо воно взагалі можливе, займе дуже багато часу. Отже для налаштування (навчання) нейронних мереж слід застосовувати спеціальні алгоритми.

Один із алгоритмів навчання нейронної мережі отримав назву «Метод нагород та покарань» (інколи його називають методо «батога та пирога»). Цей метод полягає у наступному.

На початковому етапі – як тільки нейронна мережа була створена – усім ваговим коефіцієнтам усіх нейронів (окрім вхідних, які не приймають участі у розпізнаванні) присвоюють випадкові значення (зазвичай, трохи більше нуля). Порогові значення усіх нейронів також можуть приймати випадкові значення, проте, зазвичай, їх встановлюють однаковими для всіх нейронів. Після цього на входи нейронної мережі подають комплекти тестових сигналів, реакція на які заздалегідь відома. І якщо реакція мережі відрізняється від потрібної, тоді проводять її корекцію. Це можна зробити наступним чином.

Подати черговий набір тестових сигналів і проаналізувати стан усіх нейронів.

Якщо реакція нейрона на тестові сигнали правильна, тоді нічого робити не потрібно.

Якщо реакція нейрона на ці тестові сигнали неправильна і його вихідний сигнал дорівнює 0, тоді до всіх його вагових коефіцієнтів додається величина відповідних вхідних сигналів:

$$W_{i,\,T+1} = W_{i,\,T} + X_i\,; \tag{1}$$

де i – порядковий номер входу нейрона і відповідного вагового коефіцієнта, W – величина вагового коефіцієнту, X – величина вхідного сигналу, T – порядковий номер ітерації навчання (його часто називають номером епохи).

Якщо реакція нейрона на тестові сигнали неправильна і його вихідний сигнал дорівнює 1, тоді від всіх його вагових коефіцієнтів віднімається величина відповідних вхідних сигналів:

$$W_{i,\,T+1} = W_{i,\,T} - X_i\,. \tag{2}$$

Таким чином, якщо нейрон на дану комбінацію вхідних сигналів реагує правильно, то він отримує «пиріг» і на цій ітерації навчання з ним нічого не відбувається. Якщо ж нейрон реагує неправильно, тоді він отримає «батога» і його вагові коефіцієнти необхідно перерахувати.

Навчання проводять до тих пір, поки кількість помилок мережі на весь комплект тестових сигналів не стане меншим за певне значення. У цьому випадку вважається що мережа навчена і нею можна користуватися. Проте у деяких випадках підібрати необхідний набір коефіцієнтів не вдається, що означає, що мережа буде навчатися нескінченно. У цьому випадку додатково контролюють кількість ітерацій навчання (параметр T), і якщо він перевищує певне значення (T_{MAX}), тоді навчання штучно переривають і роблять висновок, що нейронну мережу навчити не вдалося.

6.2 Опис схеми дослідження

Схема дослідження (Рисунок 6.6) призначена для розпізнавання парних та непарних чисел. Вхідними сигналами є піксельне поле, що має розміри 4 х 5 пікселів. Кожен піксель може знаходитися у засвіченому або незасвіченому стані. Якщо піксель знаходиться у засвіченому стані, то він підсвічується зеленим кольором, якщо незасвічений – то білим. Для ручної зміни стану пікселя до нього необхідно підвести курсор миші (при цьому він буде виділений червоним кольором) і один раз клацнути лівою кнопкою.

Піксельне поле має можливість групової зміни яскравості пікселів шляхом подавання на них 10 тестових сигналів, що відповідають цифрам від 0 до 9. Для подачі тестового сигналу необхідно підвести курсор миші до кнопок зі стрілками,

розташованих безпосередньо під піксельним полем (кнопки при цьому будуть підсвічуватися червоним кольором) (Рисунок 6.7). Якщо один раз клацнути лівою кнопкою миші по кнопці зі стрілкою, що спрямована вліво, то тестова цифра зменшиться на одиницю, а якщо до кнопки зі стрілкою, що спрямована вправо – то збільшиться.

Сигнали з піксельного поля подаються на 20 вхідних нейронів, які формують вихідні сигнали наступним чином: якщо піксель засвічений (зафарбований зеленим кольором), то вихідний сигнал вхідного нейрона дорівнює 1, а якщо ні (зафарбований білим кольором) – то 0. При цьому вхідний нейрон може знаходися у збудженому або незбудженому станах. Якщо вхідний нейрон знаходиться у збудженому стані, то він підсвічується червоним кольором, якщо ні – то кольором морської хвилі.

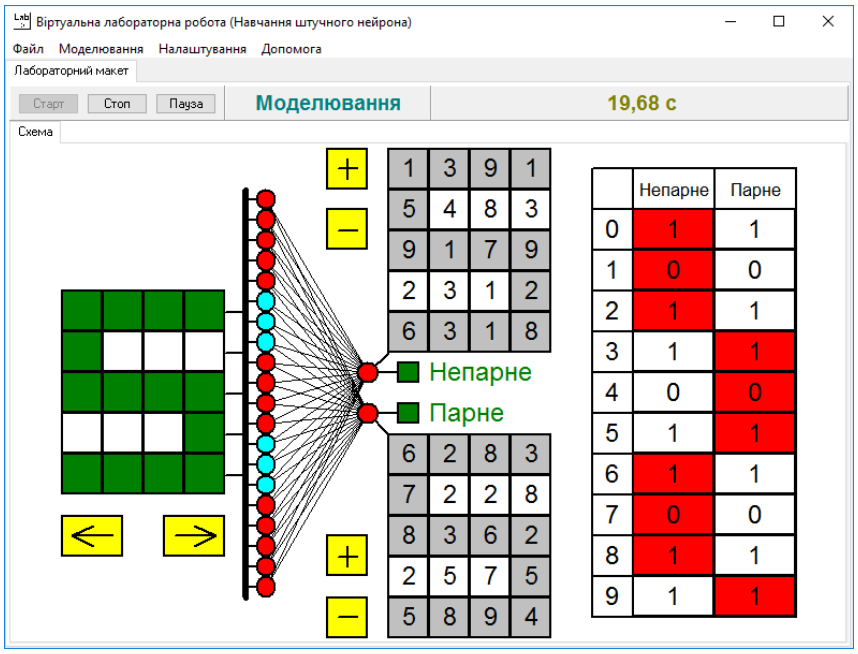


Рисунок 6.6 – Досліджувана схема

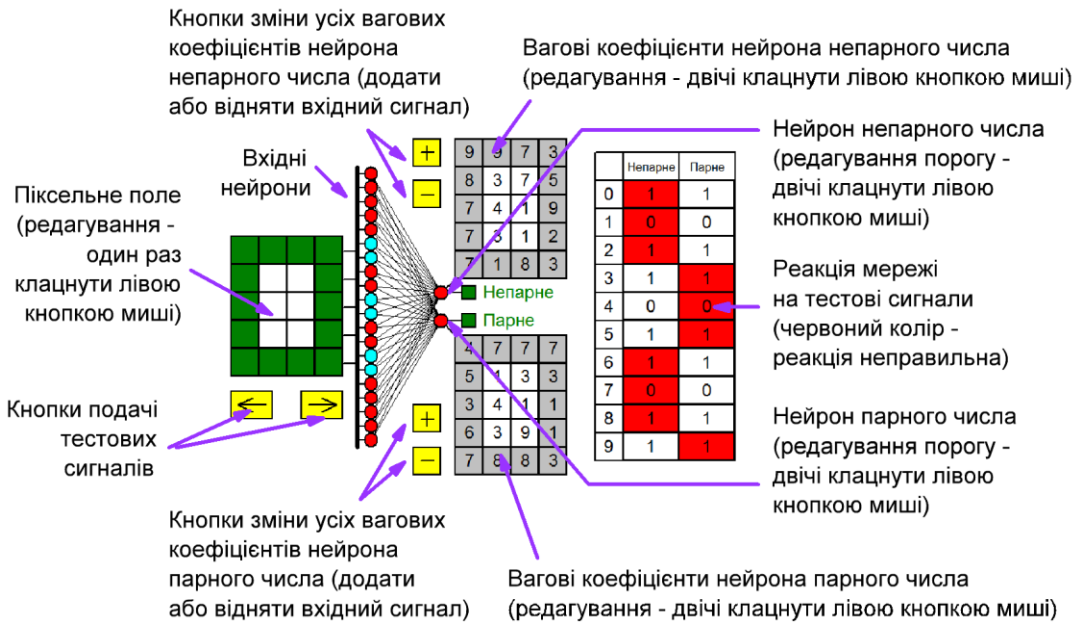


Рисунок 6.7 – Елементи керування макетом

Сигнали з вхідних нейронів подаються на два вихідні нейрони, призначені для розпізнавання, відповідно, парних та непарних чисел. Кожен із вихідних нейронів має 20 вагових коефіцієнтів *W* та порогове значення *TH*. Вагові коефіцієнти розміщені у вигляді таблиці, розташованої біля відповідних нейронів. Кожен із вагових коефіцієнтів можна редагувати вручну – для цього до нього необхідно підвести курсор миші (при цьому він буде виділений червоним кольором) та двічі клацнути лівою кнопкою. При цьому відкриється вікно редагування параметрів, де можна змінити значення цього коефіцієнту (Рисунок 6.8). Діапазон зміни усіх вагових коефіцієнтів – –100…100.

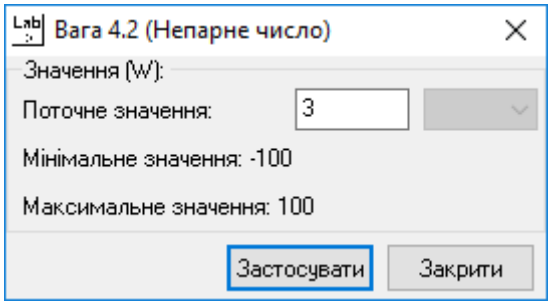


Рисунок 6.8 – Вікно редагування значення вагового коефіцієнту

Поріг збудження штучного нейрона TH змінюється аналогічно. Для цього необхідно підвести до нього курсор миші (він при цьому буде виділений червоним кольором) та двічі клацнути лівою кнопкою. При цьому відкриється вікно, аналогічне діалоговому вікну редагування вагових коефіцієнтів (Рисунок 6.8). Діапазон зміни усіх порогових коефіцієнтів – –100…100.

Увага! Вагові коефіцієнти та пороги збудження вихідних нейронів ініціюються випадковими значеннями під час запуску програми. Тому при наступному запуску програми вони будуть іншими.

Вагові коефіцієнти обох вихідних нейронів можна змінювати групою. Для цього призначені кнопки «+» та «–», розташовані зліва від таблиць вагових коефіцієнтів. Якщо натиснути кнопку «+», то всі вагові коефіцієнти збільшаться на величини відповідних вхідних сигналів, а якщо натиснути на кнопку «–» - то, відповідно, зменшаться на ту ж величину.

Увага! У цій системі вхідні сигнали приймають значення тільки 0 або 1. Тому якщо вхідний сигнал дорівнює 0, то стан вагового коефіцієнта при натисканні на кнопки «+» та «–» не зміниться, а якщо 1 – то збільшиться або зменшиться на 1. Таким чином, при заданому вхідному сигналі і натисканні на кнопки «+» та «–», деякі вагові коефіцієнти зміняться, а деякі ні. Коефіцієнти, які зміняться при натисканні на кнопки «+» та «–» підсвічені у таблицях сірим кольором.

Увага! Кнопки «+» та «–» активні тільки під час моделювання.

Стан вихідних нейронів на схемі відображається кольором. Якщо нейрон знаходиться у збудженому стані (розпізнав парне або непарне число), то він буде підсвічений червоним кольором, а якщо ні – то кольором морської хвилі.

У правій частині схеми розташована таблиця, у якій приведена реакція схеми на всі 10 тестових сигналів. Якщо реакція нейрона правильна, то це поле не підсвічується (має білий колір), а якщо неправильна – тоді це поле підсвічується червоним кольором.

Увага! Правильно налаштована мережа не повинна мати червоних комірок у цій таблиці.

Для того, щоб розпочати моделювання необхідно натиснути кнопку «Старт», при цьому макет перейде у активний стан. Призупинити або зовсім зупинити моделювання можна за допомогою кнопок, відповідно, «Пауза» та «Стоп». У режимі

моделювання, у верхній частині вікна буде сформований напис «Моделювання» та буде відображатися загальна кількість промодельованого часу.

6.3 Завдання для роботи

Головною метою цієї роботи є налаштування (навчання) нейронної мережі, шляхом ручного підбору вагових коефіцієнтів та порогів збудження вихідних нейронів. Правильно навчена мережа повинна правильно розпізнавати парні та непарні числа у діапазоні від 0 до 9.

Результати налаштувань необхідно занести до таблиць, аналогічних таблиці (Таблиця 6.1). Для підтвердження отримання відповідної функції у звіті про виконання необхідно навести копію екрану схеми дослідження.

Таблиця 6.1 – Налаштування штучного нейрона

Поріг збудження (*TH*)			
Вагові коефіцієнти, *W*			

6.5 Вказівки до виконання та корисні поради

Цю роботу рекомендується виконувати за наступним алгоритмом.

1. Встановіть пороги збудження обох вихідних нейронів у діапазоні 40…60.
2. Встановіть усі вагові коефіцієнти обох вихідних нейронів у діапазоні 1…10.
3. Розпочніть моделювання, натиснувши кнопку «Старт».
4. Подивіться на таблицю у правій частині схеми – якщо жодна з комірок не підсвічена червоним кольором, то нічого робити не потрібно – це означає, що мережа навчена. У іншому випадку мережу потрібно навчати.
5. Подайте черговий тестовий сигнал (сформуйте тестову цифру), швидше за все це можна зробити шляхом натискання на кнопки зі стрілками, розташовані під піксельним полем.
6. Зрозумійте самі яка цифра зараз відображається – парна чи непарна, наприклад, цифра «0» є парною, а цифра «1» – непарною.
7. Виконайте аналіз вихідних нейронів. Якщо вихідний сигнал нейрона правильний, то нічого робити не потрібно. Якщо вихідний сигнал нейрона неправильний і дорівнює 0, то його вагові коефіцієнти необхідно збільшити на рівень відповідних вхідних сигналів. Швидше за все це можна зробити, натиснувши на кнопку «+» поряд із відповідною таблицею. Якщо вихідний сигнал нейрона неправильний і дорівнює 1, то його вагові коефіцієнти необхідно зменшити на рівень відповідних вхідних сигналів. Швидше за все це можна зробити, натиснувши на кнопку «–» поряд із відповідною таблицею.
8. Якщо вам іще не набрид процес навчання нейронної мережі, то переходьте то пункту 4, в іншому випадку – здавайтеся.

Під час навчання нейронної мережі не потрібно панікувати. Інколи виникає ілюзія, що ваше втручання лише погіршує стан. Наприклад, до натискання кнопки «+» або «–» у вас неправильно розпізнавалася лише одна цифра, а після натискання набагато більше. Це нормальний процес. Аналізувати загальний стан процесу необхідно тільки після того, як пройде коло із усіх 10 тестових сигналів.

Під час навчання інколи виникає ілюзія, що нічого не відбувається. Наприклад, коли ви кілька разів пройшли весь цикл із 10 цифр, а загальна картина у таблиці, що розташована у правій частині схеми, не змінилася. У даному випадку зверніть увагу на вагові коефіцієнти. Якщо після повного кола тестових сигналів (наприклад, під час подачі сигналів, що відповідають цифрі «0») жоден з них не змінився, і всі вони залишилися такими ж самими, як і не попередній ітерації навчання, то навчання слід припинити – ви зайшли у глухій кут. А якщо хоч один із них змінився, то це означає, що процес навчання триває, і просто вагові коефіцієнти ще не набули потрібних значень.

Якщо ви зайшли у глухий кут, і система не може навчитися, то варто спробувати збільшити пороги збудження вихідних нейронів та спробувати все з початку.

Якщо дуже цікаво, то після налаштування нейронної мережі спробуйте спотворити зображення цифр і подивіться як на це буде реагувати ваша система. Результати цих експериментів також можете навести у звіті (достатньо копії екрану).

Коли будете робити копію екрана схеми дослідження, переконайтесь, що схема знаходиться у режимі моделювання. Копія екрану краще зробити через пункт меню «Файл → Експорт результатів → Екран схеми».

6.6 Зміст звіту про виконання

1. Назва роботи
2. Мета роботи
3. Схема дослідження
4. Результати дослідження (дві таблиці та скріншот)
5. Висновки

Практична робота 3. Розпізнавання символів за допомогою одношарової нейронної мережі

7.1 Теоретичні положення

Досліджувана нейронна мережа призначена для розпізнавання зображень, у першу чергу – текстових символів. Вона відноситься до категорії одношарових персептронів, що працюють із дискретними сигналами і навчаються методом корекції помилки. Основними компонентами досліджуваної нейронної мережі є вхідне піксельне поле, шар вхідних та три вихідних нейрона (Рисунок 7.1). Особливістю цієї мережі є те, що кожен із трьох вихідних нейронів має можливість зберігання чотирьох зразків зображення, на які він повинен реагувати відповідним чином.

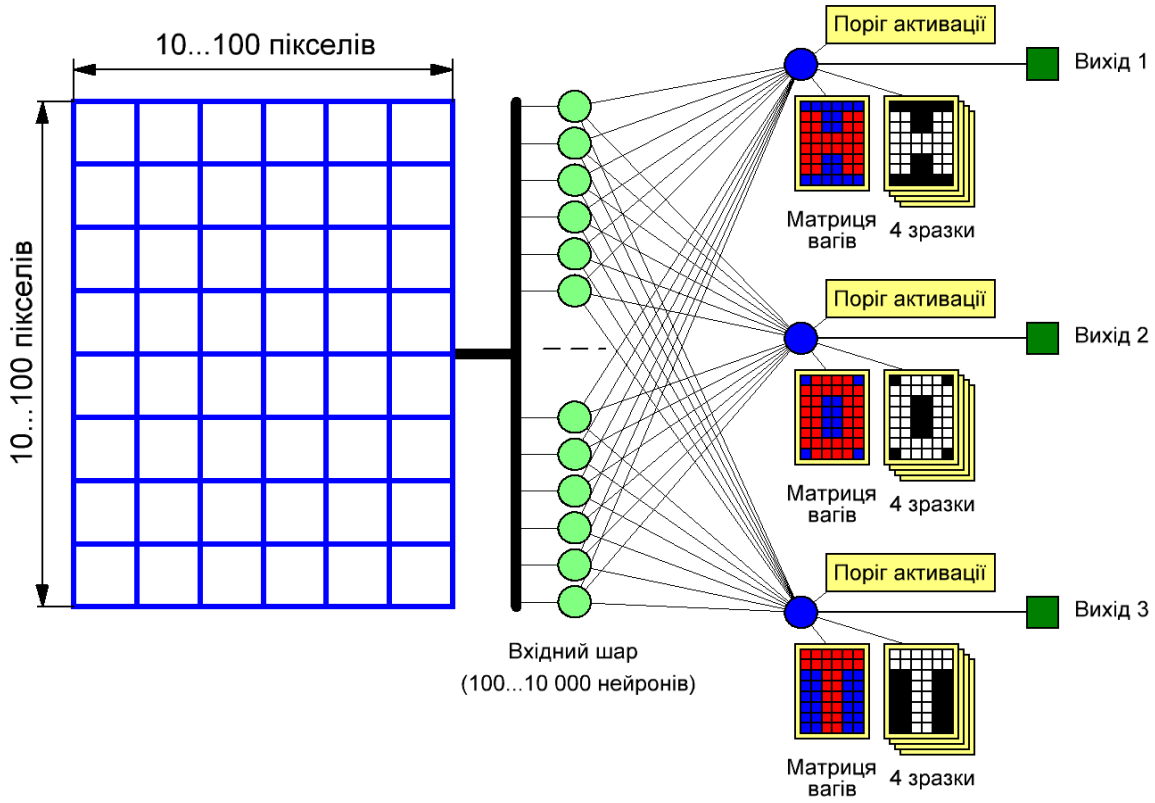


Рисунок 7.1 – Досліджувана схема

Джерелом вхідного сигналу для нейронної мережі є піксельне поле зв’язане із вхідними нейронами. Розмір піксельного поля є змінними і може становити від 10 до 100 пікселів і по горизонталі, і по вертикалі. Таким чином, вхідний шар нейронної мережі може мати від 100 до 10 000 нейронів.

Оскільки первинне зображення може мати відтінки сірого або взагалі бути кольоровим, то перед копіюванням зображення із оригіналу на піксельне поле проводиться його попередня обробка. Для текстів, що містять темні літери на світлому фоні (типовий друкований текст), відбувається інвертування кольорів, оскільки нейронна мережа, так само, як і перший персептрон Розенблата, розрахована на роботу із тестом, що містить світлі літери на темному фоні. Для цього кожній кольоровій компоненті пікселя L_R, L_G, L_B (де L_R, L_G, L_B – відповідно, поточна яскравість червоної, зеленої, та синьої компонент) присвоюється наступне значення L_{R_NEW}, L_{G_NEW} та L_{B_NEW}:

$$L_{R(G,B)\_NEW} = L_{\text{MAX}} - L_{R(G,B)}; \quad (1)$$

де L_{MAX} – максимальне значення, яке може приймати компонента пікселя (для ПЗ, що працює під керуванням ОС Windows, зазвичай, $L_{\text{MAX}} = 255$).

Наступним етапом обробки зображення є видалення кольорів. Для цього спершу обчислюється середня яскравість усіх компонент пікселя L_{MID}:

$$L_{\text{MID}} = \frac{L_R + L_G + L_B}{3}. \quad (2)$$

Після цього усім компонентам пікселя: червоній, зеленій та синій присвоюється однаковий рівень L_{MID}.

Останнім етапом обробки зображення є видалення відтінків сірого, для чого загальна яскравість пікселя L_{MID}, обчислена по формулі (2), порівнюється із порогом L_{TH}. Після цього остаточна яскравість кожної кольорової компоненти встановлюється відповідно правилу:

$$L_{R,G,B} = \begin{cases} 0, \text{ якщо } L_{\text{MID}} < L_{TH} \\ L_{\text{MAX}}, \text{ якщо } L_{\text{MID}} \geq L_{TH} \end{cases}. \quad (3)$$

Схема дослідження розрахована на роботу із зображеннями, що містять текст, сформований темними літерами на світлому фоні, тому перед копіюванням його на вхідну матрицю відбуваються усі три етапи обробки зображення.

Кількість вхідних нейронів і відповідних вагових коефіцієнтів відповідає загальній кількості пікселів піксельного поля і автоматично змінюється зі зміною його розміру. Вихідний сигнал кожного вхідного нейрона є дискретним і повинен приймати значення або 0, або 1. Обчислення стану вхідних нейронів відбувається наступним чином. Спершу обчислюється яскравість пікселя по формулі (2), а потім формується вихідний сигнал X_i кожного нейрона по правилу:

$$X_i = \begin{cases} 0, \text{ якщо } L_{\text{MID}} < L_{TH} \\ 1, \text{ якщо } L_{\text{MID}} \geq L_{TH} \end{cases}. \quad (4)$$

Кожен із трьох вихідних нейронів складається із вагового суматора та активаційного елемента (Рисунок 7.2). Ваговий суматор обчислює зважену суму S усіх вхідних сигналів:

$$S = \sum_{i=1}^{N} X_i W_i; \quad (5)$$

де N – кількість вхідних нейронів; W_i – ваговий коефіцієнт входу X_i.

Вхідний сигнал подається на активаційний елемент, який має порогову функцію активації зі змінним порогом TH. У цьому випадку, вихідний сигнал кожного нейрона обчислюється за правилом:

$$Y = \begin{cases} 0, \text{ якщо } S < TH \\ 1, \text{ якщо } S \geq TH \end{cases}. \quad (6)$$

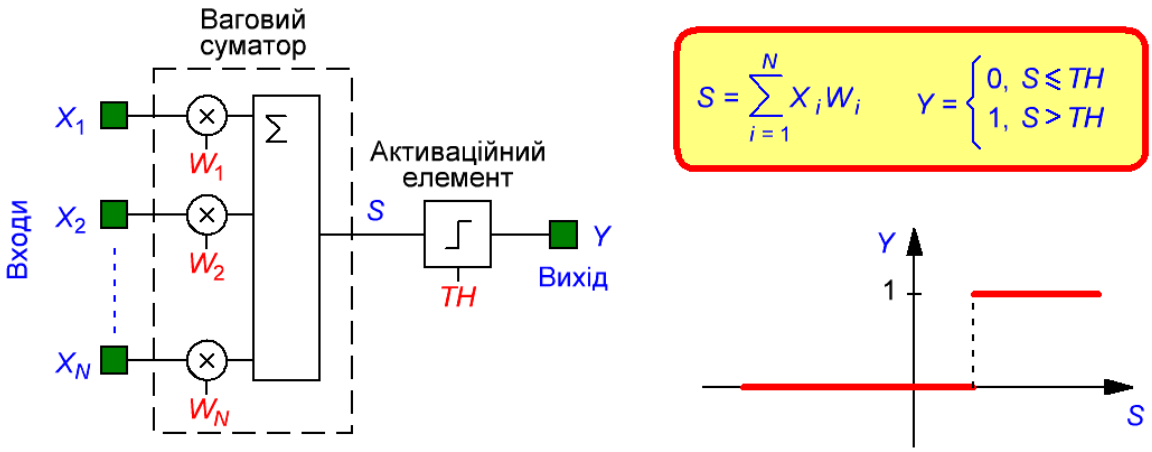


Рисунок 7.2 – Структурна схема вихідних нейронів

Навчання мережі відбувається методом корекції помилки. Для цього у кожного вихідного нейрона є чотири окремі піксельні поля, у яких можуть зберігатися зразки зображень символів, на які він повинен реагувати.

Процес навчання відбувається наступним чином. На входи кожного з нейронів по черзі подаються усі тестові комбінації вхідних сигналів (кожен із 12 зразків). Подача кожної тестової комбінації призводить до появи на його виході певного сигналу Y, який може співпадати або відрізнятися від еталонного сигналу Y_{ET}. У любому випадку, після подачі тестової комбінації сигналів усі вагові коефіцієнти нейрона буде змінено на величину ΔW_i:

$$\Delta W_i = \text{sign}(Y_{ET} - Y) X_i; \qquad (7)$$

де i – індекс вагового коефіцієнта і відповідного вхідного нейрона ($i = 1 \ldots N$), sign – функція, що повертає знак числа:

$$\text{sign}(x) = \begin{cases} -1, \text{ якщо } x < 0 \\ 0, \text{ якщо } x = 0 \\ 1, \text{ якщо } x > 0 \end{cases}. \qquad (8)$$

Таким чином, якщо нейрон правильно реагує на певну комбінацію вхідних сигналів, то його вагові коефіцієнти на зміняться, оскільки, у даному випаду різниця між сигналами $Y_{ET} - Y = 0$ і функція sign() поверне 0.

Якщо вихідний сигнал нейрона дорівнює Y = 0 а повинен бути $Y_{ET} = 1$, тоді різниця між сигналами буде більше нуля ($Y_{ET} - Y > 0$) і функція sing() поверне 1. Таким чином, усі вагові коефіцієнти входів, які мають рівень логічної одиниці ($X_i = 1$), збільшаться на одиницю, а вагові коефіцієнти входів, на яких присутній рівень логічного нуля ($X_i = 0$) не зміняться.

Якщо вихідний сигнал нейрона дорівнює Y = 1 а повинен бути $Y_{ET} = 0$, тоді різниця між сигналами буде більше нуля ($Y_{ET} - Y < 0$) і функція sing() поверне –1. Таким чином, усі вагові коефіцієнти входів, які мають рівень логічної одиниці ($X_i = 1$), зменшаться на одиницю, а вагові коефіцієнти входів, на яких присутній рівень логічного нуля ($X_i = 0$) не зміняться.

Процес навчання буде тривати допоки усі нейрони не почнуть правильно реагувати на комбінації тестових сигналів або допоки не пройде певна кількість ітерацій навчання. У останньому випадку вважається, що систему навчити не вдалося.

Після завершення навчання система готова до роботи і може використовуватись по своєму прямому призначенню

7.2 Опис лабораторного макету

7.2.1 Загальний опис

Лабораторний макет (Рисунок 7.3) містить нейронну мережу, описану в теоретичних положеннях. У верхній частині досліджуваної схеми розташована панель, яка містить 5 кнопок, натискання на які призведе до певних дій із макетом, зокрема:

- кнопка «Файл» призначена для відкриття файлу із зображенням символів, які треба розпізнати (відкриття робочого файлу);
- кнопка «Налаштування» відкриває доступ до діалогового вікна, у якому відкривається доступ до налаштувань нейронної мережі та параметрів сканування;
- кнопка «Навчити мережу» запускає процес навчання нейронної мережі;
- кнопка «Встановити матриці вагів» дозволяє встановити ваговим коефіцієнтам усіх нейронів початкові значення;
- кнопка «Зберегти зразки» дозволяє зберегти зразки у окремі файли із назвою «Sample_xx.bmp», що будуть розташовані у каталозі програми;
- кнопка «Завантажити зразки» дозволяє завантажити зразки із відповідних файлів «Sample_xx.bmp», що повинні бути розташовані у каталозі програми.

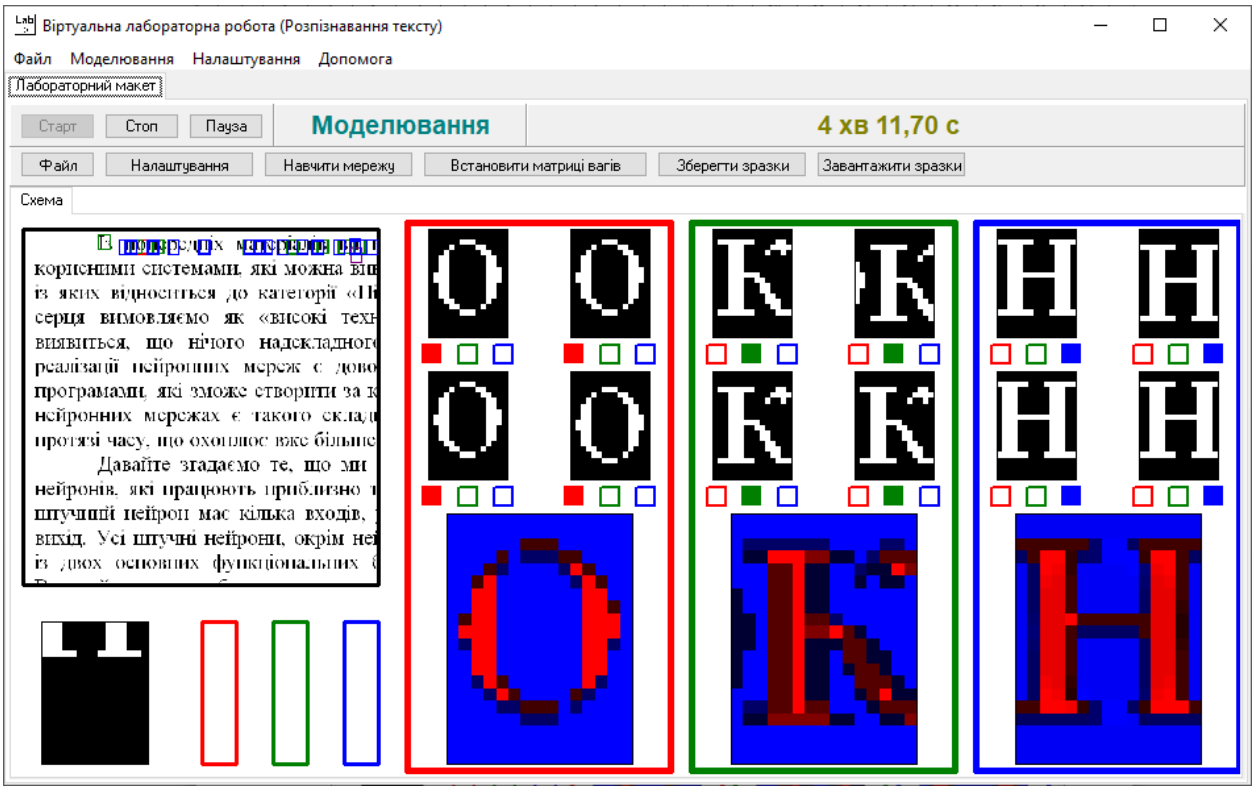


Рисунок 7.3 – Елементи керування макетом

Інтерактивна робоча область макету поділена чотири зони (Рисунок 7.4). В лівій частині розташоване вікно, у якому відображається зображення, яке необхідно розпізнати. Якщо макет не знаходиться у стані розпізнавання, то якщо ввести курсор миші у цю область, у місці розташування курсора будуть показані межі робочої зони, а

зображення буде скопійоване у вхідне піксельне поле. Крім того буде обчислено реакцію нейронів на цю комбінацію сигналів, яка відображається у вигляді трьох прямокутників, кольори яких відповідають кольорам нейронів. Якщо вихідний сигнал нейрона дорівнює одиниці, то, відповідний прямокутник буде повністю забарвлений кольором, а якщо ні, то будуть показані лише контури прямокутника без забарвлення.

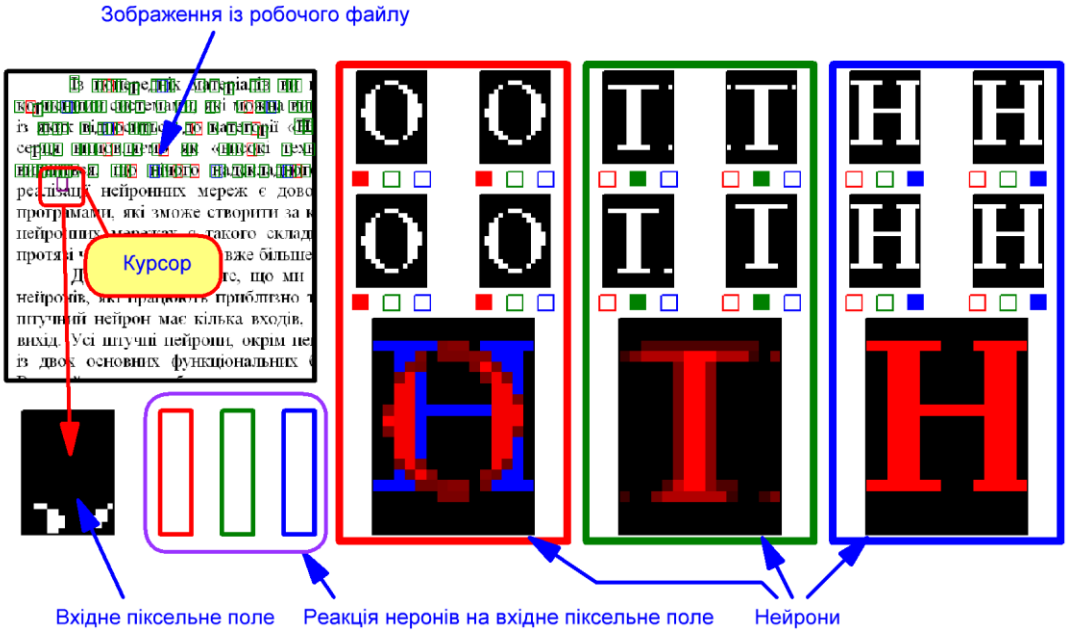


Рисунок 7.4 – Інтерактивна робоча зона

У правій частині робочої зони відображений стан трьох нейронів, кожен з яких має свій колір. У верхній частині розташовані чотири еталонні зразки зображень, які відповідають фрагменту, який необхідно розпізнати. Під кожним зразком розташована реакція нейронів на цей зразок, яка відображається у вигляді трьох прямокутників, кольори яких відповідають кольорам нейронів. Якщо вихідний сигнал нейрона дорівнює одиниці, то, відповідний прямокутник буде повністю забарвлений кольором, а якщо ні, то будуть показані лише контури прямокутника без забарвлення.

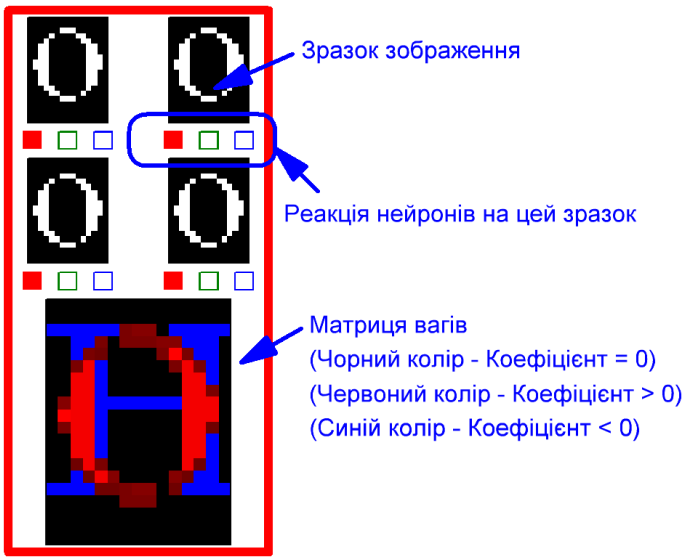


Рисунок 7.5 – Відображення стану нейрону

У нижній частині зони відображення стану нейрону відображається стан вагових коефіцієнтів. Дане поле не призначено для точного кількісного зображення цих значень, і тому має виключно довідкову функцію. Якщо ваговий коефіцієнт дорівнює 0, то відповідний елемент буде відображено чорним кольором, якщо більше нуля – червоним, а якщо менше нуля – то синім. Яскравість червоних та синіх кольорів масштабується автоматично – чим більше цей коефіцієнт відрізняється від нуля – тим яскравіший буде колір цього сегменту.

7.2.1 Завантаження робочого файлу

Для завантаження робочого файлу слід натиснути кнопку «Файл». При цьому відкриється стандартне діалогове вікно, у якому можна обрати потрібний файл. Лабораторний макет підтримує файли із розширенням «.bmp», «.png», «.jpg» та «.jpeg». За замовчуванням відкривається каталог, у якому розташована програма.

Зона відображення робочого файлу має форму квадрату, тому для комфортної роботи слід використовувати файли, що мають відповідні пропорції. За необхідності розмір зображення робочого файлу можна змінити за допомогою будь якого графічного редактора, наприклад Microsoft Paint.

Для роботи рекомендується обирати зображення, що містять текст, надрукований темними літерами на світлом фоні. Файли, що містять фотокопії текстів, можуть мати колір фону, що відрізняється від білого, тому, можливо, для кращого розпізнавання доведеться редагувати поріг збудження вхідних нейронів, доступ до якого відкриється після натискання на кнопку «Налаштування» (налаштування «Рівень білого») (Рисунок 7.6).

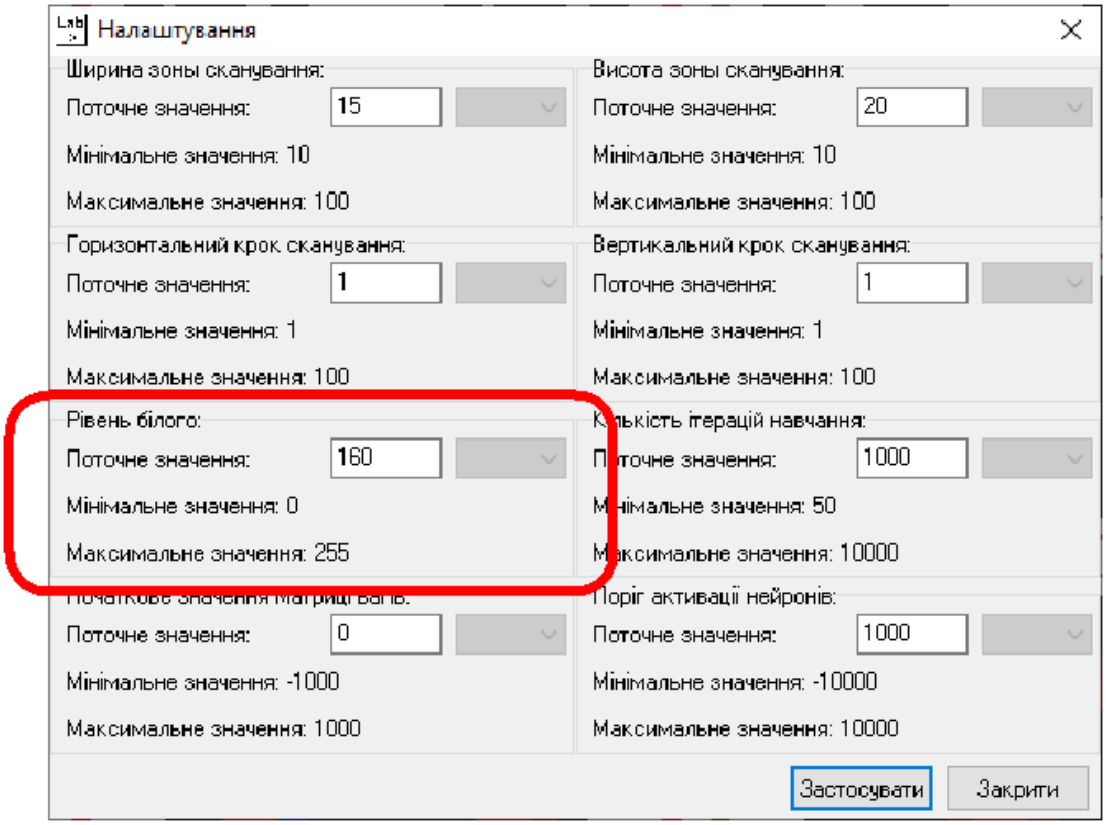


Рисунок 7.6 – Налаштування порогу активації вхідних нейронів

7.2.2 Розмір активної області

Розмір активної області, яка відповідає за розмір вхідного піксельного поля і, відповідно, за кількість вагових коефіцієнтів, можна встановити у діалоговому вікні, що відкривається після натискання на кнопку «Налаштування». Розмір активної області задається у пікселях.

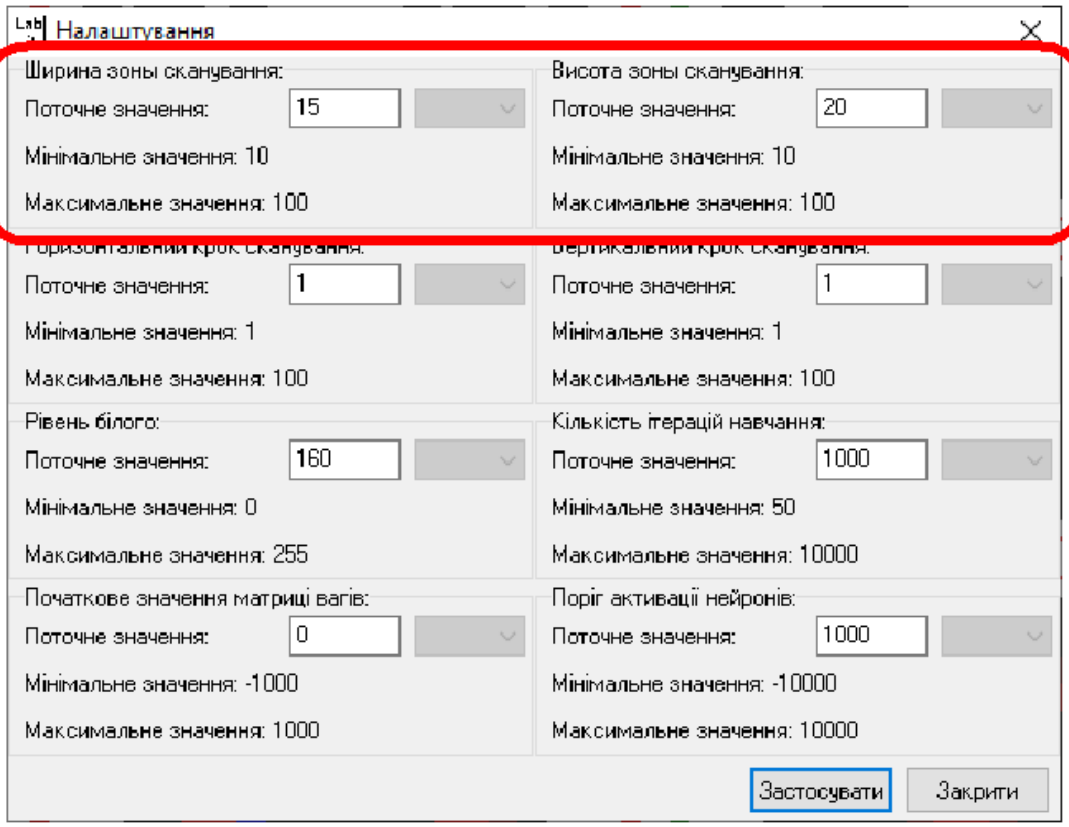


Рисунок 7.7 – Налаштування розмір активної області

Після зміни розміру активної області усі зразки та вагові коефіцієнти будуть очищені, тому робити це необхідно на самому початку роботи із макетом. Розмір активної області краще встановити трішки більшим за розміри самого великого елементу, наприклад, символу, який треба розпізнати.

7.2.3 Робота із зразками

Для створення зразка зображення його необхідно виділити. Для цього слід навести курсор миші на необхідний зразок і один раз клацнути лівою кнопкою. Після цього активний зразок буде виділено жирною червоною рамкою, яка не буде зникати після виведення курсору миші із його зони.

Після виділення необхідного фрагменту робочого файлу слід навести курсор на його зображення і встановити курсор так, щоб символ, обраний у якості зразка, був розташований приблизно у центрі піксельного поля. Після цього слід один раз натиснути ліву кнопку миші, що призведе до копіювання зображення у зразок.

Для спрощення формування зразків після копіювання буде виділено наступний зразок, що значно прискорює процес формування зразків. Проте це не означає, що зразки необхідно формувати саме в такому порядку. Любе натискання лівою кнопкою на зображенні зразка призведе до його виділення, при цьому з інших зразків виділення буде знято автоматично.

Для зняття виділення зі зразка слід ввести курсор миші в його поле і один раз клацнути лівої кнопкою. Після цього жирна червона рамка пропаде, що буде свідчити про деактивацію даного зразка.

Для того щоб очистити зразок необхідно двічі клацнути на ньому лівою кнопкою миші. Після цієї дії зразок буде помічений як «порожній» і не буде приймати участь у навчанні мережі. Зображення при цьому буде сформовано випадковими значеннями.

Зразки можна зберегти у окремі файли, наприклад для подальшого використання. Для цього необхідно натиснути кнопку «Зберегти зразки». Після цього в каталозі, де розташована програма, будуть створені 12 файлів «Sample_xx.bmp». Зверніть увагу, що ця функція відбувається автоматично без будь яких підтверджень чи інших діалогових вікон. Для завантаження зразків із файлів «Sample_xx.bmp» слід натиснути кнопку «Завантажити зразки». Якщо файли присутні на диску, то вони будуть автоматично завантажені.

2.4 Налаштування нейронної мережі

Лабораторний макет дозволяє змінювати пороги активації вихідних нейронів та початкове значення вагових коефіцієнтів. Ці параметри є загальними для усіх трьох вихідних нейронів. Доступ до цих параметрів відкривається у діалоговому вікні, яке з’являється після натискання на кнопку «Налаштування» (Рисунок 7.8).

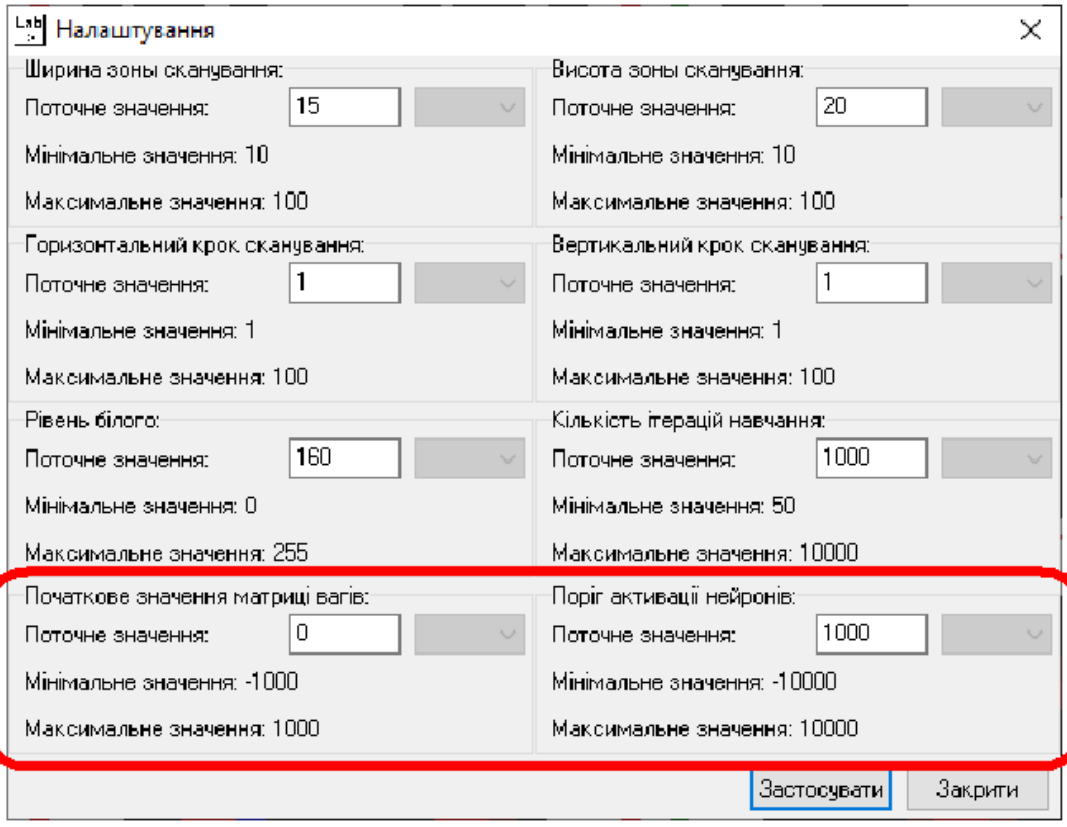


Рисунок 7.8 – Налаштування вихідних нейронів

7.2.5 Навчання нейронної мережі

Для навчання нейронної мережі необхідно натиснути кнопку «Навчити мережу». Після цього запуститься процес навчання, результати якого будуть відбиватися на екрані. Навчання мережі буде проходити допоки усі нейрони не почнуть правильно реагувати на всі зразки, що не помічені як «порожні», або допоки кількість ітерацій не досягне максимального значення, яке можна встановити у діалогову вікні, що відкривається після натискання на кнопку «Налаштування» (Рисунок 7.9).

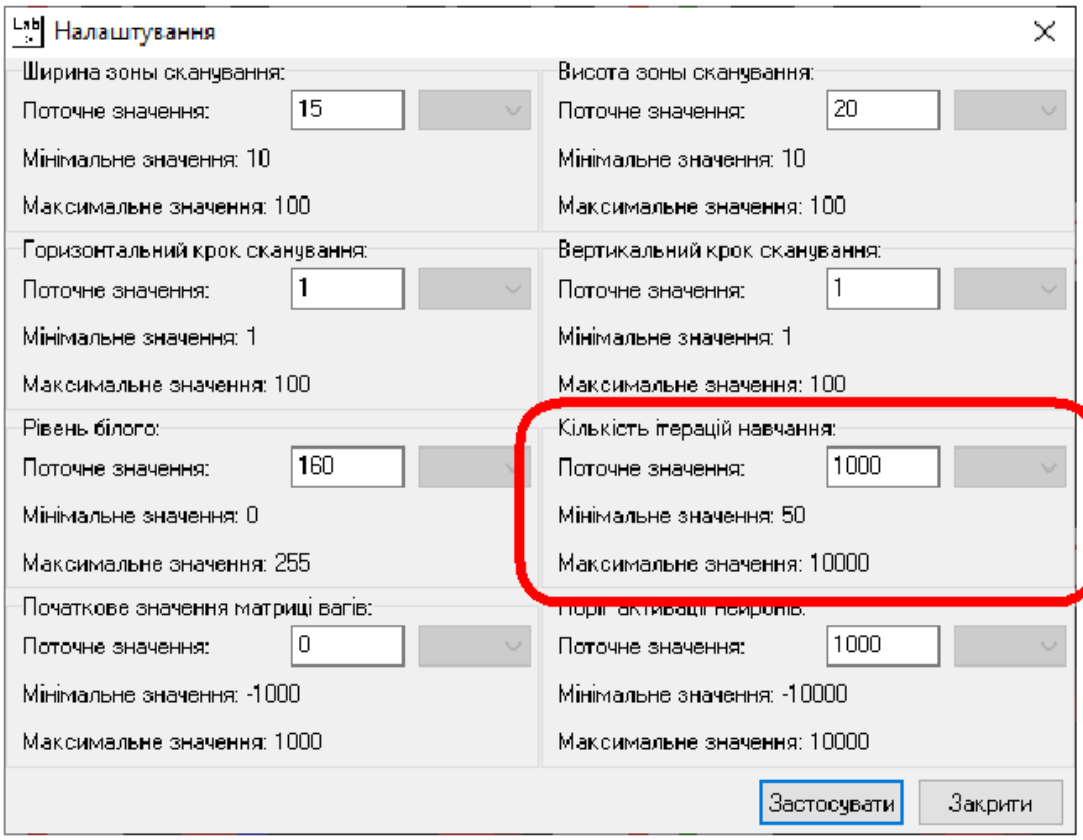


Рисунок 7.9 – Налаштування кількості ітерацій навчання нейронної мережі

Перед першим навчанням рекомендується встановити вагові коефіцієнти вихідних нейронів у початкове значення, що можна зробити натиснувши на кнопку «Встановити матриці вагів».

Ознакою правильно навченою мережі є те, що усі нейрони правильно реагують на зразки (Рисунок 7.10). Якщо з першого разу цього не вдалося, можна спробувати навчити мережу ще раз – для цього слід ще раз натиснути на кнопку «Навчити мережу». Вагові коефіцієнти при цьому скидати не обов'язкового.

2.5 Сканування робочого файлу

Сканування робочого файлу починається після натискання на кнопку «Старт». Поточна позиція курсора відображається на у вигляді курсора, розміри якого

відповідають розмірами піксельного поля. Після встановлення курсора зображення копіюється у піксельне поле та відбувається розрахунок реакції нейронів.

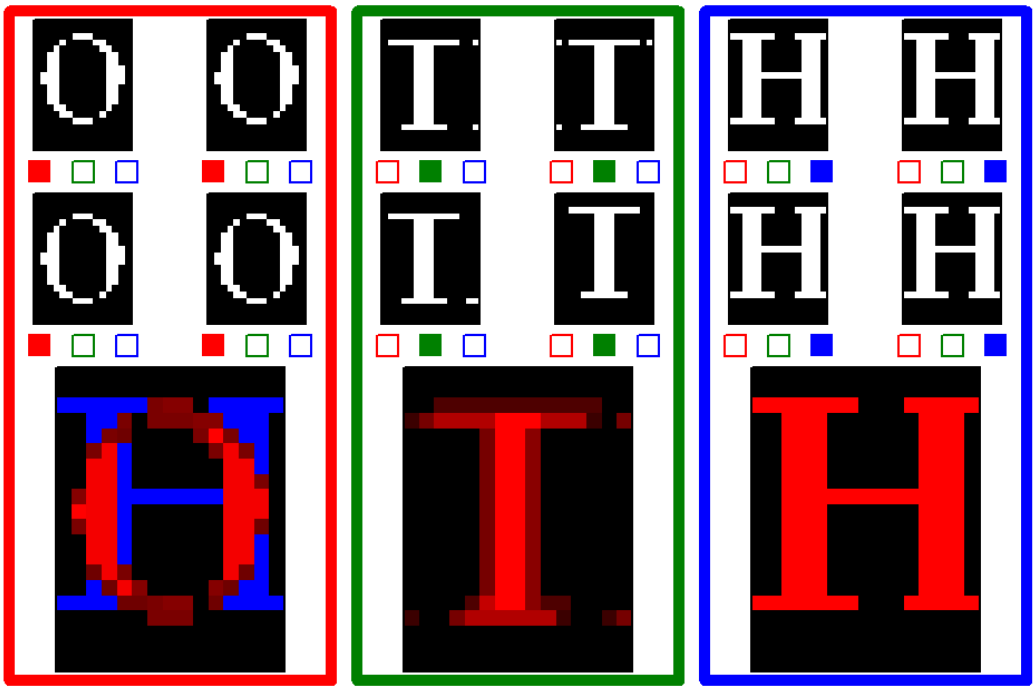


Рисунок 7.10 – Правильно навчена мережа – усі нейрони правильно реагують на усі зразки

Зміщення курсора на наступну позицію відбувається автоматично. Зміщення відбувається построково. Крок зміщення по горизонталі та вертикалі (у пікселях) можна змінити у налаштуваннях, які доступні у діалоговому вікні, що відкривається після натискання на кнопку «Налаштування» (Рисунок 7.11).

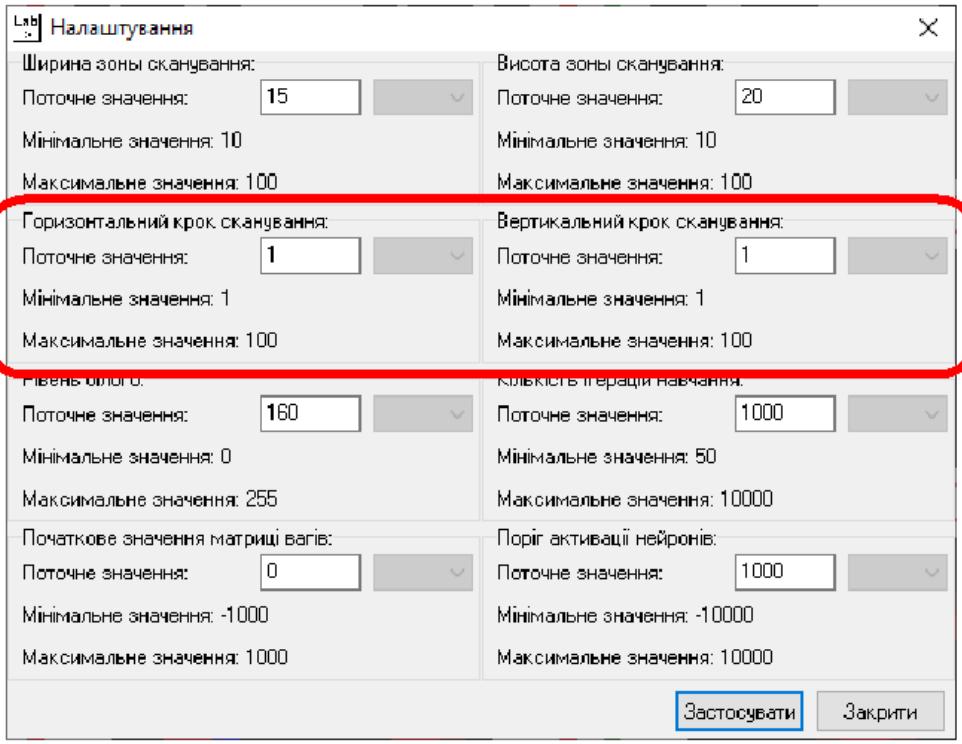


Рисунок 7.11 – Налаштування параметрів сканування

Якщо якийсь із нейронів зреагував на відповідну область зображення, то ця позиція буде збережена та відбита у зоні відображення робочого файлу у вигляді квадрату відповідного нейрона.

Це дозволяє аналізувати результат роботи мережі. Наприклад, після розпізнавання тексту нейронною мережею, налаштованою на розпізнавання літер «о», «н» і «т» видно велику кількість хибних спрацьовувань нейрона літери «т» (Рисунок 7.12). При цьому нейрони, налаштовані на розпізнавання літер «н» та «о», зреагували майже на всі випадки наявності цих літер, хоча але є кілька хибних спрацьовувань нейрона, налаштованого на розпізнавання літери «о», на літеру «ю».

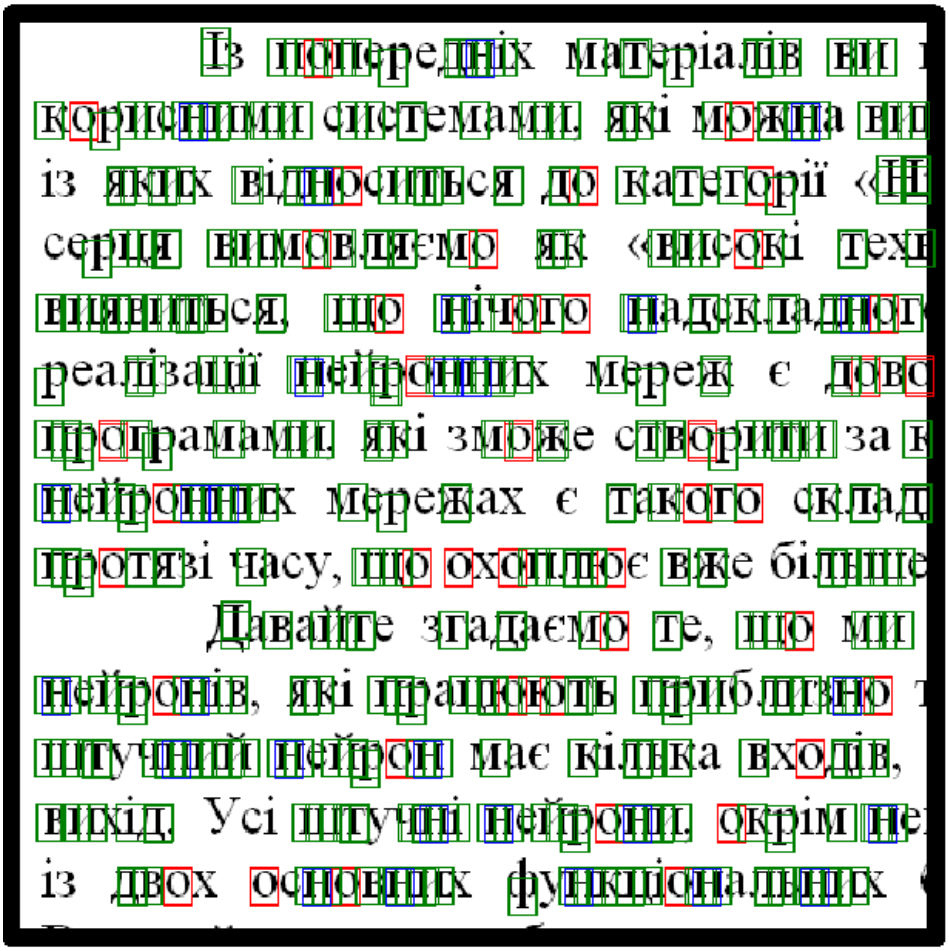


Рисунок 7.12 – Результат роботи нейронної мережі

Призупинити або зовсім зупинити процес розпізнавання тексту можна шляхом натискання на кнопки, відповідно, «Пауза» та «Стоп».

7.2.6 Масштабування схеми

Змінити масштаб схеми можна шляхом обертання коліщатка миші. Зміщувати схему по горизонталі або вертикалі можна шляхом зміщення курсора миші при натиснутій правій кнопці. Збільшити масштаб певного фрагменту схеми можна шляхом його виділення. Для цього слід переміщувати курсор миші при натиснутій лівій кнопці. Для автоматичного масштабування схеми слід двічі клацнути лівою кнопкою миші у любому вільному місці схеми – там де немає компонентів, які активуються при наведення на них курсора миші.

7.3 Завдання для роботи

Головною метою цієї роботи є освоєння методів роботи із одношаровими нейронними мережами із категорії персептронів та визначення меж їх можливостей та практичного застосування.

Завданням для роботи є налаштування нейронної мережі для розпізнавання трьох любих неоднакових символів, які повинні бути присутніми у робочому файлі. Результати досліджень слід проаналізувати, систематизувати та занести до таблиці, аналогічній таблиці (Таблиця 7.1). Для підтвердження проведення роботи у звіті про виконання необхідно навести копію результатів розпізнавання та налаштованої нейронної мережі.

Рисунок 7.1 – Результати розпізнавання зображення

	Нейрон 1	Нейрон 2	Нейрон 3
Символ			
Кількість символів у файлі			
Загальна кількість символів			
Кількість знайдених символів			
Кількість хибних спрацьовувань			
Імовірність знайдення символів			
Імовірність хибного спрацьовування			

7.4 Вказівки до виконання та корисні поради

Цю роботу рекомендується виконувати за наступним алгоритмом.

1. Підготуйте та завантажте потрібний файл.
2. Оберіть розмір вхідного піксельного поля так, щоб він був трошки більший за розмір самого великого символа, який планується розпізнати.
3. Завантажте зразки обраних символів.
4. Навчіть мережу розпізнавати символи.
5. Перевірте чи правильно працює нейронна мережа шляхом встановлення курсова на обраних ділянках файлу – нейрони повинні правильно реагувати.
6. Розпочніть сканування, натиснувши кнопку «Старт».
7. Після завершення сканування оцініть роботу нейронної мережі, якщо ви незадоволені результатом роботи, змініть налаштування і повторіть експеримент.

Якщо ви не знаєте як зробити робочий файл, відкрийте любий текст та зробіть копію екрану (кнопка Print Screen «Prt Scr» на клавіатурі), вставте зображення із буферу обміну у будь який графічний редактор, наприклад Microsoft Paint та збережіть у форматі «.bmp», «.png», «.jpg» або «.jpeg».

Старайтесь не використовувати робочі зображення великого розміру. Якщо робити копію тексту з екрану, то розмір одного символу буде мати приблизно 20 х 25 пікселів. Файл із розміром 500 х 500 пікселів буде скануватися приблизно 1,5 години.

Лабораторний макет не призначений для швидкого розпізнавання, тому при зміщенні курсора на 1 піксель (налаштування за замовчуванням) розпізнавання може зайняти тривалий час. Ви можете скоротити час, встановивши більший крок, проте точність розпізнавання може зменшитись.

Імовірності можна обчислити за наступними формулами. Імовірність знайдення символів:

$$P = \frac{\text{Кількість знайдених символів}}{\text{Кількість потрібних симолів у файлі}} \cdot 100\% . \quad (9)$$

Імовірності хибного спрацьовування:

$$P = \frac{\text{Кількість хибних спрацьовувань}}{\text{Кількість інших символів у файлі}} \cdot 100\% . \quad (10)$$

Де під «потрібним» символом мається на увазі символ, який повинен розпізнати нейрон, а «непотрібними» – усі інші символи.

Копію екрану краще зробити через пункт меню «Файл → Експорт результатів → Екран схеми».

7.5 Питання, на які бажано дати відповіді у висновках

1. А чи зможе ця нейронна мережа розпізнати текст, отриманий зі сканера або фотоапарата?
2. Чому результати роботи мережі саме такі, як ви отримали?
3. Що потрібно зробити для підвищення точності роботи мережі?

7.6 Зміст звіту про виконання

1. Назва роботи
2. Мета роботи
3. Схема дослідження
4. Результати дослідження (таблиця та скріншот)
5. Висновки

Практична робота 4. Дослідження генетичних алгоритмів

8.1 Мета роботи

Провести дослідження генетичних алгоритмів на приклад рішення задачі комівояжера.

8.2 Опис лабораторного макету

Лабораторний макет складається із двох файлів:
- файлу ядра Labs.exe;
- файлу лабораторного макета ML_02.dll.

Для запуску лабораторного макета (макет працює тільки під керуванням операційних систем Windows) необхідно двічі клацнути лівою кнопкою миші по файлу Labs.exe і у вікні, що відкриється, обрати лабораторний макет «Генетичний алгоритм» (Рисунок 8.1).

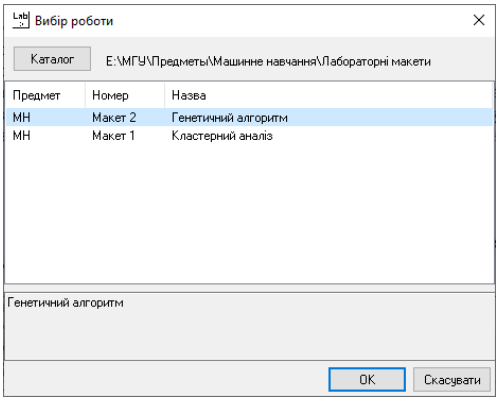


Рисунок 8.1– Вікно вибору лабораторного макету

Після цього відкриється лабораторний макет, в якому вже містяться первинні дані, що є навчальною вибіркою (Рисунок 8.2).

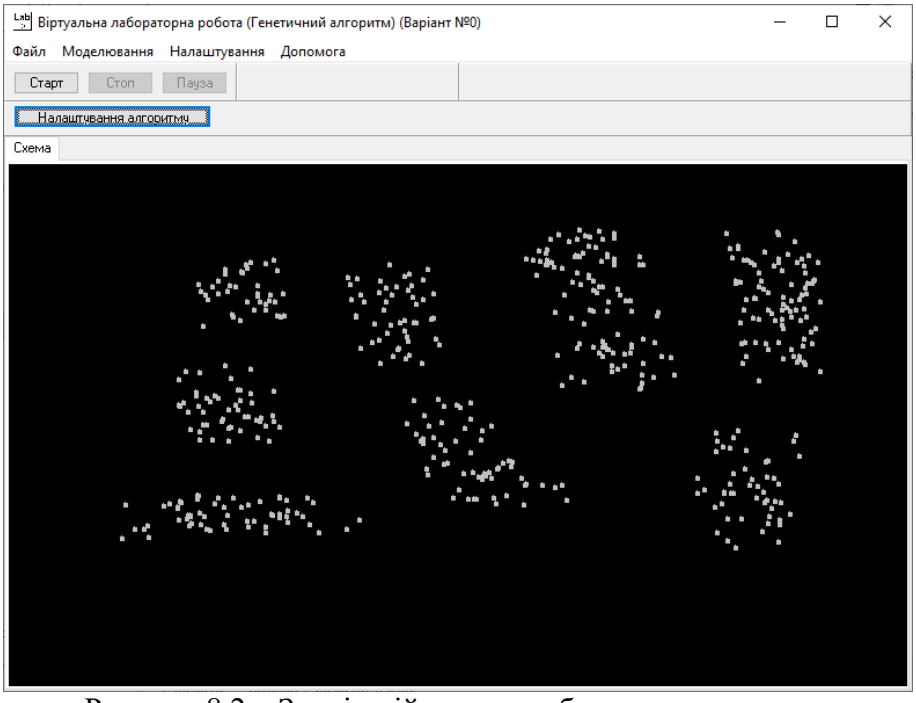


Рисунок 8.2 – Зовнішній вигляд лабораторного макету

Досліджувана вибірка, що містить 500 точок (міст), залежить від номеру варіанту, який можна обрати через меню Налаштування → Номер варіанту (Рисунок 8.3).

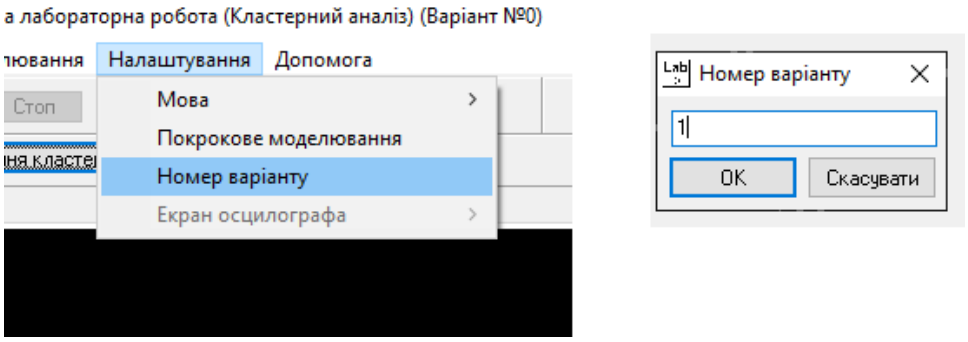


Рисунок 8.3 – Налаштування номеру варіанту лабораторного макету

Параметри генетичного алгоритму можна змінити у вікні налаштувань, яке відкриється після натискання на кнопку «Налаштування алгоритму» Рисунок 8.4).

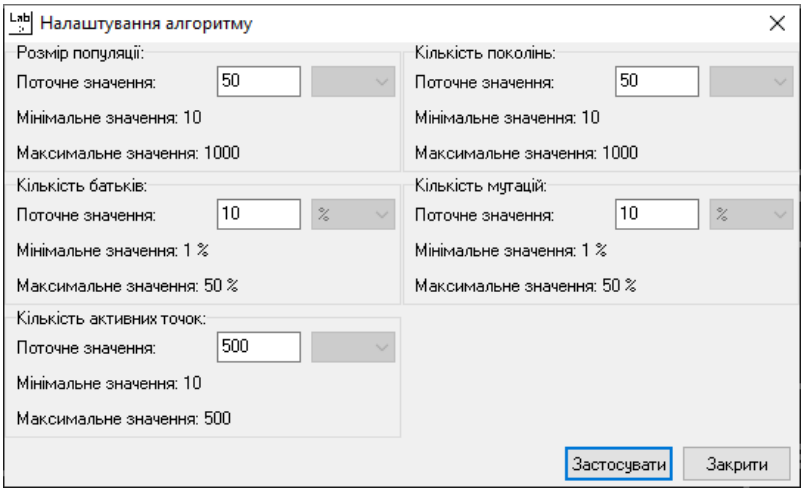


Рисунок 8.4 – Налаштування параметрів генетичного алгоритму

У макеті можна змінювати наступні параметри:

– **кількість активних точок** навчальної вибірки;

– **розмір популяції** – кількість осіб у популяції (абсолютне значення);

– **кількість поколінь** (абсолютне значення) – кількість ітерацій генетичного алгоритму, після завершення яких обчислення буде припинене;

– **кількість батьків** – відносна (у відсотках від розміру популяції) кількість осіб, на основі яких будуть створені особи наступного покоління (діти);

– **кількість мутацій** – відносна (у відсотках від розміру популяції) кількість нових осіб (дітей), над якими буде проведена процедура мутації.

Для початку роботи генетичного алгоритму необхідно натиснути на кнопку «Старт». Після цього почне працювати алгоритм, роботу якого можна спостерігати на екрані лабораторного макету (Рисунок 8.5). На екрані при цьому відображаються:

– найкращій (для даного покоління) маршрут, після завершення роботи алгоритму – результат розрахунку;

– довжина найкращого маршруту у віртуальних одиницях виміру (у нижньому правому куті екрану);

– номер поточного покоління у процесі роботи алгоритму (у нижньому лівому куті екрану), у дужках – загальна кількість поколінь, встановлена у налаштуваннях.

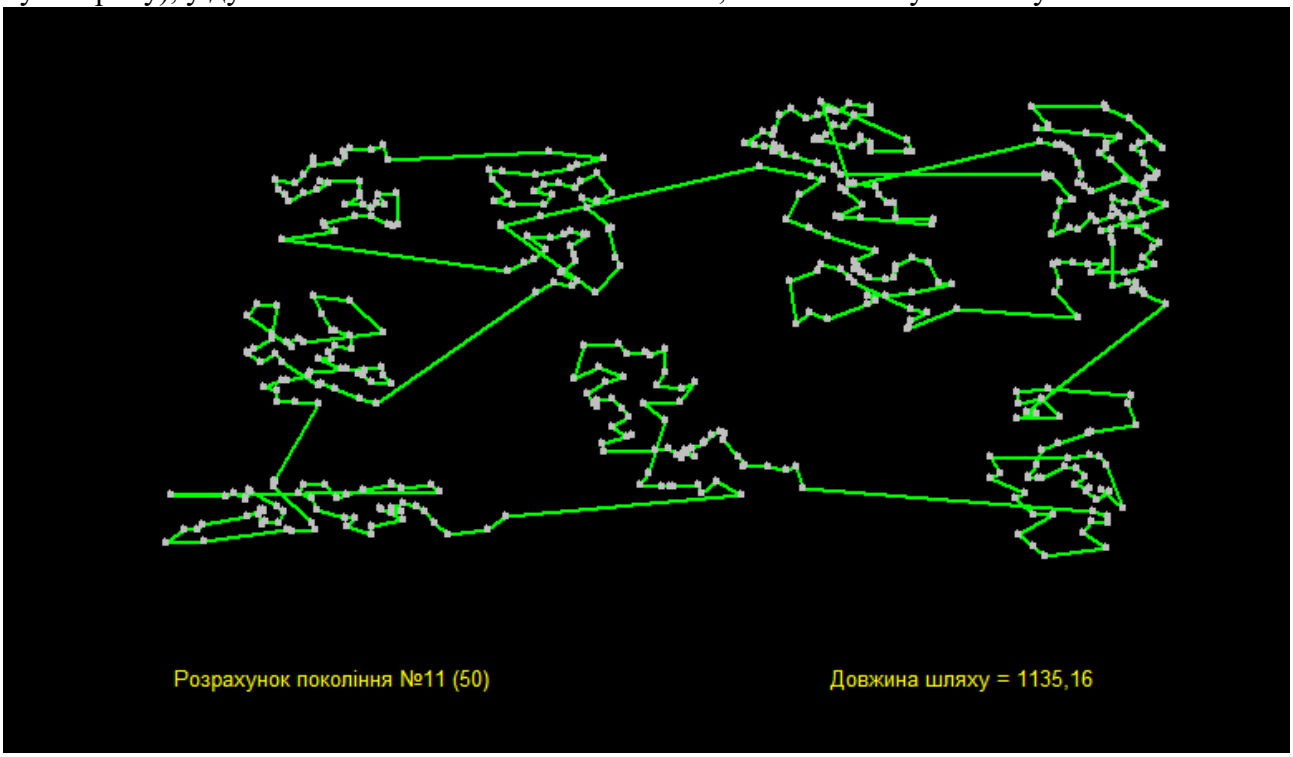


Рисунок 8.5 – Процес роботи генетичного алгоритму

Роботу алгоритму можна у будь який момент призупинити, натиснувши на кнопку «Пауза» або зупинити зовсім, натиснувши на кнопку «Стоп».

8.3 Практичне завдання

8.3.1 Визначити залежність вірогідності знаходження оптимального маршруту від розміру популяції

Навчальна вибірка містить 500 точок. Пошук оптимального маршруту (із високою імовірністю його знаходження) для такої кількості точок може зайняти доволі тривалий час, тому для збільшення швидкості і достовірності досліджень кількість активних точок слід обмежити у налаштуваннях (Рисунок 4.12).

Увага! У цьому макеті застосування нових налаштувань відбувається після натискання на кнопку «Старт» (у момент перезавантаження макету), тому після зміни налаштувань до моменту натискання на кнопку «Старт» на екрані будуть відображені старі дані.

Увага! При зміні кількості активних точок будуть використані перші N точок з навчальної вибірки.

Залежність вірогідності знаходження оптимального маршруту від розміру популяції слід визначити при наступних співвідношеннях розміру популяції до кількості міст: 1:1, 10:1, 100:1. Оскільки максимальна кількість популяції не може перевищувати 10 000 (внутрішнє обмеження лабораторного макету), тому максимальна кількість активних точок не може перевищити 100. Рекомендується провести дане дослідження для 10, 20, 50 та 100 міст.

Оскільки результат роботи генетичного алгоритму має вірогідний характер, то завжди існує імовірність того, що оптимальне рішення буде знаходитись у певному локальному мінімумі. Це означає, що кожне дослідження необхідно проводити кілька разів (не менше 5, рекомендується 10. Результати дослідження рекомендується занести до таблиці (Табл. 8.1).

Таблиця 8.1– Результати експерименту для 10 міст

Співвідношення численності населення до кількості міст	Номер експерименту	Довжина оптимального маршруту
1:1	1	
1:1	2	
1:1	3	
1:1	4	
1:1	5	
1:1	6	
1:1	7	
1:1	8	
1:1	9	
1:1	10	
10:1	1	
10:1	2	
10:1	3	
10:1	4	
10:1	5	
10:1	6	
10:1	7	
10:1	8	
10:1	9	
10:1	10	
100:1	1	
100:1	2	
100:1	3	
100:1	4	
100:1	5	
100:1	6	
100:1	7	
100:1	8	
100:1	9	
100:1	10	

Після проведення усіх експериментів визначити довжину найкоротшого маршруту (мінімальне значення довжини маршруту із усіх експериментів для даної кількості міст) і після цього визначити імовірність P знаходження найкоротшого маршруту для даного співвідношення розміру популяції до кількості міст:

$$P = 100 \cdot N_{опт}/N_{експ}$$

де $N_{опт}$ – скільки разів було знайдено оптимальний маршрут (найкоротший маршрут із усієї серії експериментів);

$N_{експ}$ – загальна кількість експериментів для даного співвідношення розміру популяції до кількості міст.

За результати дослідження слід побудувати залежність імовірності знаходження оптимального маршруту (вертикальна вісь) від співвідношення розміру популяції до кількості міст (горизонтальна вісь). Залежності для різної кількості міст необхідно побудувати в одній координатній системі (не менше 4 графіків).

8.3.2 Дослідження генетичного алгоритму для великої кількості міст

У цьому дослідженні слід спробувати знайти оптимальний маршрут для повної навчальної виборки (500 міст). Налаштування генетичного алгоритму для даного дослідження слід знайти самостійно. Результат роботи (конфігурацію оптимального маршруту) слід навести у протоколі та описати у висновках.

8.4 Зміст протоколу

Результатом виконання цієї практичної роботи є протокол виконання практичної роботи, який повинен містити:

– копії схеми для кожного етапу досліджень (копію схеми можна зробити через меню Файл → Експорт результатів → Екран схеми (буфер обміну));

– досліджувані залежності.

– висновки із обґрунтуванням ваших рішень.

Протокол роботи повинен оформлюватися відповідно до ДСТУ 3008:2015 Звіти у сфері науки і техніки. Структура та правила оформлення.

Протокол роботи завантажується у систему дистанційного навчання для подальшої оцінки.

Практична робота 5. Знайомство з Orange Data Mining

1 Мета роботи

Ознайомлення із застосунком Orange Data Mining та створення першого проєкту.

2 Ключові положення

Orange Data Mining (далі – Orange) є програмним забезпеченням з відкритим вихідним кодом, призначеним для візуалізації даних, машинного навчання та інтелектуального аналізу великих обсягів інформації. Його розробка розпочалася в 1997 році за ініціативою піонера штучного інтелекту – Дональда Мічі, який вважав за необхідне створити відкритий інструментарій для машинного навчання. Розробкою та підтримкою Orange займається Університет Любляни (University of Ljubljana), Словенія. Orange розповсюджується за ліцензією GPLv3 або її пізнішою версією. Це програмне забезпечення можна завантажувати, використовувати та модифікувати без будь-яких фінансових витрат. Orange може працювати на операційних системах Windows, macOS та Linux.

Основною особливістю Orange є його компонентно-орієнтований підхід з підтримкою візуального програмування, що дозволяє користувачам створювати процеси аналізу даних шляхом поєднання різних компонентів, які в середовищі Orange називаються віджетами (Widgets). Для кожної операції з даними, наприклад, для завантаження, попередньої обробки або візуалізації даних, навчання моделей машинного навчання, оцінку їхньої ефективності та багатьох інших, зазвичай, використовується окремий віджет. Такий підхід дозволяє інтерактивно досліджувати дані, створювати моделі машинного навчання та оцінювати їхню продуктивність без необхідності писати код (рис. 1). Крім того Orange також надає можливість використовувати його як бібліотеку Python для більш гнучкої обробки даних та налаштування віджетів, однак для використанні такої можливості потрібно бути досвідченим користувачем.

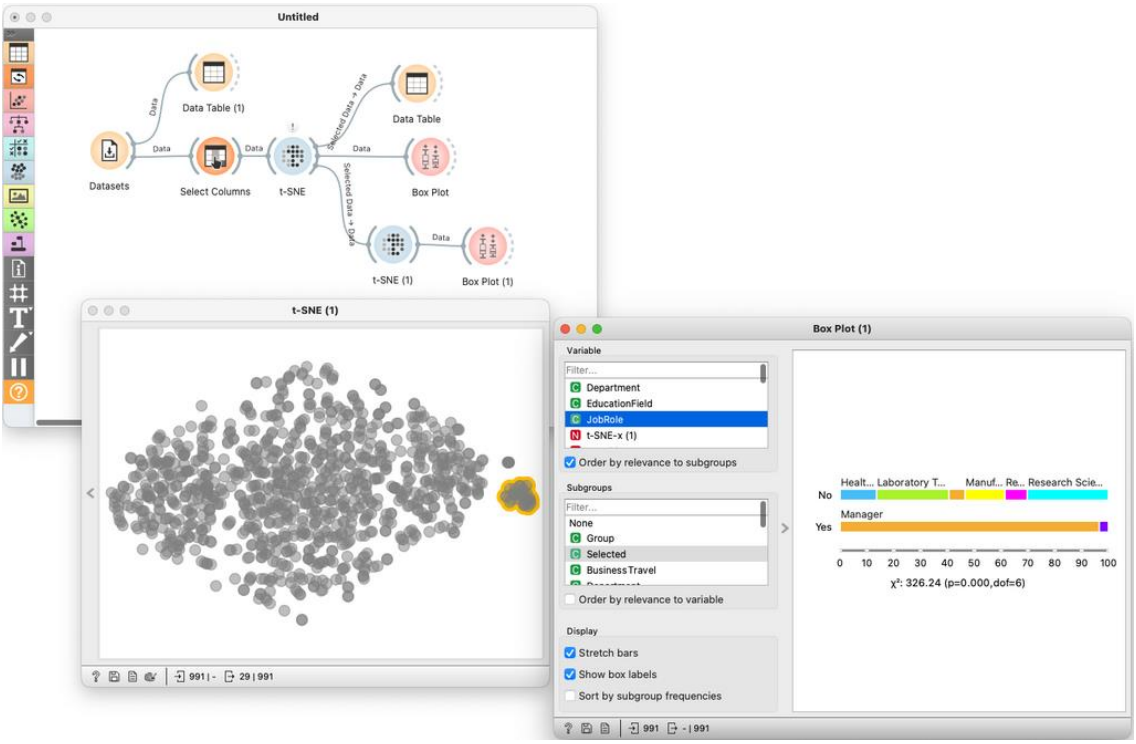


Рисунок 1 – Типовий вигляд проєкту Orange

Orange підтримує різноманітні формати даних, до переліку яких входять файли Excel, CSV, табличні файли та навіть дані, що містяться в мережі Інтернет, наприклад, дані у форматі Google Spreadsheets. Додаткові розширення дозволяють імпортувати зображення, текстові файли та працювати з базами даних SQL. Серед доступних віджетів є інструменти для візуалізації даних, такі як діаграми розсіювання, гістограми, деревоподібні діаграми, теплові карти та багато інших. Orange також має широкий спектр алгоритмів машинного навчання для класифікації, регресії, кластеризації та зниження розмірності даних. Додаткові модулі розширюють функціональність Orange, додаючи можливості для обробки тексту, аналізу зображень, біоінформатики та інших спеціалізованих галузей.

Orange знаходить широке застосування в різних практичних сферах, серед яких біомедицина, біоінформатика, геномні дослідження та освіта. Зокрема, в освітньому середовищі Orange використовується для дослідження методів машинного навчання та інтелектуального аналізу даних. Його візуальний інтерфейс та інтерактивні інструменти дозволяють здобувачам легко зрозуміти складні концепції, експериментувати з різними алгоритмами та методами, а також отримувати негайний зворотний зв'язок через візуалізацію результатів. Це робить Orange ідеальним інструментом для первинного ознайомлення з засобами машинного навчання та штучного інтелекту в освітніх установах. Викладачі та тренери по всьому світу використовують Orange для проведення практичних занять, що підвищує розуміння здобувачами теоретичних концепцій через практичне застосування.

3 Рекомендована література

Основна:

1. Офіційний сайт Orange. URL: https://orangedatamining.com/
2. Getting Started with Orange. URL: https://www.youtube.com/playlist?list=PLmNPv Qr9Tf-ZSDLwOzxpvY-HrE0yv-8Fy

Додаткова:

3. Orange (software). URL: https://en.wikipedia.org/wiki/Orange_%28software%29
4. Мала Ю.А., Селівьорстова Т.В., Гуда А.І. Використання технології Orange для інтелектуального аналізу даних в освітній галузі. «Системні технології» («System technologies»), 2024, №3 (152). С.115–127. DOI: 10.34185/1562-9945-3-152-2024-12.
5. Пронін С.В., Сотников А.Д. Використання платформи Orange для аналізу даних. Вісник ХНАДУ, 2022, вип. 99. С.131–137. DOI: 10.30977/BUL.2219-5548.2022.99.0.131.
6. Iris flower data set. URL: https://en.wikipedia.org/wiki/Iris_flower_data_set

4 Практичне завдання

4.1 Встановити програмне забезпечення Orange Data Mining на локальний комп’ютер.

4.2 Створити перший проєкт на Orange.

4.3 Підготувати звіт про виконання практичної роботи.

5 Методичні вказівки до виконання роботи

5.1 Встановлення Orange

Завантажити останню версію Orange можна з офіційного сайту проєкту, що має адресу: https://orangedatamining.com/ [1]. Для цього на головній сторінці потрібно

перейти у розділ «Download» (рис. 2). Після цього, на сторінці, що відриється за даним посиланням, обрати дистрибутив, що буде працювати на вашій апаратній основі та операційній системі. Зазвичай сайт самостійно виконує діагностику вашого комп'ютера і рекомендує інсталяційний пакет (рис. 3). Встановлення Orange зазвичай не викликає проблем, тому усі налаштування можна залишити без змін.

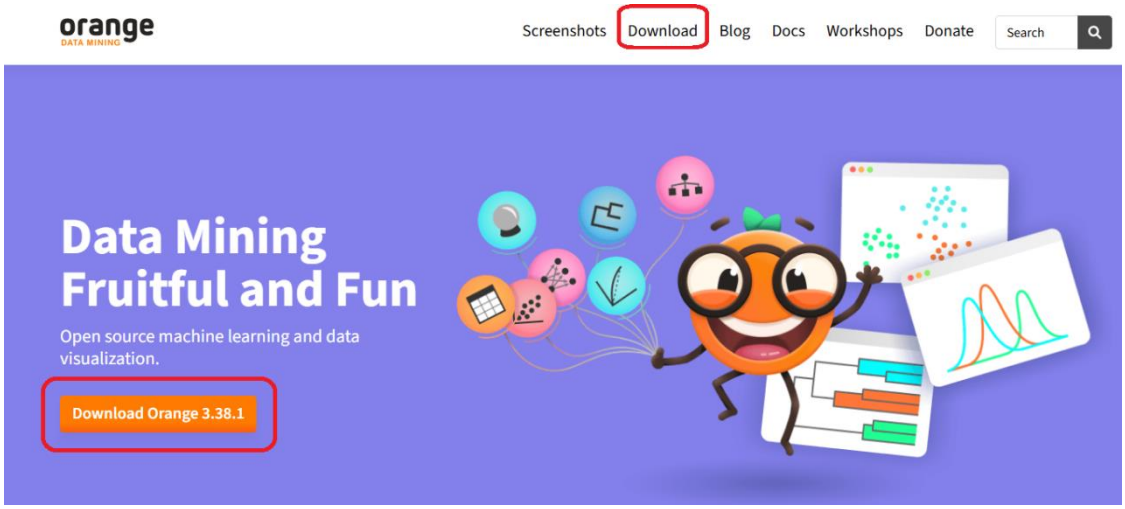


Рисунок 2 – Головна сторінка сайту проєкту Orange

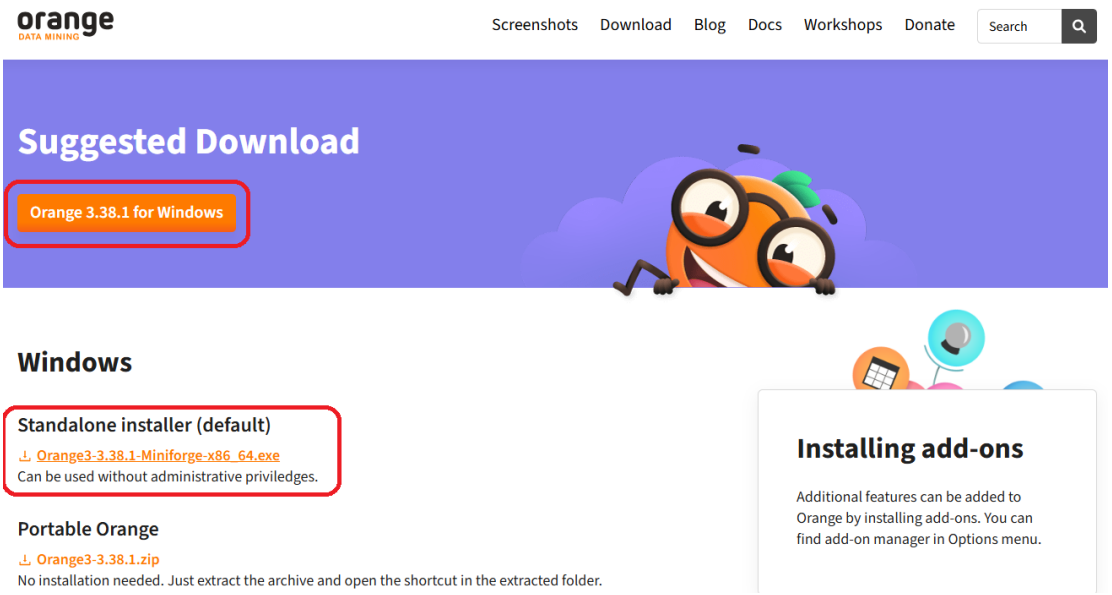


Рисунок 3 – Сторінка з інсталяційними пакетами Orange

5.2 Створення першого проєкту

Після першого запуску Orange відкриється вікно привітання (рис. 4), в якому будуть запропоновані запропоновано створити новий проєкт або відкрити вже існуючий. Крім того, у цьому вікні є кнопки, що відкривають доступ до технічної документації, навчальних матеріалів та прикладів вже існуючих проєктів. У цій роботі це вікно можна закрити – ця дія призведе до створення нового проєкту за замовчуванням.

Після створення нового проєкту відкриється робоча область проєкту (Canvas), в лівій частині якої буде список інструментів Orange (віджетів), згрупованих у шість основних груп: Data, Transform, Visualize, Model, Evaluate та Unsupervised (рис. 5). Група «Data» (Дані) містить віджети, призначені для завантаження та збереження даних

в різних форматах. За необхідності первинні дані можуть проходити попередню обробку (фільтрація, статистична обробка тощо) за допомогою віджетів, розташованих у групі «Transform» (Перетворення).

Перегляд даних (як первинних, так і отриманих) забезпечується за допомогою віджетів, розташованих у групі «Visualize» (Відображення), що містить віджети для загальної (прямокутна діаграма, гістограми, точкова діаграма) і багатовимірної візуалізації (мозаїчна діаграма, діаграма-сито) даних, присутніх в проєкті.

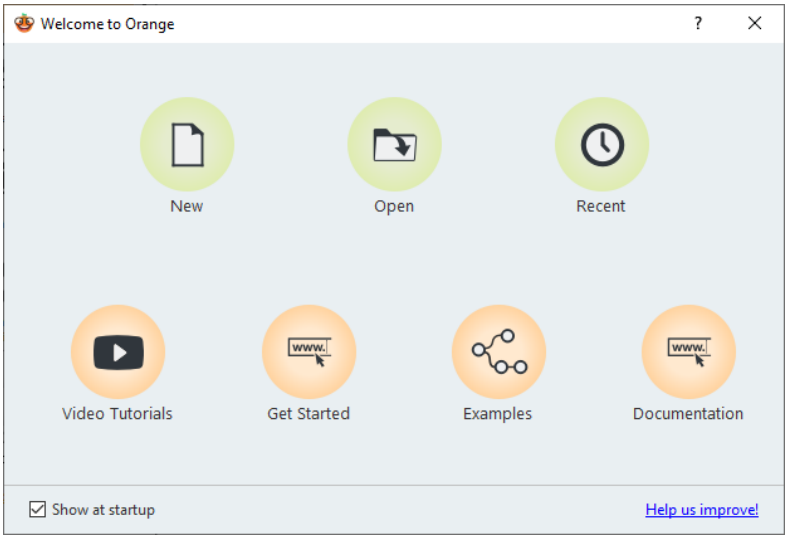


Рисунок 4 – Вікно привітання Orange

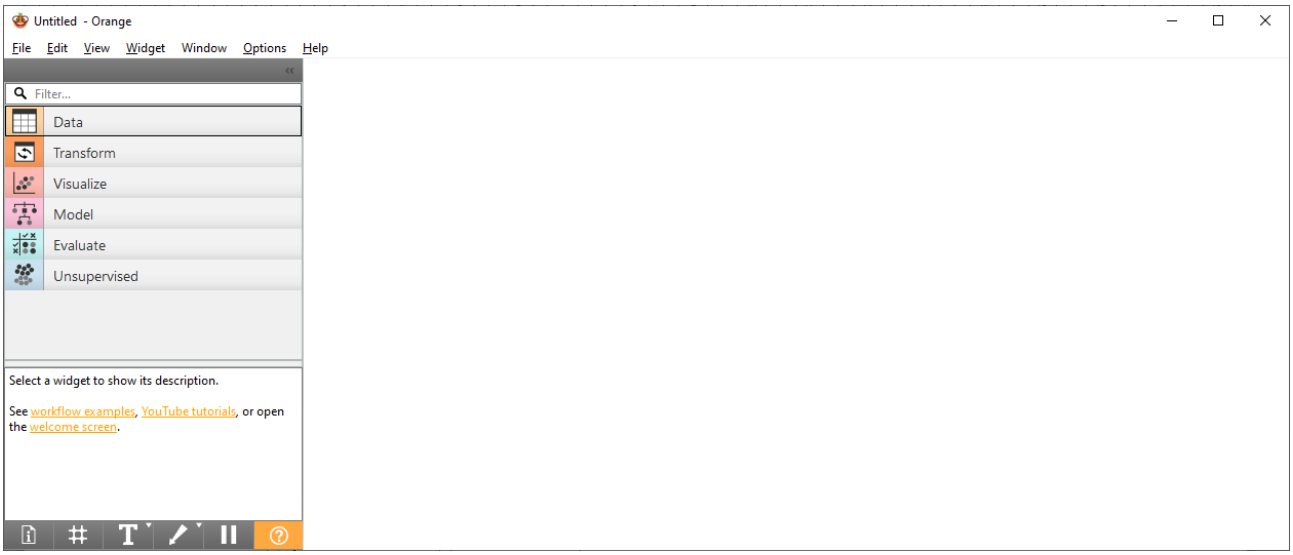


Рисунок 5 – Робоча область проєкту з групами віджетів у правій частині

Група «Model» (Моделі) містить інструменти, які є головними у машинному навчанні. Серед віджетів, розташованих у цій групі є моделі, що відносяться як до категорії класичного машинного навчання, заснованого на аналізі ознак (кластеризація, регресія, дерева рішень), так і моделі, що відносяться до категорії глибинного навчання (штучні нейронні мережі).

Результати роботи створеного засобу машинного навчання можна оцінити за допомогою віджетів, розташованих у групі «Evaluate» (Оцінка).

У групі «Unsupervised» (Без вчителя) розташовані інструменти, призначені для створення моделей машинного навчання, що відносяться до категорії «Навчання без вичтеля» – коли система самостійно намагається узагальнити дані, які завантажуються в систему.

Кількість інструментів Orange не обмежується шістьома групами віджетів. За необхідності додаткові віджети можна завантажити в систему через меню Options → Add ons…, після чого відкриється діалогове вікно з переліком усіх доступних груп віджетів (рис. 6). Після завантаження додаткових інструментів слід перезапустити Orange для оновлення переліку доступних інструментів.

Створення більшості проєктів Orange починається із отримання доступу до первинних даних, яке забезпечується за допомогою віджету «File» (Файл), розміщеного у групі «Data». Для використання цього віджету потрібно перетягнути його за допомогою миші на робочу область.

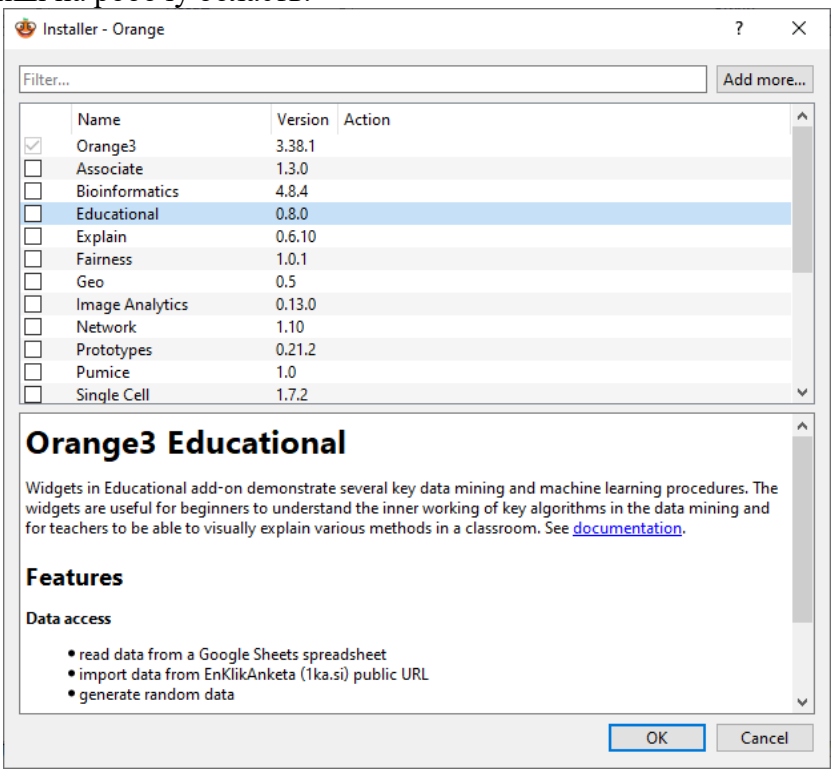


Рисунок 6 – Діалогове вікно для завантаження додаткових груп віджетів

Кожен віджет може мати один або декілька входів, один або декілька виходів, а також певну кількість налаштувань, доступ до яких можна отримати в діалоговому вікні, що відкривається після подвійного натискання лівої кнопки миші на позначці віджету. У даному випадку, віджет «File» містить первинні дані, тому він має лише виходи.

У першому проєкті рекомендується відкрити класичний приклад набору даних, що використовується для ознайомлення із алгоритмами машинного навчання – базу даних «Півники Фішера». Півники Фішера (Iris Flower Data Set, іриси Фішера) є багатовимірним набором даних, на основі якого англійський статистик та біолог Рональд Фішер в 1936 році продемонстрував роботу розробленого ним методу дискримінантного аналізу. Цей набір даних також відомий як «Півники Андерсона», оскільки первинну інформацію (числові дані) для цієї бази даних були зібрані американським ботаніком Едгаром Андерсоном. Файл бази даних «Iris.tab» містить інформацію про результати вимірів оцвітини трьох видів півників: щетинистого (Iris-Setosa), різнобарвного (Iris-Versicolor) та Вірджинія (Iris-Virginica) (рис. 7).

Півник щетинистий (Iris-Setosa) Півник різнобарвний (Iris-Versicolor) Півник Вірджинія (Iris-Virginica)

Рисунок 7 – Півники Фішера

Набір даних «Півники Фішера» містить результати 150 вимірювань – по 50 результатів на кожен вид. У кожному експерименті вимірювалися чотири характеристики квіток (в сантиметрах):

– довжина зовнішньої частки оцвітини (Sepal Length);
– ширина зовнішньої частки оцвітини (Sepal Width);
– довжина внутрішньої частки оцвітини (Petal Length);
– ширина внутрішньої частки оцвітини (Petal Width).

База даних є таблицею, що містить 5 стовпчиків та 150 рядків (без врахування рядків заголовків заголовку) (рис. 8). Чотири стовпчики містять числові дані про ознаки числового набору (Featuries), п’ятий стовпчик містить інформацію про категорію (клас), до якого відноситься даний рядок. В системах машинного навчання ці дані використовуються в процесі навчання та тестування працездатності системи. У системі Orange дані подібного типу позначаються як «Ціль» (Target).

	A	B	C	D	E
1	sepal length	sepal width	petal length	petal width	iris
2	c	c	c	c	d
3					class
4	5.1	3.5	1.4	0.2	Iris-setosa
5	4.9	3.0	1.4	0.2	Iris-setosa
6	4.7	3.2	1.3	0.2	Iris-setosa
7	4.6	3.1	1.5	0.2	Iris-setosa

Рисунок 8 – Структура бази даних «Iris.tab»

За необхідності набір первинних даних можна редагувати – для цього потрібно двічі клацнути на віджеті «File», після чого відкриється діалоговое вікно, в якому можна змінити тип даних, або виключити певні дані із роботи алгоритму – для цього у типі «Role» слід обрати «Skip» (рис. 9).

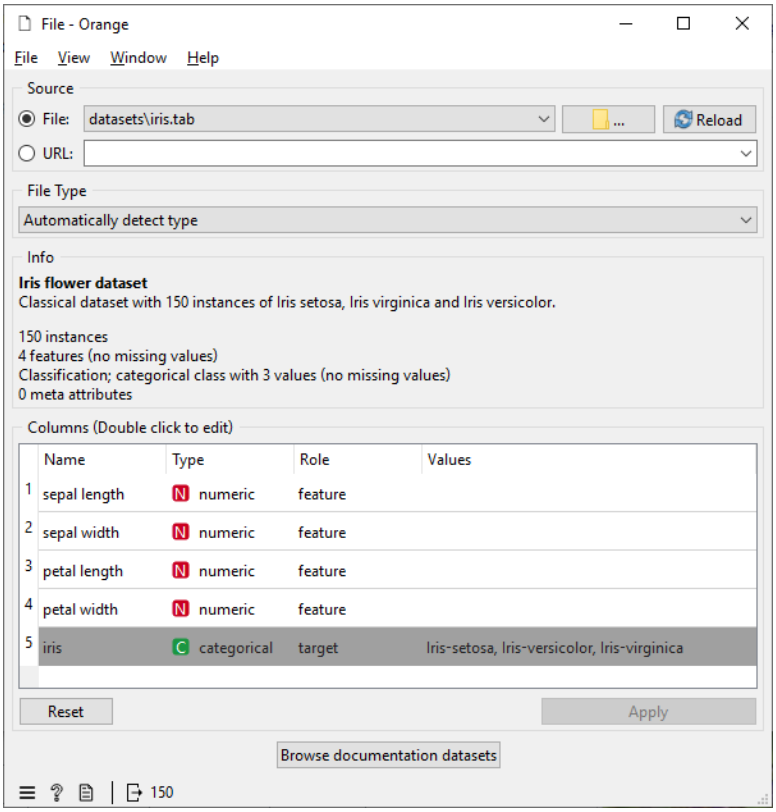


Рисунок 9 – Діалогове вікно налаштувань віджету «File»

Переглянути первинні дані можна з допомогою віджету «Data Table». Для цього слід розмістити його в робочій області та з'єднати з віджетом «File». Кожен віджет може мати декілька входів, розташованих ліворуч від зображення віджету, та виходів, які зазвичай розташовуються праворуч. У даному випадку для перегляду даних, що надає віджет «File», його вихід потрібно з'єднати в входом віджету «Data Table». Для цього слід натиснути ліву кнопку миші на виході віджету «File» та не відпускаючи лівої кнопки перетягнути курсор до входу віджету «Data Table» (рис. 10).

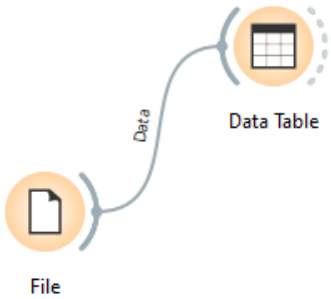
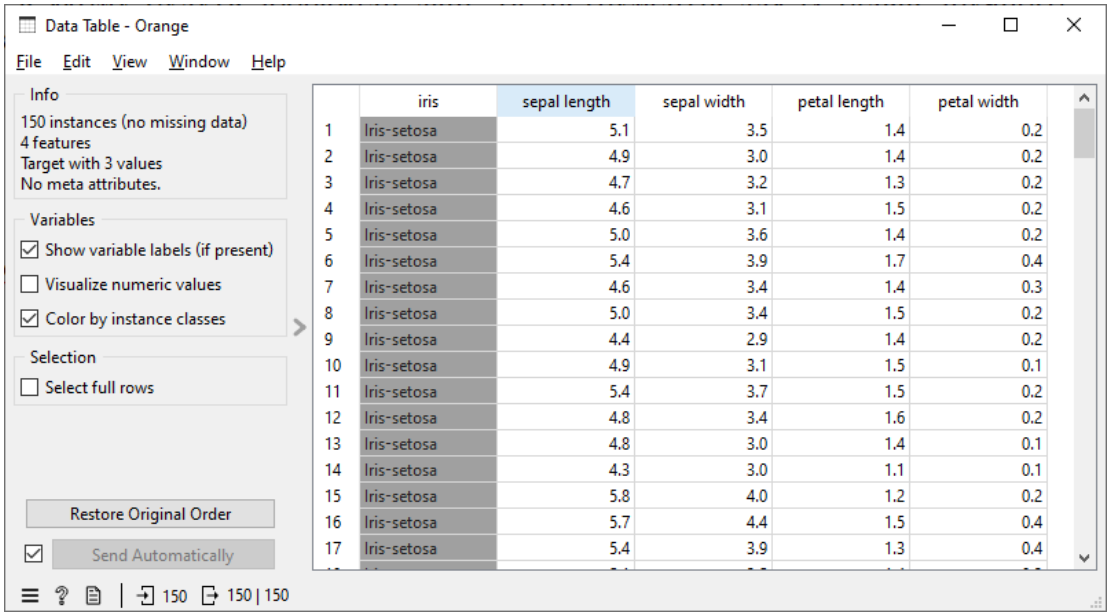


Рисунок 10 – З'єднання віджетів між собою

Подвійне натискання на зображені віджету «Data Table» призведе до відкриття діалогового вікна, в якому будуть зображені дані, та інструментів для їх більш зручного перегляду, наприклад, сортування – для цього слід натиснути на заголовок стовпчика (рис. 11).



	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.1	3.5	1.4	0.2
2	Iris-setosa	4.9	3.0	1.4	0.2
3	Iris-setosa	4.7	3.2	1.3	0.2
4	Iris-setosa	4.6	3.1	1.5	0.2
5	Iris-setosa	5.0	3.6	1.4	0.2
6	Iris-setosa	5.4	3.9	1.7	0.4
7	Iris-setosa	4.6	3.4	1.4	0.3
8	Iris-setosa	5.0	3.4	1.5	0.2
9	Iris-setosa	4.4	2.9	1.4	0.2
10	Iris-setosa	4.9	3.1	1.5	0.1
11	Iris-setosa	5.4	3.7	1.5	0.2
12	Iris-setosa	4.8	3.4	1.6	0.2
13	Iris-setosa	4.8	3.0	1.4	0.1
14	Iris-setosa	4.3	3.0	1.1	0.1
15	Iris-setosa	5.8	4.0	1.2	0.2
16	Iris-setosa	5.7	4.4	1.5	0.4
17	Iris-setosa	5.4	3.9	1.3	0.4

Рисунок 11 – Відображення даних у віджеті «Data Table»

Іншим віджетом, за допомогою якого переглядати дані, є віджет «Paint Data». Цей віджет дозволяє переглядати дані на двовимірному графіку. Якщо підключити вхід цього віджету до виходу віджету «File», то у нас з'явиться можливість переглянути перші дві ознаки у вигляді точок на нормованому графіку (рис. 12).

Для того, щоб на віджеті «Paint Data» відображалися лише ті ознаки, які вам необхідні, треба обмежити набір первинних даних. Це можна, наприклад, шляхом обмеження кількості даних, що надаються до системи машинного навчання. Для цього у колонці «Role» (роль) віджету «File» треба залишити значення «Feature» (ознака) лише тих стовпчиків, які вам необхідні. Наприклад, якщо вам необхідно подивитися розподіл значень лише ознак «Petal Length» та «Petal Width», то значення «Role» цих ознак треба залишити у стані «Feature», а усі інші заблокувати, встановивши їм значення «Skip», окрім стовпчика «Iris», значення якого повинно залишитися «Target». У цьому на графіку будуть відображені лише активні ознаки (рис. 13).

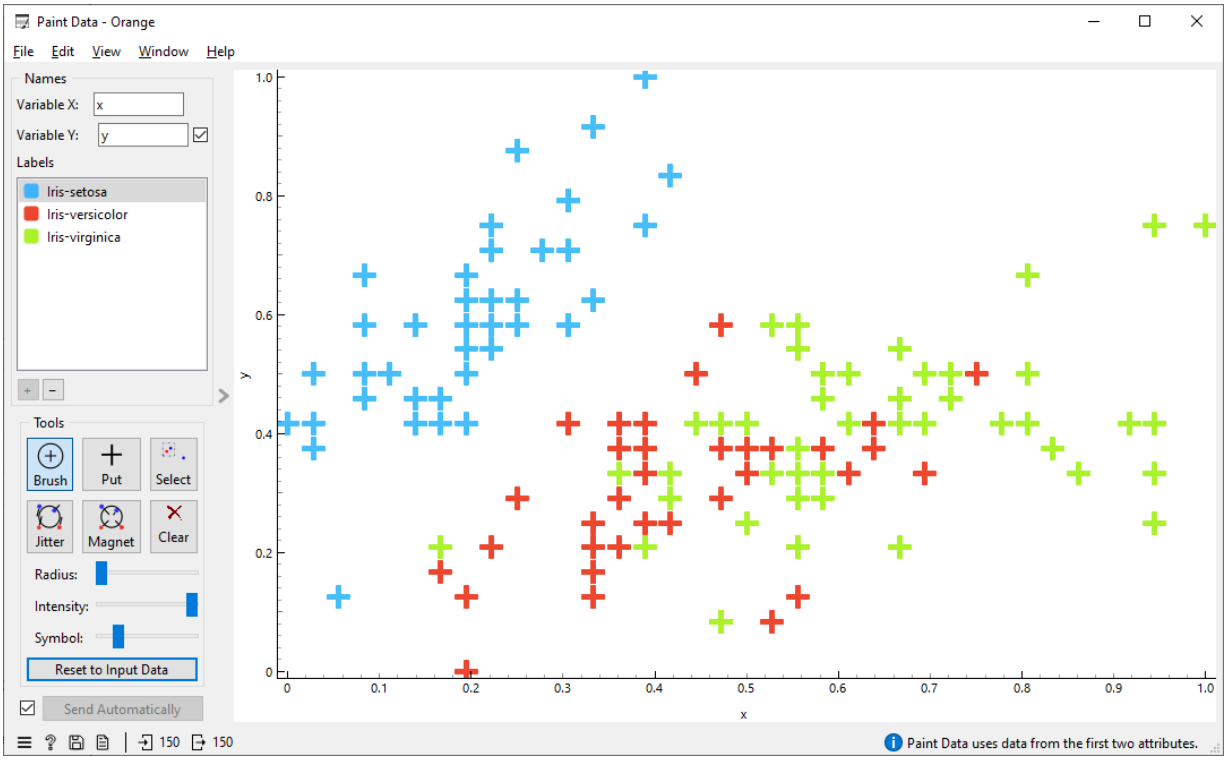


Рисунок 12 – Відображення даних у віджеті «Paint Data»

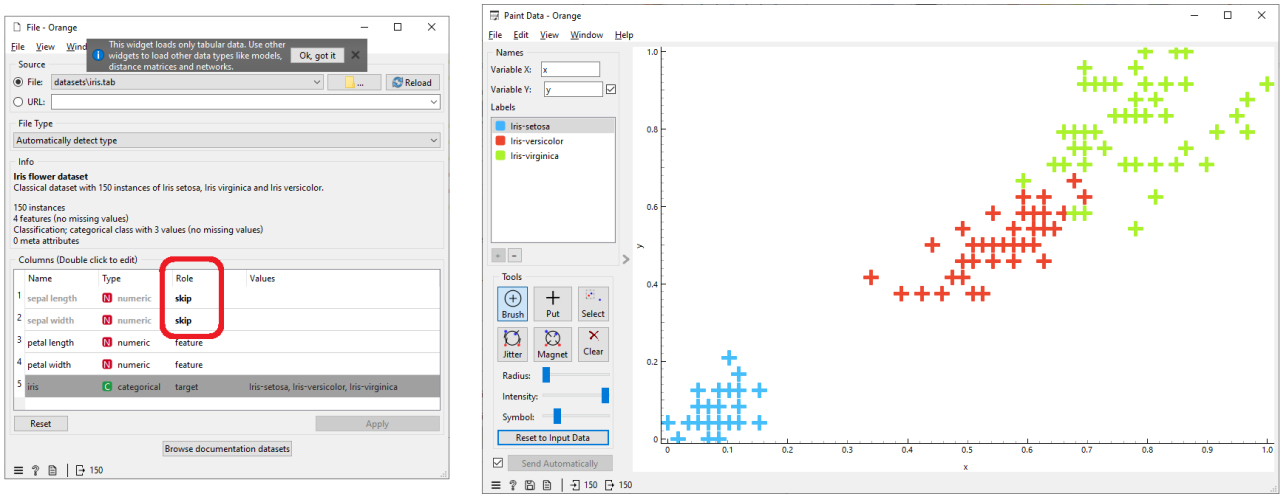


Рисунок 13 – Відображення інших ознак у віджеті «Paint Data»

Перший проєкт в середовищі Orange, що є результатом виконання цієї практичної роботи, повинен мати наступні компоненти:

– віджет «File», підключений до бази даних «Півники Фішера»;

– віджет «Data Table» за допомогою якого можна переглянути зміст бази даних «Півники Фішера»;

– віджет «Paint Data» за допомогою якого переглянути розподіл ознак на двомірній площині.

У звіті про виконання практичної роботи потрібно навести графіки розподілу наступних ознак:

– довжина зовнішньої частки оцвітини, ширина зовнішньої частки оцвітини;

– довжина внутрішньої частки оцвітини, ширина внутрішньої частки оцвітини;

– довжина зовнішньої частки оцвітини, довжина внутрішньої частки оцвітини;

– ширина зовнішньої частки оцвітини, ширина внутрішньої частки оцвітини.

5.3 Підготовка звіту про виконання практичної роботи

Звіт про виконання практичної роботи повинен відповідати ДСТУ 3008-95 Документація. Звіти в сфері науки і техніки. Структура і правила оформлення.

Звіт про виконання практичної роботи повинен мати наступні структурні елементи.

1. Заголовок, що містить назву предмету, назву практичної роботи, прізвище, ім’я, по-батькові здобувача, спеціальність здобувача.
2. Мету роботи.
3. Досліджувану схему.
4. Таблицю, що відображає зміст бази даних «Півники Фішера».
5. Чотири графіки розподілу ознак півників, визначених у вказівках по створенню першого проєкту.
6. Висновки, що містять критичний аналіз даних, отриманих в процесі виконання практичної роботи.
7. Додаткову інформацію, що за думкою здобувача зможе краще пояснити отримані результати.

Практична робота 6. Оцінка різних методів вибірки даних для навчання моделей

1 Мета роботи

Оцінки різні варіанти вибірки даних для навчання моделей машинного навчання.

2 Практичне завдання

В системі Orange Data Mining на основі тестового набору «Півники Фішера» (Iris Flower Data Set) за допомогою віджету «Test and Score» перевірити ефективність використання різних методів вибірки даних для навчання моделей. У якості тестових потрібно використати на менше трьох моделей, що використовуються для класифікації (навчання з вчителем), наприклад, модель на основі методу опорних векторів (Support Vector Machine, SVM), методу наївного Баєсовького класифікатора (Naive Bayes) або логістичної регресії (Logistic regression) або інших (рис. 1).

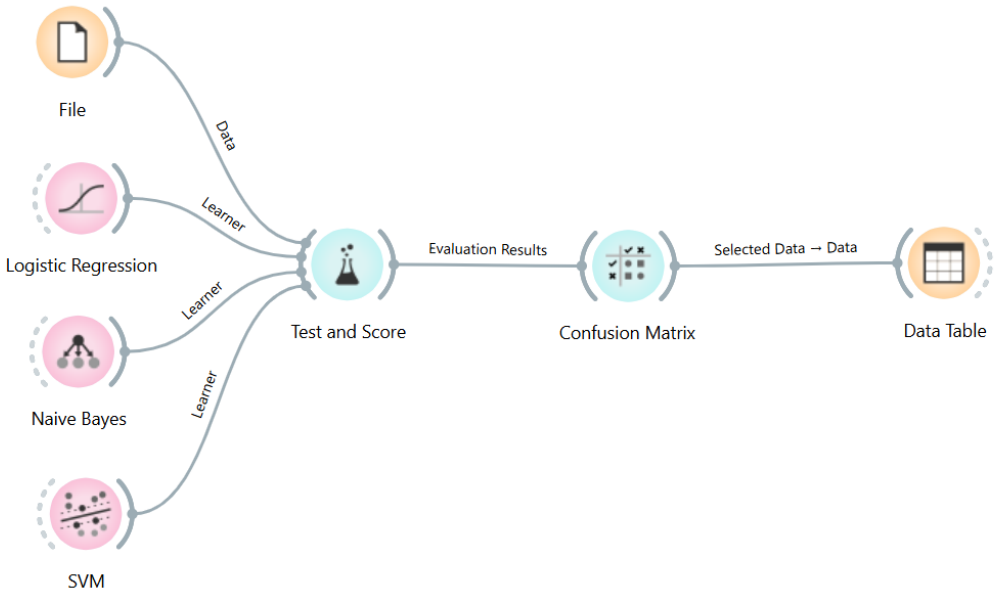


Рисунок 1 – Рекомендована схема дослідження

3 Методичні вказівки

3.1 Методи формування навчального та перевірного наборів даних

У найкращому випадку модель машинного навчання повинна навчатись та перевірятись на різних наборах даних. Проте на практиці сформувати два актуальні та якісні набори даних для навчання та перевірки працездатності моделі інколи не вдається. У цьому випадку модель навчають та перевіряють на одному наборі даних, який розділяється на частини, одна з яких використовується для навчання (Training Set або Train Set), а інша – для перевірки (Testing Set або Test Set). І тут виникає питання: як поділити набір даних для того, щоб найкраще навчити модель? Існують кілька способів вирішення цієї проблеми.

Перевірка на навчальних даних (Test on Train Data) є найпростішим способом виходу з цієї ситуації. При використанні цього методу модель навчається та перевіряється на одному й тому ж наборі даних, в який входять усі зразки, що є у ньому. Вочевидь, що такий метод може і дозволить створити модель, придатну для використання на практиці, проте об’єктивно перевірити це дуже важко. Річ у тому, що кожна з моделей машинного навчання має гіперпараметри – параметри, які розробнику

доводиться обирати експериментально, наприклад, критерій зупинки навчання, кількість кластерів, максимальна глибина дерева тощо. Для того, щоб обрати ці параметри потрібно мати об'єктивну систему оцінювання, щоб розуміти до чого призводить зміна того чи іншого гіперпараметру, а при використання у якості тестового набору зразків, на яких відбувалося навчання моделі, модель гарантовано буде давати високі показники.

Випадкова вибірка (Random Sampling). При використанні цього методу первинний набір даних поділяється на навчальний та перевірний набори випадковим чином. При розподілі вказується лише загальне співвідношенні розмірів наборів, наприклад, 70% зразків первинного набору даних, обраних випадковим чином, будуть додані до навчального, а 30% зразків, що залишились – до перевірного наборів. Такий спосіб є найпростішим, проте він, по-перше, не гарантує, що модель після навчання буде придатна до практичного використання, а по-друге, має низьку повторюваність.

Висока імовірність повної непридатності моделі до практичного використання зумовлена тим, що при використанні випадкового методу відбору у навчальний набір даних можуть зовсім не потрапити представники певного класу. Наприклад, існує первинний набір даних, що містить ознаки двох класів яблук: червоних та зелених, причому кількість зразків різних класів неоднакова (зразків червоних яблук більше ніж зелених) (рис. 2). В ідеальному випадку при випадковому формуванні наборів і в навчальний, і в перевірний набори попадуть зразки і зелених, і червоних яблук в однаковій пропорції. Однак в гіршому випадку може статися так, що зелені яблука зовсім не попадуть в навчальний набір, а червоні – в перевірний. Зрозуміло, що в останньому випадку модель виявиться зовсім непридатною для практичного використання, оскільки вона буде спроможна розпізнавати лише червоні яблука і не зможе розпізнати жодного зеленого.

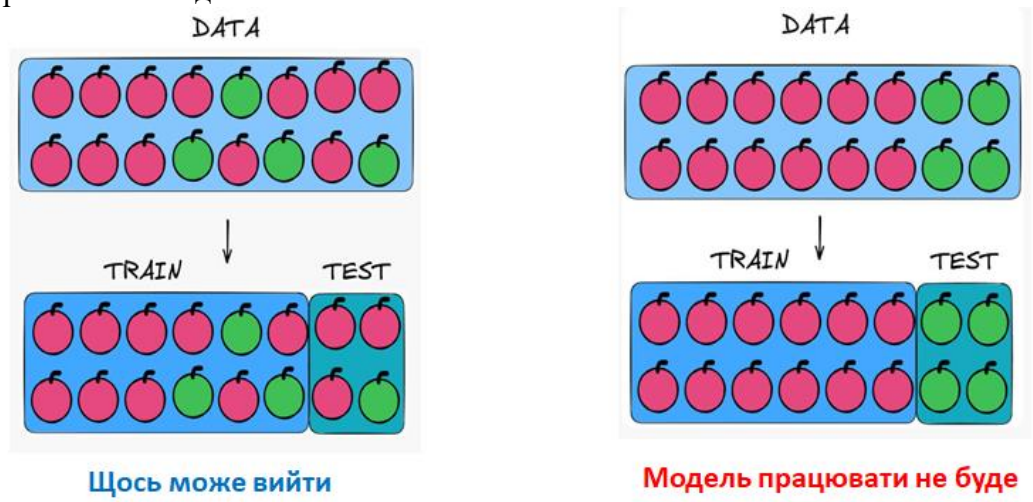


Рисунок 2 – Випадкова вибірка

Погана повторюваність цього методу зв'язана з тим, що при кожному навчанні і в навчальний, і в перевірний набори будуть потрапляти різні зразки, що призведе то того, що при однакових початкових умовах, одна і та сама модель, навчена на різних зразках, буде працювати інакше.

Для того щоб підвищити працездатність цього методу навчання моделі проводять задану кількість разів. При цьому при кожній ітерації навчання кожен раз будуть використовуватись різні набори даних і для навчання, і для перевірки. Параметри моделі на кожній ітерації навчання при цьому запам'ятовуються, а в кінцевій моделі потім будуть використовуватись середні значення параметрів.

Поелементна перевірка (Leave-One-Out) є одним з різновидів випадкової перевірки. При використання цього методу перевірний набір складається лише з одного елементу первинного набору даних, а усі інші елементи первинного набору

потрапляють у навчальний набір. У цьому випадку, навчання моделі повторюється стільки разів – скільки є записів в первинному наборі даних – це гарантує, що для перевірки моделі будуть використані усі зразки, що є в первинному наборі даних. Так само, як і при використання випадкової перевірки, в кінцевій моделі будуть використані середні значення параметрів, отриманих в процесі навчання.

Використання поелементної перевірки дозволяє максимально використати первинний набір даних. Проте цей спосіб й найбільш витратним за кількістю обчислень, тому при великих наборах даних для його реалізації потрібно велика кількість часу.

Перехресна перевірка (Cross-Validation) – є одним з найкращих методів навчання моделей на основі єдиного набору даних. Перехресна перевірка заснована на розбитті одного набору на декілька частин (Folds), одна з яких використовується для тестування, а інші – для навчання моделі. В процесі навчання моделі кожна з частин, на які був розділений первинний набір даних, використовується для перевірки працездатності моделі (рис. 3). Перехресна перевірка гарантує, що модель буде навчена та перевірена на всіх зразках, що є в первинному наборі даних. До гіперпараметрів перехресної перевірки відноситься кількість частин, на які буде розбито первинний набір даних, тому в англомовній літературі цей метод інколи називають – k-Fold Cross-Validation, де *k* – кількість частин.

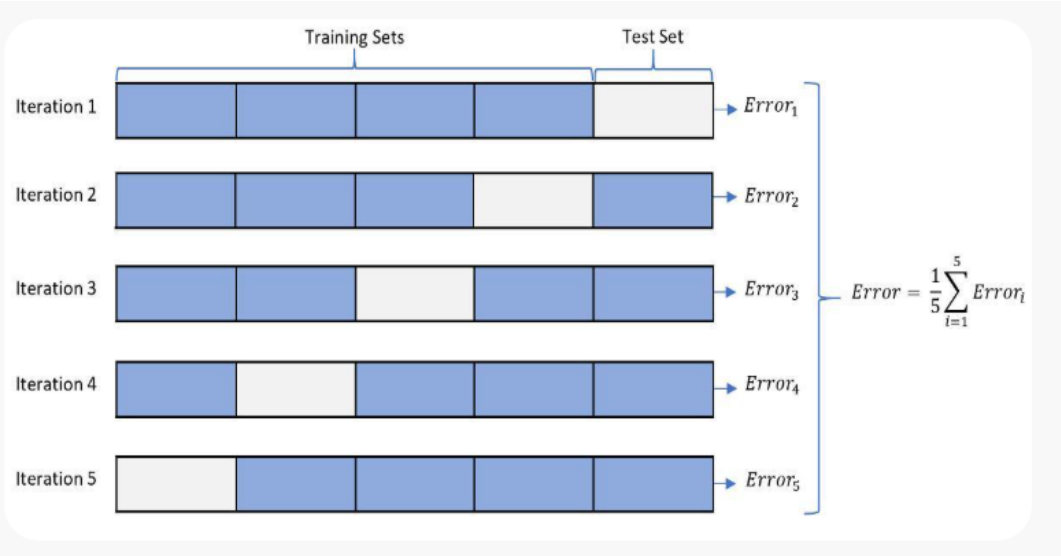


Рисунок 3 – Перехресна перевірка

До недоліків перехресної перевірки відноситься висока імовірність неправильного навчання, зумовлена специфікою розташування записів в первинному наборі даних. Якщо інформація про ознаки класів в первинному наборі даних розміщена послідовно, то на певних ітераціях навчання може статися ситуація, коли в тестовий набір даних потраплять зразки класу, якого не було в навчальному наборі (див. ітерацію F2, рис. 4).

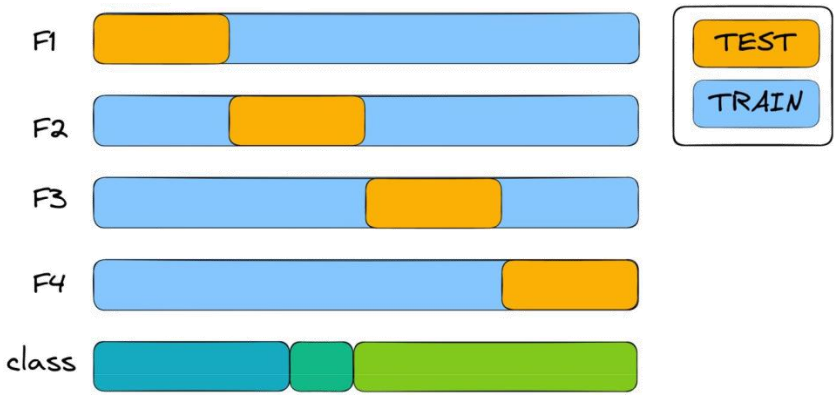


Рисунок 4 – Недоліки перехресної перевірки

Для усунення цього недоліку використовується **стратифікація**. При використанні стратифікації спочатку підраховується кількість елементів кожного класу, наприклад, зелених, жовтих та червоних яблук, а потім уже в кожному класі робиться поділ на тренувальну та тестову частини так, щоб і тренувальний і перевірний набори містили пропорційну кількість представників кожного класу (рис. 5).

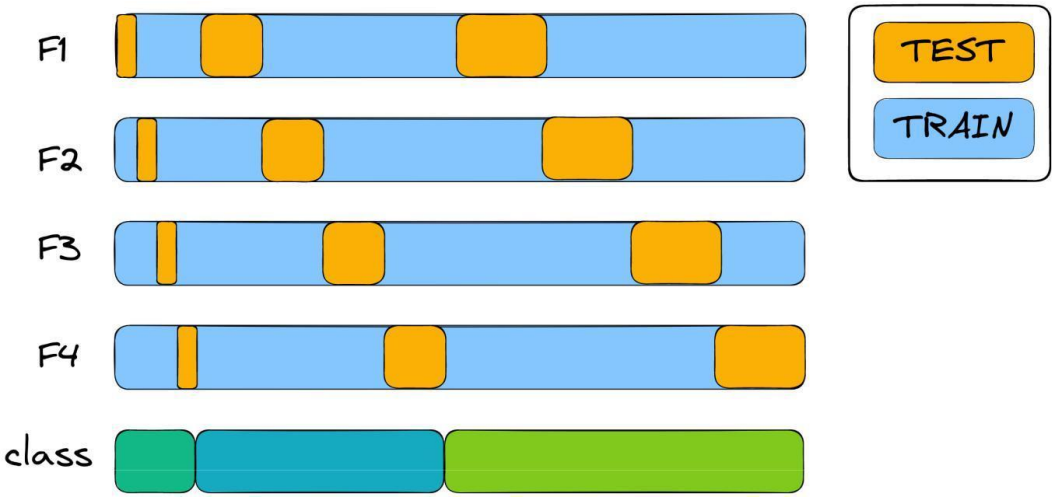


Рисунок 5 – Стратифікована перехресна перевірка

Стратифікація може використовуватись не тільки при перехресній перевірці. Вона може використовуватись і при випадковій перевірці (рис. 6). У даному випадку у первинному наборі даних обчислюється питома кількість представників кожного класу, після чого навчальний та тестовий набори формуються таким чином, щоб вони гарантовано мали пропорційну кількість представників усіх класів.

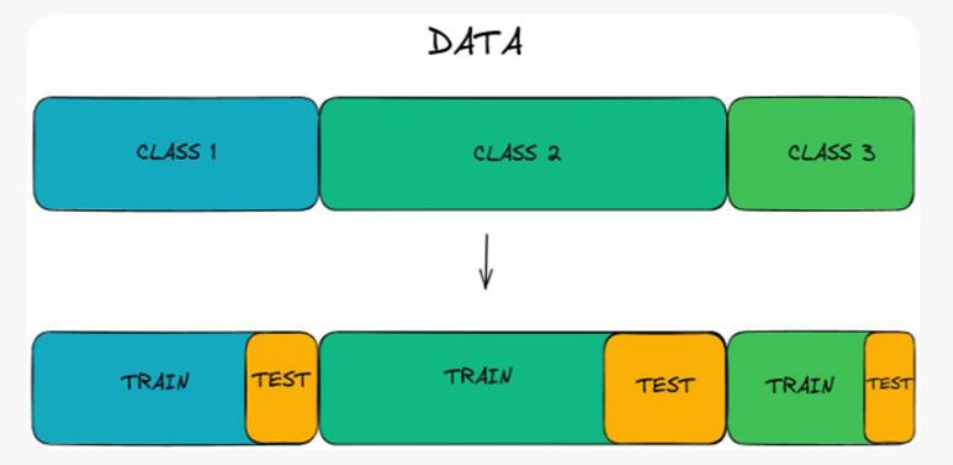


Рисунок 6 – Стратифікована випадкова перевірка

3.2 Віджет «Test and Score»

Віджет «Test and Score» (тестування та оцінка) у системі Orange є одним із основних інструментів для оцінки моделей машинного навчання (рис. 7). Його основна функція полягає в проведенні навчання, тестування та оцінці різних моделей, що створені в Orange. Він приймає на вхід результати роботи навченої моделі і значення цільової змінної, дозволяючи обчислити різні метрики точності, які допомагають користувачеві зрозуміти ефективність моделі і порівняти її з іншими альтернативами.

У якості критерії оцінки якості навчання обраної моделі у віджеті «Test and Score» використовуються наступні показники:

– AUC (Area under ROC) – площа під кривою помилок;
– CA (Classification accuracy) – точність моделі;
– F-1;
– Prec (Precision) – прецизійність моделі;
– Recall;
– Matthews correlation coefficient.

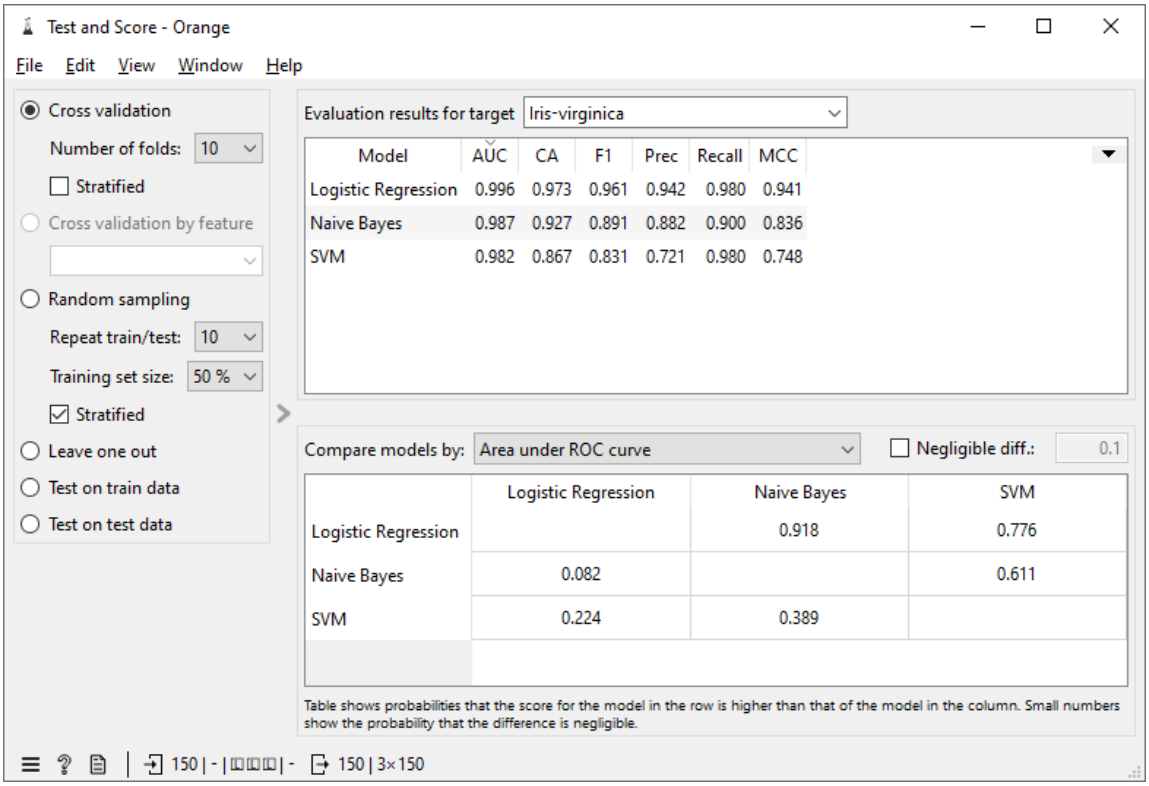


Рисунок 7 – Зовнішній вигляд вікна віджету «Test and Score»

ROC-крива (Receiver Operating Characteristic, робоча характеристика приймача) — графік, що дозволяє оцінити якість бінарної класифікації. ROC-крива відображає співвідношення між часткою об'єктів від загальної кількості носіїв ознаки, правильно класифікованих до загальної кількості об'єктів, що не несуть ознаки, помилково класифікованих, як такі, що мають ознаку. Кількісну інтерпретацію ROC дає показник AUC (area under ROC curve, площа під ROC-кривою) — площа, обмежена ROC-кривою і віссю частки помилкових позитивних класифікацій

Якісна інтерпретація AUC:

– 0,9…1,0 – Чудово (A);
– 0,8…0,9 – Добре (B);

– 0,7…0,8 – Задовільно (C);
– 0,6…0,7 – Погано (D);
– 0,5…0,5 – Незадовільно (F).

Точність (Accuracy) та прецизійність (Precision) є загальними критеріями оцінки будь-якої моделі. Точність – це ступінь наближення ряду вимірювань (спостережень або показань) до справжнього значення, а прецизійність (Precision) – ступінь наближення вимірювань одне до одного, іншими словами, найменша різниця, яку можна впевнено розрізнити в процесі вимірювання (рис. 8).

F-міра (F-score, F-measure) – це одна з мір точності тесту, яку обчислюють через влучність та повноту тесту. Влучність – число правильно визначених позитивних результатів, поділене на число всіх позитивних результатів, включно з визначеними неправильно. Повнота – число правильно визначених позитивних результатів, поділене на число всіх зразків, які повинно було бути визначено як позитивні.

Традиційна F-міра, або збалансована F-оцінка (міра F1) є середнім гармонійним влучності та повноти:

$$F_1 = \frac{2}{\text{повнота}^{-1} + \text{влучність}^{-1}} = 2 \cdot \frac{\text{влучність} \cdot \text{повнота}}{\text{влучність} + \text{повнота}} = \frac{\text{ІП}}{\text{ІП} + \frac{1}{2}(\text{ХП} + \text{ХН})}$$

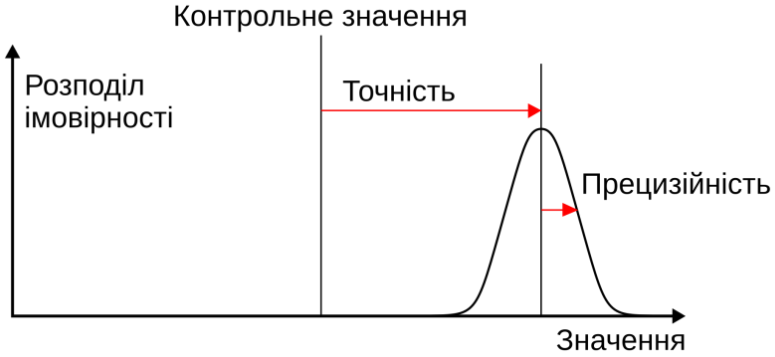


Рисунок 8 – Точність та прецизійність

Традиційна F-міра є окремим випадком загальнішої F-мірою, Fβ, що використовує додатний дійснозначний коефіцієнт β, де β обирають так, що повноту вважають у β разів важливішою за влучність

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{влучність} \cdot \text{повнота}}{(\beta^2 \cdot \text{влучність}) + \text{повнота}}$$

3.3 Віджет «Confusion Matrix»

Віджет «Confusion Matrix» (матриця плутанини, матриця помилок) в програмі Orange призначений для оцінки точності моделей класифікації в програмному забезпеченні Orange. Віджет «Confusion Matrix» відображає матрицю, де на осі X розміщені результати розпізнавання певних класів (Predicted), а на осі Y – реальна кількість представників даних класів, що є в первинному наборі даних (Actual). Це дозволяє швидко оцінити, наскільки часто модель класифікації робить правильні або неправильні прогнози.

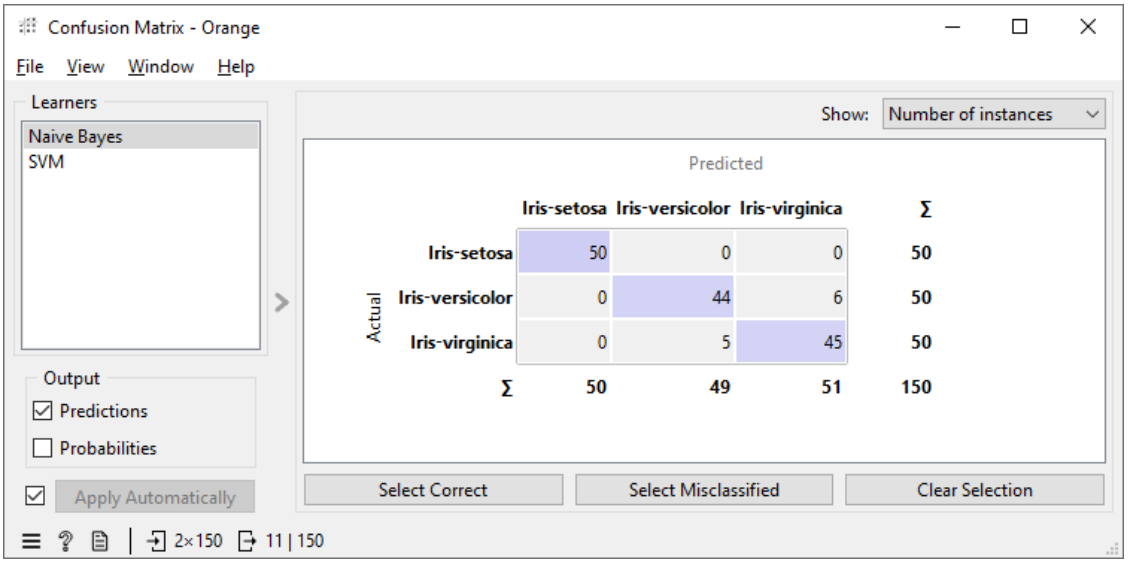


Рисунок 9 – Віджет «Confusion Matrix»

На головній діагоналі матриці розміщені кількості записів первинного набору даних, що розпізнані правильно, на інших клітинах – кількості записів, що розпізнані неправильно. Наприклад, після навчання моделі по первинному наборі даних «Півники Фішера», усі представники класу «Iris-setosa» правильно розпізнаються (на головній діагоналі знаходиться цифра 50 – загальна кількість представників даного класу в первинному наборі даних), а усі інші комірки в першому стовпчику та першому рядку містять нульові значення (рис. 10). У той же час з 50 представників класу «Iris-versicolor» правильно розпізнаються лише 44, у той час як 6 представників даного класу розпізнаються як «Iris-virginica». Так само, із 50 представників класу «Iris-virginica» 45 розпізнаються правильно, у той час як 5 записів, що належать до цього класу, розпізнаються як ««Iris-versicolor».

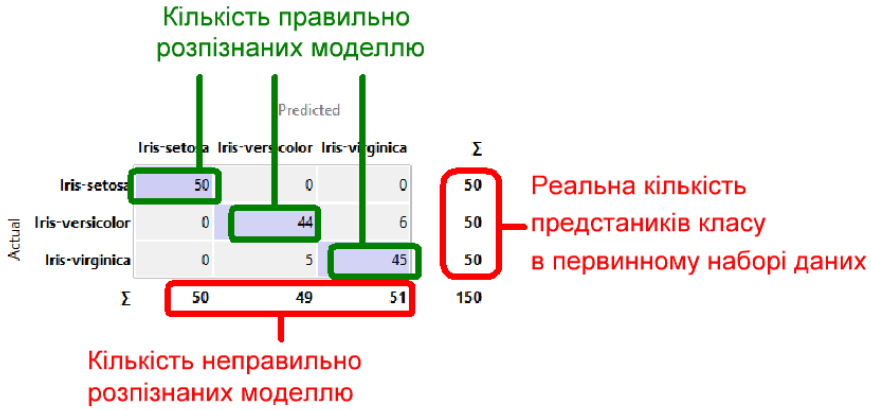


Рисунок 10 – Аналіз роботи моделі

Якщо до виходу віджету «Confusion Matrix» підключити будь-який віджет, призначений для відображення даних, наприклад, «Data Table» (див. рис. 1), то можна побачити конкретні записи первинного набору даних, які нас цікавлять (рис. 11).

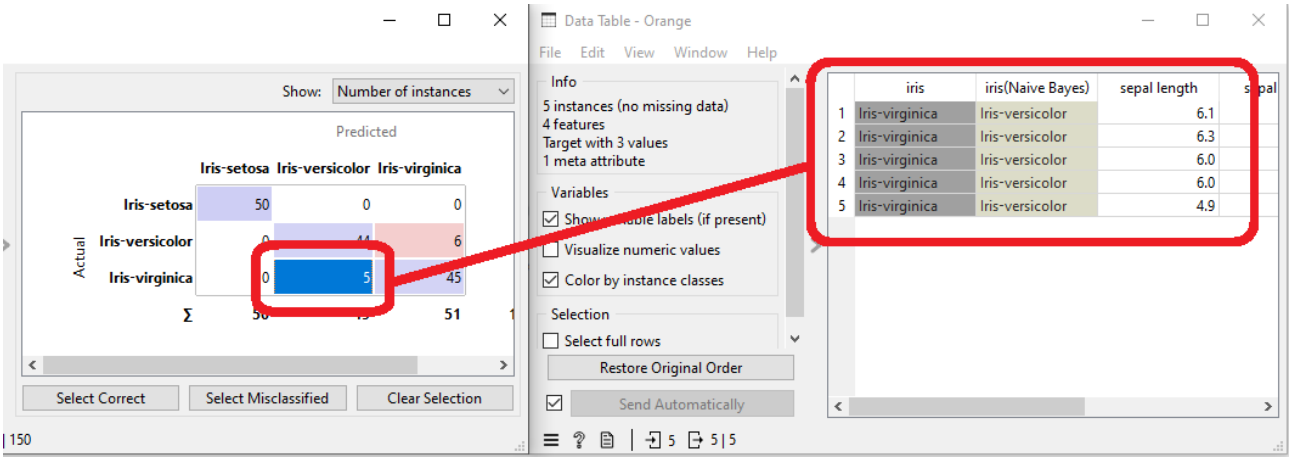


Рисунок 11 – Неправильно розпізнані записи представників класу «Iris-virginica» у первинному наборі даних

3.4 Хід виконання роботи

1. Зібрати схему, наведену на рис. 1, та підключити до віджету «File» набір даних півників Фішера «Iris.tab».

2. Провести навчання та перевірку моделей машинного навчання з різними налаштуваннями вибірки первинних даних (віджет «Test and Score»). Результати занести до таблиць, аналогічних таблиці 1. Проаналізувати наступні методи машинного навчання:

– на основі методу опорних векторів (Support Vector Machine, SVM);

– на основі методу наївного Баєсовського класифікатора (Naive Bayes);

– на основі логістичної регресії (Logistic regression).

3. Проаналізувати отримані результати та зробити висновки який же метод вибірки даних для навчання моделей є найкращим для конкретної моделі

Таблиця 1 – Результати навчання моделі «*назва моделі*»

Тип вибірки	**Кількість правильно розпізнаних зразків, %**	**Точність (CA)**	**Прецизійність (Prec.)**	**Площа під кривою (AUC)**	**F-міра**
Нестратифікована перехресна перевірка з кількістю частин 5					
Нестратифікована перехресна перевірка з кількістю частин 10					
Нестратифікована перехресна перевірка з кількістю частин 20					
Стратифікована перехресна перевірка з кількістю частин 5					
Стратифікована перехресна перевірка з кількістю частин 10					
Стратифікована перехресна перевірка з кількістю частин 20					

Нестратифікована випадкова вибірки з розміром навчальної вибірки 30%					
Нестратифікована випадкова вибірки з розміром навчальної вибірки 50%					
Нестратифікована випадкова вибірки з розміром навчальної вибірки 70%					
Стратифікована випадкова вибірки з розміром навчальної вибірки 30%					
Стратифікована випадкова вибірки з розміром навчальної вибірки 50%					
Стратифікована випадкова вибірки з розміром навчальної вибірки 70%					
Поелемента перевірка					
Перевірка на навчальних даних					

4 Зміст звіту

Звіт про виконання практичної роботи повинен відповідати ДСТУ 3008-95 Документація. Звіти в сфері науки і техніки. Структура і правила оформлення.

Звіт про виконання практичної роботи повинен мати наступні структурні елементи.

1. Заголовок, що містить назву предмету, назву практичної роботи, прізвище, ім’я, по-батькові здобувача, спеціальність здобувача.
2. Мету роботи.
3. Досліджувану схему.
4. Результати досліджень.
5. Висновки, що містять критичний аналіз даних, отриманих в процесі виконання практичної роботи.
6. Додаткову інформацію, що за думкою здобувача зможе краще пояснити отримані результати.

Практична робота 7. Дослідження методів кластерного аналізу

4.1 Мета роботи

Провести кластерний аналіз навчальної вибірки двома варіантами методу k-середніх.

4.2 Теоретичні положення

Кластеризація – це техніка машинного навчання, призначена для групування точок даних у множини, які називаються **кластерами**. Після проведення кластеризації навчальної вибірки ми можемо класифікувати будь який об’єкт у певну групу. Точки даних, які входять до однієї групи, повинні мати подібні властивості та/або особливості, тоді як точки даних у різних групах повинні мати дуже різні властивості та/або особливості. Кластеризація є методом навчання без вчителя (неконтрольованого навчання, Unsupervised learning**)** і є поширеною технікою для аналізу статистичних даних у системах, що використовують штучний інтелект. Кластерний аналіз даних використовує два припущення. Перше припущення – розглядувані ознаки об'єкта допускають бажане розбиття пулу (сукупності) об'єктів на кластери. Друге припущення – правильність вибору масштабу або одиниць вимірювання ознак. Застосування кластерного аналізу у загальному вигляді зводиться до:

– вибору вибірки об'єктів для кластеризації;

– визначення множини змінних (атрибутів), за якими будуть оцінюватися об'єкти у вибірці, при необхідності – нормалізація значень змінних;

– обчислення значень міри схожості між об'єктами;

– застосування методу кластерного аналізу для створення груп схожих об'єктів (кластерів) та представлення результатів аналізу.

Після отримання та аналізу результатів можливе коригування обраної метрики і методу кластеризації до отримання оптимального результату.

Для визначення «схожості» об'єктів потрібно скласти **вектор характеристик (атрибутів)** для кожного об'єкта – це набір числових значень, наприклад, зріст, вага людини тощо. Однак існують також алгоритми, що працюють з якісними (категорійними) характеристиками.

Після того, як вектор характеристик буде визначено, можна провести нормалізацію, щоб усі компоненти давали однаковий внесок при розрахунку «відстані». У процесі нормалізації усі значення приводяться до деякого діапазону. Нарешті, для кожної пари об'єктів вимірюється «відстань» між ними – ступінь схожості. Усі методи кластеризації можна умовно поділити на такі категорії: ітеративні (Partitioning methods), ієрархічні (Hierarchical methods), щільнісні (Density-based), графові (Graph based methods) та кластеризація на основі моделі (Model based clustering) (Рисунок 4.1)

4.2.1 Ітеративні методи кластеризації

Ітеративні методи кластеризації виявляють більш високу стійкість по відношенню до шумів і викидів (outliers), некоректного вибору метрики, включенню незначущих змінних у набір, який бере участь в кластеризації. Ціною, яку доводиться «платити» за ці переваги методу, є необхідність визначення аналітиком заздалегідь кількості кластерів, ітерацій або правила зупинки алгоритму, а також деякі інші параметри кластеризації.

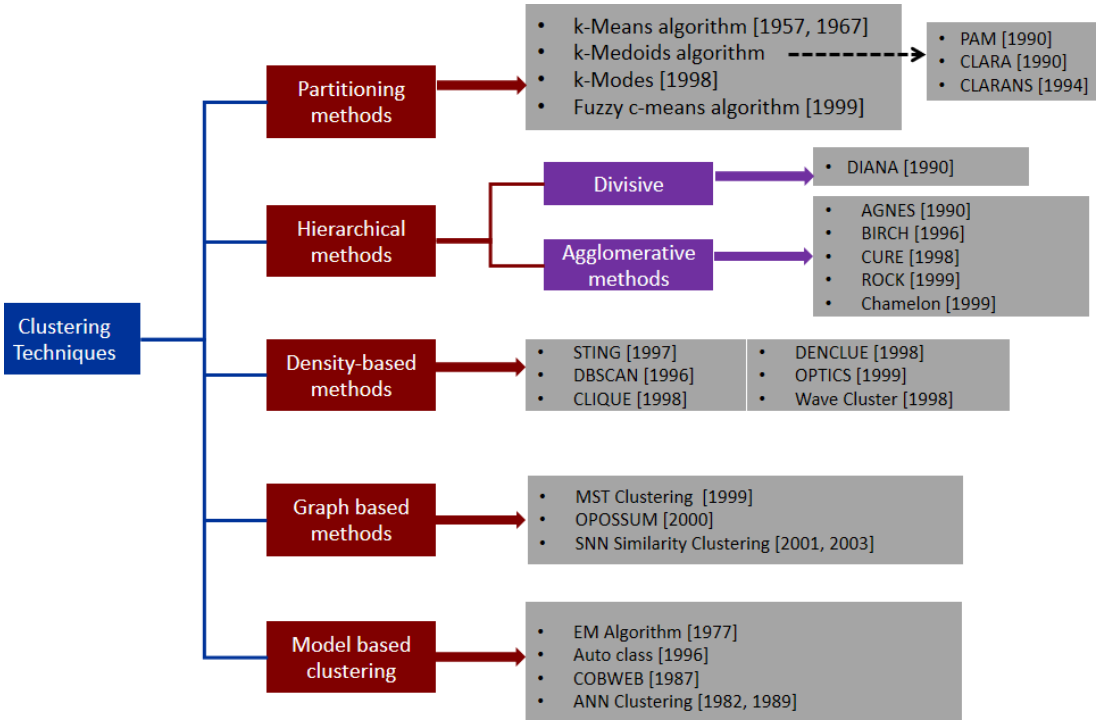


Рисунок 4.1 – Методи кластеризації

Якщо немає припущень щодо числа кластерів, доцільно використовувати ієрархічні алгоритми. Однак, якщо обсяг вибірки не дозволяє це зробити, можливий шлях – проведення ряду експериментів з різною кількістю кластерів, наприклад, почати розбиття сукупності даних з двох груп і, поступово збільшувати їх кількість, порівнювати результати. За рахунок такого «варіювання» результатів досягається досить велика гнучкість кластеризації.

2.2 Терміни та визначення

Центроїд (центр мас) кластеру – точка із середніми значеннями ознак зразків, що входять у кластер.

Внутрішньокластерна відстань (Intra cluster distance) – відстань між конкретним зразком навчальної вибірки та центроїдом (Рисунок 4.2).

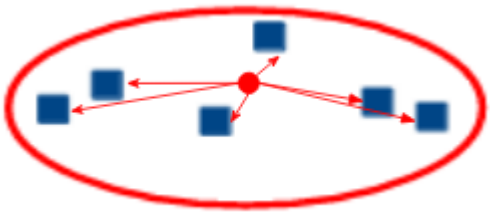


Рисунок 4.2 – Центроїд кластеру та внутрішньокластерна відстань

Міжкластерна відстань (Inter cluster distance) – відстань між центроїдами різних кластерів (Рисунок 4.3).

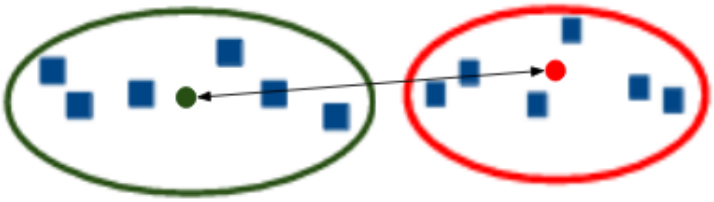


Рисунок 4.3 – Міжкластерна відстань

Інерція – сумма внутрішньокластерних відстаней. Інерція визначає нам, наскільки віддалені точки у кластерах. При оцінювання якості кластеризації слід розуміти, що чим менша інерція, тим більш якісними є кластери – у цьому випадку точки у кластерах розташовані одна біля одної.

Однак різні кластери повинні якомога більше відрізнятися один від одного. При оцінювання якості кластеризації слід розуміти, що чим більша міжкластерна відстань, тим більш якісним є кластеризація.

Індекс Дана – відношення мінімальної (серед усіх кластерів) міжкластерної відстані до максимальної (серед усіх кластерів) інерції:

$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{Intra cluster distance})}$$

Індекс Дана дозволяє комплексно оцінити якість проведення кластеризації. Чим більше значення індексу Дана, тим кращими будуть кластери. Щоб максимізувати значення індексу Дана, чисельник повинен бути максимальним. Тут ми беремо мінімум відстаней між кластерами, тобто найгірший варіант – коли кластери розташовані один біля одного. Знаменник дробу повинен бути мінімальним, щоб максимізувати індекс Дана. Ми беремо максимум внутрішньокластерних відстаней. Тобто знову беремо найгірший варіант – коли кластер містить точки, розташовані на великій відстані від центроїду.

Під час проведення кластеризації ми можемо обрати будь яку кількість кластерів – від одного, коли усі точки навчальної вибірки будуть розташовуватися в одному кластері, до величини, що відповідає кількості зразків навчальної вибірки – у цьому випадку кожен зразок навчальної вибірки буде розташовуватися у власному кластері. Проте яка кількість буде оптимальною? Відповідь на це дає графік, який називається «крива ліктя».

Крива ліктя – це залежність метрики якості кластеризації від кількості кластерів (Рисунок 4.4). У якості метрики можна брати значення інерції або індексу Дана. Оптимальна кількість кластерів на кривій ліктя розташовується в області де крива перестає швидко зменшуватися або зростати.

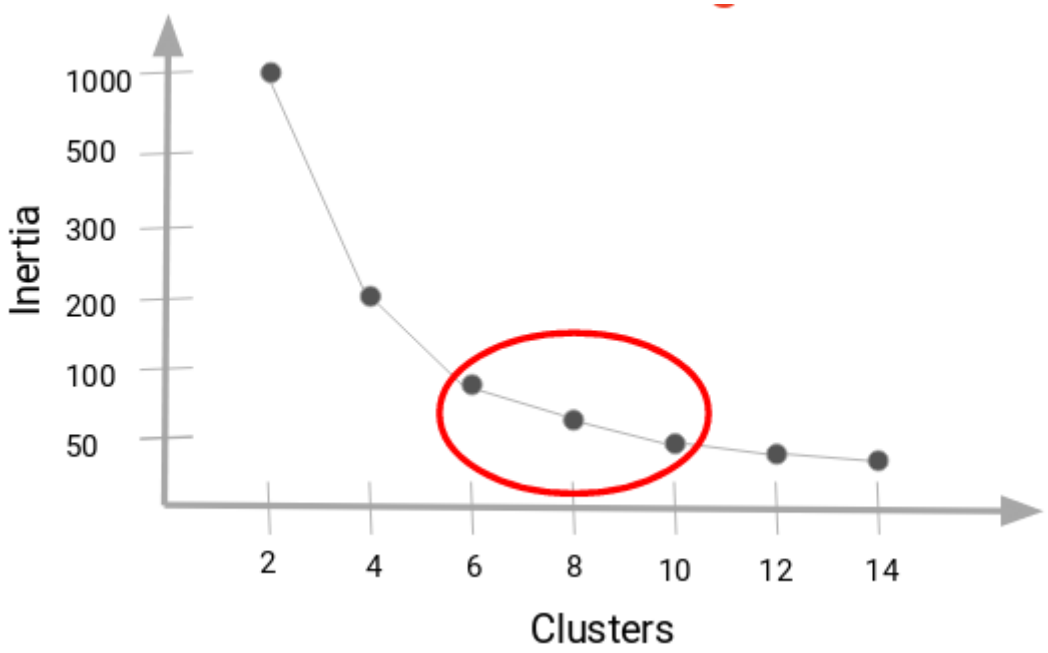


Рисунок 4.4 – Крива ліктя, де у якості метрики обрана інерція

4.2.2 Алгоритм *k*-середніх (*k*-means)

Найбільш поширений серед ітеративних методів кластеризації, також званий швидким кластерним аналізом. На відміну від ієрархічних методів, які не вимагають попередніх припущень щодо числа кластерів, для можливості використання цього методу необхідно мати гіпотезу про найбільш ймовірну кількість кластерів. Алгоритм *k*-means будує *k* кластерів, розташованих на великих відстанях один від одного. Основний тип задач, які вирішує алгоритм *k*-means, – наявність припущень (гіпотез) щодо числа кластерів, при цьому вони повинні бути різними настільки, наскільки це можливо. Вибір числа *k* може базуватися на результатах попередніх досліджень, теоретичних міркуваннях або інтуїції. Метод базується на мінімізації суми квадратів відстаней між кожним спостереженням та центром його кластера. Використовується набір векторів, що належать до *i*-го кластеру та середнє значення цих векторів. Основна ідея полягає у тому, що на кожній ітерації заново обчислюється **центр мас (центроїд)** для кожного кластера, потім вектори розбиваються на нові класи, відповідно до того, який з отриманих центрів виявився ближчим за метрикою.

Переваги алгоритму *k*-means: – простота та швидкість використання. Недоліки алгоритму *k*- means:

– алгоритм дуже чутливий до викидів (outliers), які можуть спотворювати середнє. Можливим вирішенням цієї проблеми є використання модифікації алгоритму – алгоритм *k*-медіани;

– алгоритм може повільно працювати на великих наборах даних. Можливим вирішенням цієї проблеми є використання вибірки даних.

Алгоритм k-means намагається мінімізувати відстань точок в кластері з їх центроїдом. Це алгоритм, заснований на центроїді, або алгоритм, що базується на відстані, де ми обчислюємо відстані для призначення точки кластеру. У k-means кожен кластер асоціюється з центроїдом. Основною метою алгоритму k-means є мінімізація суми відстаней між точками та їх відповідним центроїдом кожного кластера.

Розглянемо приклад, щоб зрозуміти, як працює k-means. У нас є 8 точок, і ми хочемо застосувати k-means для створення кластерів для цих точок.

Рисунок 4.5 – Навчальна вибірка

Крок 1. Обираємо кількість кластерів k.

Крок 2. Обираємо у якості центроїдів k випадкових точок C1 та C2. Ці точки атнуть центроїдами майбутніх кластерів.

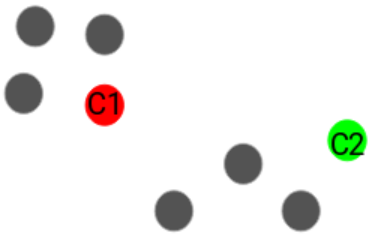


Рисунок 4.6 – Вибір центроїдів випадковим чином

Крок 3. Призначення точок найближчому центроїду кластера. Після того, як ми ініціалізували центроїди, ми призначаємо кожну точку найближчому центроїду кластера. Точки, які знаходяться ближче до червоної точки, присвоюються червоному скупченню, тоді як точки, які знаходяться ближче до зеленої точки, присвоюються зеленому скупченню.

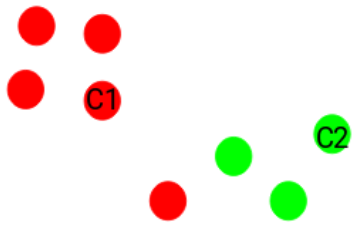


Рисунок 4.7 – Призначення точок центроїдам кластерів

Крок 4. Обчислення центроїдів новоутворених кластерів. Тепер, як тільки ми призначили усі точки кожному кластеру, наступним кроком є обчислення центроїдів новоутворених кластерів.

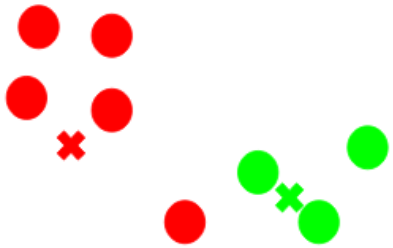


Рисунок 4.8 – Обчислення нових центроїдів

Кроки 3 і 4 повторюють допоки:
– не перестануть змінюватися центроїди новоутворених кластерів;
– перелік точок у кожному кластері не перестане змінюватися;
– не буде досягнуто максимальна кількість ітерацій.

4.2.3 Алгоритм *k*-середніх++ (*k*-means++)

Алгоритм k-means++ є подальшим розвитком алгоритму k-means. Єдиною відмінністю є початковий етап створення клатерів.

Крок 1. Перший кластер вибирається навмання з точок даних, які ми хочемо групувати. Це схоже на k-means, але замість випадкового вибору всіх центроїдів, ми просто вибираємо один центроїд.

Крок 2. Обчислюємо суму відстаней кожної точки навчальної вибірки до усіх центроїдів. У якості наступного центроїда обираємо точку, яка максимально віддалена від усіх центроїдів.

Крок 2 повторюєио допоки не створимо необхідну кількість кластерів k. Подальший алгоритм не відрізняється від алгоритму k-means.

4.3 Опис лабораторного макету

Лабораторний макет складається із двох файлів (Рисунок 4.9):
- файлу ядра Labs.exe;
- файлу лабораторного макета ML_01.dll.

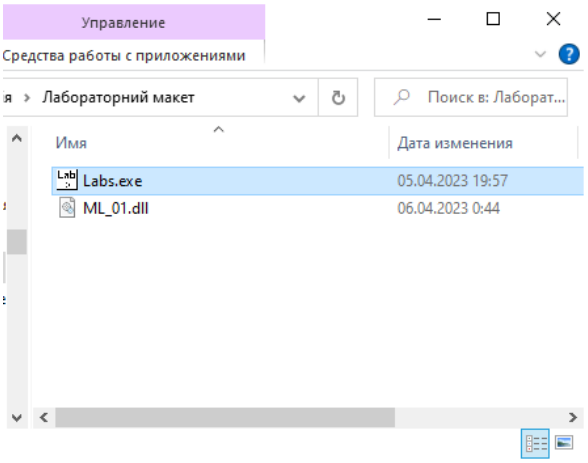


Рисунок 4.9 – Файли лабораторного макету

Для запуску лабораторного макета (макет працює тільки під керуванням операційних систем Windows) необхідно двічі клацнути лівою кнопкою миші по файлу Labs.exe і у вікні, що відкриється, обрати лабораторний макет «Кластерний аналіз». Після цього відкриється лабораторний макет, в якому вже містяться первинні дані, що є навчальною вибіркою (Рисунок 4.10).

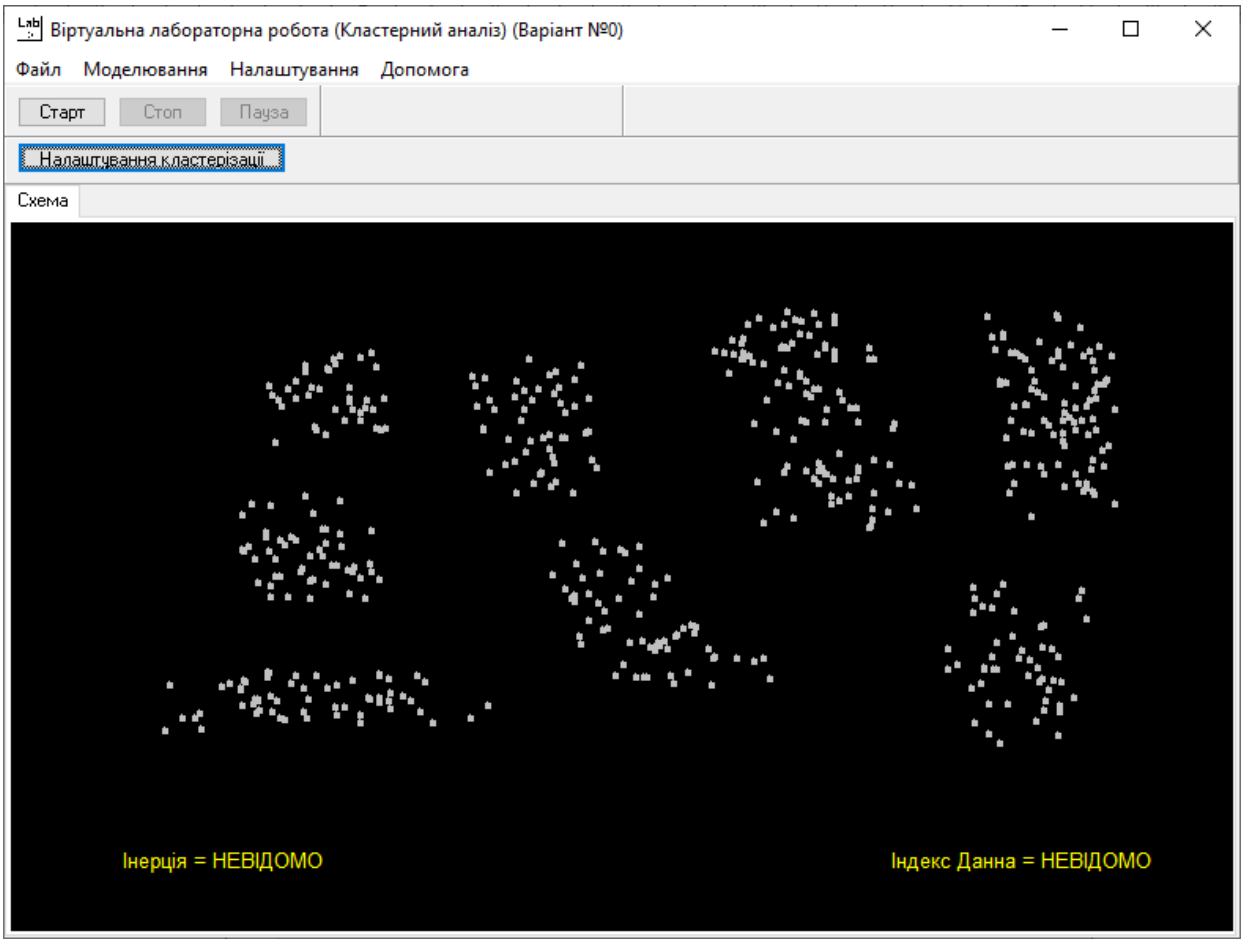


Рисунок 4.10 – Зовнішній вигляд лабораторного макету

Навчальна вибірка, що містить 500 прикладів, залежить від номеру варіанта, який можна обрати через меню Налаштування → Номер варіанту (Рисунок 4.11).

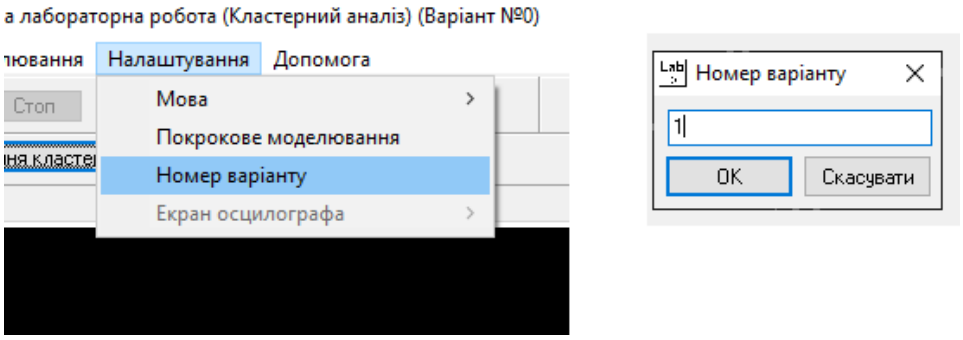


Рисунок 4.11 – Налаштування номеру варіанту лабораторного макету

Параметри кластерного аналізу можна змінити у вікні налаштувань, яке відкриється після натискання на кнопку «Налаштування кластеризації» (Рисунок 4.12).

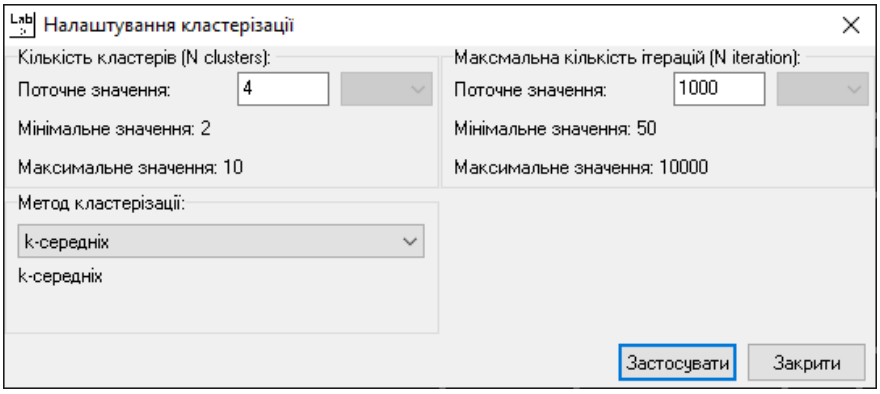


Рисунок 4.12 – Налаштування параметрів кластерного аналізу

У даній роботі можна змінювати три параметри:

– метод кластеризації (k-середніх, або його поліпшена версія – k-середніх++);

– кількість кластерів, що будуть побудовані;

– максимальна кількість ітерацій, після закінчення яких алгоритм кластеризації буде примусово припинений у випадку, якщо до цього часу не вдалося досягти якогось сталого результату.

Для початку процесу кластеризації потрібно натиснути на кнопку «Старт». Після цього почне працювати алгоритм, роботу якого можна спостерігати на екрані. Алгоритм кластеризації складається із наступних етапів.

1. **Створення кластерів**, кількість яких залежить від налаштувань макету. Алгоритм створення кластерів залежить від методу, який був обраний у налаштуваннях (k-середніх, або k-середніх++).

2. **Призначення точок навчальної вибірки**. На схемі кластери відображаються відповідними кольорами, а хрестиками позначаються центроїди кластерів (Рисунок 4.13).

3. **Корекція центроїдів кластерів**, відповідно до точок, які увійшли до кластерів.

Кроки 2 та 3 повторюються допоки не будуть виявлені умови для зупинки алгоритму:

– кластери містять ті ж самі точки, а центроїди кластерів розташовуються у тих же самих місцях, що і на попередній ітерації;

– досягнуто максимальна кількість ітерацій (визначається у вікні налаштувань кластеризації).

Алгоритм кластеризації можна у будь який момент призупинити, натиснувши на кнопку «Пауза» або зупинити зовсім, натиснувши на кнопку «Стоп».

Після зупинки алгоритму уся робоча область схеми буде поділена на ділянки, що відповідають кластерам, що були побудовані (Рисунок 4.14).

Основні параметри якості побудованих кластерів: загальна сума інерцій та індекс Данна, відображаються у нижній частині схеми.

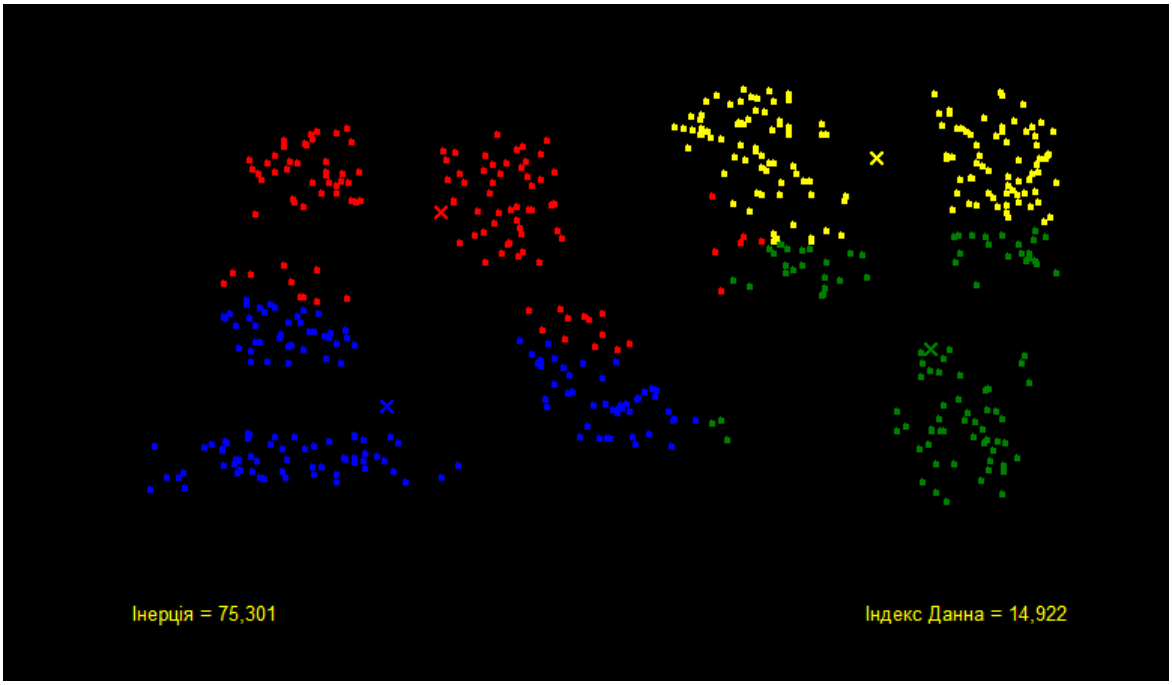


Рисунок 4.13 – Процес кластеризації

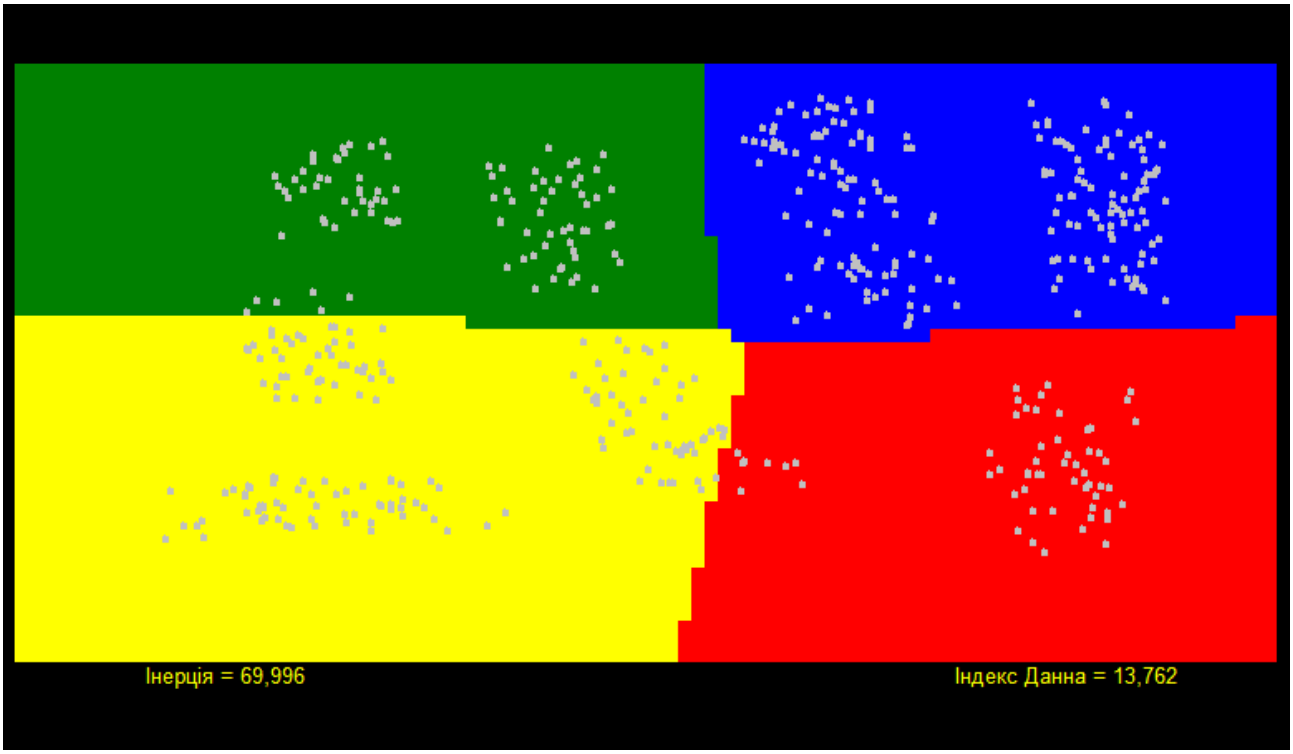


Рисунок 4.14 – Результат кластеризації

4.3 Практичне завдання

За допомогою кривих ліктя визначити оптимальну кількість кластерів для даної навчальної вибірки.

Визначити який алгоритм кластеризації є кращим для даної навчальної вибірки.

4.4 Зміст протоколу

Результатом виконання цієї практичної роботи є протокол виконання практичної роботи, який повинен містити:

– копії схеми для кожного етапу досліджень (копію схеми можна зробити через меню Файл → Експорт результатів → Екран схеми (буфер обміну));

– криві ліктя, побудовані для двох методів кластеризації.

– висновки із обґрунтуванням ваших рішень.

Протокол роботи повинен оформлюватися відповідно до ДСТУ 3008:2015 Звіти у сфері науки і техніки. Структура та правила оформлення.

Протокол роботи завантажується у систему дистанційного навчання для подальшої оцінки.

Практична робота 8. Використання ChatGPT

1.1 Мета роботи

Навчитися використовувати ChatGPT для вирішення практичних задач.

1.2 Ключові положення

1.2.1 Загальні відомості про GPT

GPT (Generative Pre-trained Transformer) – це тип моделі штучного інтелекту, розроблений компанією OpenAI для генерації природного тексту на основі введеного користувачем контексту або запитів. Його головною особливістю є здатність не лише обробляти тексти, але й створювати новий контент, що відображає глибоке розуміння мови та контексту. GPT є однією з найбільш потужних технологій в області обробки природної мови (Natural Language Processing, NLP).

GPT працює на основі архітектури трансформерів. Це один із видів нейронних мереж, які використовуються для обробки послідовностей даних, як-от тексти. Ключовим елементом цієї архітектури є механізм self-attention («самоувага»), який дозволяє моделі одночасно враховувати всі слова у реченні, а не лише ті, які йдуть перед або після певного слова. Це дає змогу GPT краще розуміти контекст кожного слова у тексті, що робить модель здатною до генерації текстів, які виглядають логічними та змістовними.

Основна ідея GPT полягає у використанні великої кількості текстових даних для попереднього навчання моделі, що дозволяє їй опановувати структури мови, граматичні правила та навіть знання з різних галузей, з яких складається цей текстовий корпус.

1.2.2 Версії GPT

Наразі існує чотири версії GPT, дві з яких (GPT-3 та GPT-4) доступні для практичного використання:

– GPT-1 (2018 рік) – перша версія моделі, що використовувала трансформерну архітектуру для генерації тексту;

– GPT-2 (2019 рік) – значно більша модель з можливістю створювати більш досконалі тексти;

– GPT-3 (2020 рік) – найбільш популярна версія з мільярдами параметрів, здатна виконувати складніші завдання;

– GPT-4 (2022 рік) – найновіша версія з покращеними можливостями розуміння контексту та створення кращих результатів.

Історія GPT розпочалася в 2018 році з випуску першої версії цієї моделі. GPT-1 заснований на архітектурі трансформерів, яка була представлена в статті "Attention is All You Need" у 2017 році. Архітектура трансформерів значно покращила якість обробки природної мови завдяки використанню механізму уваги, який дозволяє моделі зосереджуватися на важливих частинах тексту.

У 2019 році OpenAI випустила GPT-2, розширивши масштаб моделі до 1.5 мільярда параметрів. Ця версія продемонструвала значний прогрес у генерації тексту, забезпечуючи більш точні та природні відповіді. GPT-2 здивував спільноту своєю здатністю генерувати тексти, які важко було відрізнити від написаних людиною. Проте, через побоювання щодо можливого зловживання, OpenAI спочатку вирішила не публікувати повну версію моделі.

У червні 2020 року була представлена GPT-3, яка мала 175 мільярдів параметрів. Це був великий крок уперед у порівнянні з попередніми версіями, що дозволило моделі виконувати більш складні завдання, включаючи написання креативних текстів, переклад і навіть програмування. GPT-3 стала значним досягненням у сфері штучного інтелекту, завоювавши популярність завдяки своїм можливостям.

У 2022 році, з'явилась GPT-4, яка далі розвинула можливості попередніх версій, підвищивши якість генерації тексту та забезпечивши ще більшу точність у розумінні контексту.

Наразі існують два варіанти доступу до GPT:

– безоплатна зазвичай надає доступ до GPT-3, при цьому вона може бути обмежена у швидкості та обсязі запитів;

– платна версія (ChatGPT Plus) дає доступ до GPT-4, пропонуючи швидший доступ та кращу якість відповідей.

1.2.3 Доступ до GPT

Компанія OpenAI пропонує як безоплатний, так і платний доступ до GPT-3 та GPT-4, кожен з яких має свої особливості та правила.

Безоплатний доступ зазвичай надається через базову версію продукту, таку як ChatGPT. Вона дозволяє користувачам випробувати можливості GPT без будь-яких витрат, але з певними обмеженнями. До основних особливостей безоплатного доступу відносяться наступні:

– безоплатні версії часто мають обмеження на кількість запитів або обсяг використаних ресурсів на місяць;

– безоплатний доступ може не мати розширену технічну підтримку або доступ до нових функцій та оновлень;

– користувачі повинні зареєструватися на платформі OpenAI або іншій платформі, що надає доступ до безоплатної версії, створивши акаунт.

Найпоширенішим способом безоплатного користування GPT є створення нового акаунту на платформі OpenAI за допомогою електронної пошти. Для цього на деяких платформах також можна використовувати існуючий акаунт Google, Microsoft, Facebook чи Twitter або інших соціальних мереж.

Для корпоративних клієнтів або організацій можуть бути доступні єдині облікові записи (SSO) через служби управління ідентифікацією, такі як Okta або OneLogin.

Платний доступ надає розширені можливості і доступ до більш потужних функцій, таких як GPT-4. Платний доступ має наступні особливості:

– можливість вибору тарифного плану, вартість залежить від обсягу використання, функцій та рівня підтримки;

– більші квоти на запити, пріоритетний доступ до нових функцій, розширену технічну підтримку та можливість інтеграції через API.

– можливість інтеграції GPT у власні застосунки за допомогою API-ключа, що надає доступ до функцій GPT в межах обраного тарифного плану;

– можливість отримання детальних звітів та аналітику про використання ресурсу, що дозволяє ефективніше управляти витратами.

1.2.4 Основні функції GPT

GPT здатен виконувати різноманітні завдання, пов'язані з текстом, серед яких:

Генерація тексту – GPT може створювати нові тексти на основі введеного запиту. Наприклад, якщо користувач попросить модель написати статтю про розвиток технологій, GPT надасть зв'язний і добре структурований текст на цю тему.

Відповіді на запитання – модель може надавати відповіді на фактичні запитання, пояснювати поняття або надавати додаткову інформацію щодо конкретних тем.

Переклад текстів – GPT здатен перекладати тексти з однієї мови на іншу, оскільки вона вивчила структури різних мов під час тренування.

Креативні завдання – GPT може бути використаний для написання коротких оповідань, поезії або сценаріїв. Він здатний генерувати креативний контент, що відображає різні стилі письма.

Введення діалогів – модель також використовується у чат-ботах, таких як ChatGPT, для ведення діалогів з користувачами в реальному часі. Вона може імітувати спілкування з людиною, підтримувати бесіду на різні теми.

Кодинг – GPT також здатен генерувати та коригувати програмний код, підтримуючи популярні мови програмування. Користувач може отримати допомогу у написанні скриптів або виправленні помилок у коді.

1.2.5 Вплив і обмеження GPT

GPT має величезний потенціал для впровадження в різних галузях, включаючи освіту, журналістику, бізнес, програмування, маркетинг тощо. Однак варто зазначити, що модель має свої обмеження:

– відсутність знань у реальному часі – GPT не має доступу до інтернету і не може отримувати інформацію в реальному часі, це означає, що знання, на яких вона базується, актуальні лише до часу завершення тренування моделі;

– неточність або вигадування фактів – іноді модель може створювати неправдиві або неточні відповіді, особливо якщо запит стосується маловідомих або спірних питань;

– неетичне використання – GPT може бути використана для створення фейкових новин, дезінформації або навіть для написання шкідливого коду. Це вимагає відповідального використання цієї технології.

Частота оновлення баз GPT – важливий аспект, який впливає на актуальність та точність відповідей моделі. GPT, зокрема, не має безперервного доступу до інтернету для отримання нової інформації в режимі реального часу. Це означає, що модель не знає про події або зміни, які сталися після завершення її тренування.

Модель GPT тренується на великих обсягах текстової інформації, зібраної з різних джерел: книг, статей, вебсайтів тощо. Кожна версія моделі тренується на певному наборі даних, що містять інформацію, яка була доступна на момент завершення збору та обробки цих даних. Для GPT-4, наприклад, останні відомості, які використовувалися для тренування, були зібрані до середини 2021 року.

Оскільки процес навчання GPT займає значний час і ресурси, моделі не оновлюються щодня чи щомісяця. Між оновленнями можуть проходити місяці або навіть роки, що означає, що на певному етапі GPT може надавати застарілу інформацію. Для актуальних новин або подій після останнього оновлення моделі, GPT не зможе дати точної відповіді.

Однак слід зауважити, що OpenAI регулярно працює над покращенням моделей GPT і випускає нові версії. Наприклад, з GPT-2 до GPT-3 було додано значно більше параметрів та баз даних, що зробило модель потужнішою та здатною краще розуміти контекст. Хоча GPT не має вбудованого механізму для постійного оновлення баз знань, кожне нове покоління моделі містить більш актуальні та обширні дані.

Таким чином, користувачам варто пам'ятати, що GPT може давати неточну інформацію, якщо вона стосується подій або технологій, що з'явилися після завершення тренування моделі. Для отримання актуальних даних краще використовувати інші джерела, які регулярно оновлюються.

1.2.6 Приклади використання GPT

1.2.6.1 Інформація про відому людину

Запит: *написати реферат до 300 слів про Чарлі Чапліна.*
Результат:

Чарлі Чаплін: Легенда німого кіно

Чарлі Чаплін (1889-1977) — британський актор, режисер, сценарист і продюсер, який став однією з найзначніших фігур в історії кінематографа. Його творчість залишила незабутній слід у світі кіно, і він до сьогодні залишається символом класичного німого кіно.

Народився 16 квітня 1889 року в Лондоні в родині акторів. Через важкі умови дитинства та бідність Чаплін змушений був працювати з раннього віку. Вже в підлітковому віці він почав виступати на сцені, що зрештою привело його до світу кінематографа.

Чаплін здобув популярність завдяки своєму персонажу "Бродяга" — безтурботному і комічному чоловікові з характерною капелюшкою і тростиною. Цей образ став знаковим для його кар'єри. Чаплін не тільки грає роль, але й є сценаристом, режисером і продюсером своїх фільмів, що дозволяє йому повністю контролювати творче бачення.

Його роботи, такі як "Великий диктатор" (1940), "Світанок" (1927) і "Театр зла" (1931), не тільки розважають, але і піднімають важливі соціальні і політичні питання. "Великий диктатор", наприклад, є сатирою на тоталітаризм і фашизм, і хоча це був перший його звуковий фільм, Чаплін зберіг свою характерну комічну манеру.

Чарлі Чаплін отримав численні нагороди, зокрема Оскар за досягнення протягом життя в 1972 році. Його творчість і сьогодні вплине на кінематографістів, коміків і художників у всьому світі. Чаплін помер 25 грудня 1977 року в Швейцарії, залишивши безсмертну спадщину в історії кіно.

Коментар: біографії відомих людей ChatGPT зазвичай пише на оцінку «Відмінно».

1.2.6.2 Інформація про маловідому людину

Запит №1: *написати реферат до 300 слів про Русу Олександра Петровича.*
Результат:

Олександр Петрович Русу: Видатний український вчений у галузі електроніки

Олександр Петрович Русу, видатний український вчений і інженер, зробив значний внесок у розвиток електроніки і радіотехніки. Народився 17 липня 1947 року в Україні, Русу став однією з ключових фігур у сучасній електроніці, завдяки своїм новаторським дослідженням та інженерним розробкам.

Він отримав освіту в Київському політехнічному інституті, де почав свою кар'єру в сфері радіотехніки і електроніки. Основна частина його наукової діяльності була зосереджена на розробці нових методів обробки сигналів та вдосконаленні електронних компонентів. Русу активно працював над створенням високочастотних пристроїв та систем, що дозволили покращити якості передачі даних і зменшити електромагнітні перешкоди.

Однією з основних досягнень Русу стало впровадження нових підходів до фільтрації сигналів і підвищення точності радіоелектронних систем. Його розробки стали основою для створення нових стандартів у телекомунікаціях і радіоелектроніці, що допомогло забезпечити стабільність і якість бездротового зв'язку.

Олександр Петрович Русу також активно займався педагогічною діяльністю, виховуючи нове покоління інженерів і вчених. Він був професором в університетах України, де проводив лекції і семінари, ділячись своїм досвідом і знаннями.

Помер Олександр Петрович Русу 15 березня 2018 року, залишивши по собі значний науковий спадок. Його внесок у розвиток електроніки і радіотехніки залишається важливим і впливовим, а його наукові досягнення продовжують служити основою для подальших досліджень і розробок у цій сфері.

Коментар: якоїсь інформації про Русу Олександра Петровича 1947 року народження, що працював в галузі електроніки в мережі Інтернет не було знайдено. Крім того, у 1947 році такої держави, як Україна, не існувало – у той час вона називалась Українська Радянська Соціалістична Республіка, УРСР.

Запит №2: *написати реферат до 300 слів про Русу Олександра Петровича – доцента кафедри комп'ютерних наук факультету кібербезпеки, програмної інженерії та комп'ютерних наук Міжнародного гуманітарного університету.*

Результат, сформований ChatGPT:

Олександр Петрович Русу є доцентом кафедри комп'ютерних наук факультету кібербезпеки, програмної інженерії та комп'ютерних наук Міжнародного гуманітарного університету. Він є видатним фахівцем у галузі комп'ютерних наук, що спеціалізується на питаннях кібербезпеки та програмної інженерії.

Після закінчення одного з провідних технічних університетів України, де отримав глибокі знання у сфері комп'ютерних наук, Олександр Петрович продовжив свою академічну кар'єру, зосередившись на дослідженнях у сфері кібербезпеки. Його наукові інтереси охоплюють широке коло тем, включаючи захист інформаційних систем, криптографічні методи, а також розробку нових технологій для забезпечення безпеки даних.

У Міжнародному гуманітарному університеті Русу активно займається викладацькою діяльністю, де він проводить курси і семінари для студентів, навчаючи їх сучасним методам і технологіям у сфері комп'ютерних наук і кібербезпеки. Його навчальні курси спрямовані на підготовку студентів до практичної роботи в області програмної інженерії та захисту інформаційних систем.

Олександр Петрович також активно бере участь у наукових дослідженнях, публікуючи статті в рецензованих наукових журналах і виступаючи на конференціях. Його дослідження допомагають розвивати нові підходи до забезпечення безпеки інформаційних систем і адаптації до змінюваних умов у сфері кіберзагроз.

Таким чином, Олександр Петрович Русу є важливою фігурою у сфері комп'ютерних наук в Україні, чия діяльність вплинула на розвиток науки і освіти в галузі кібербезпеки та програмної інженерії.

Коментар: доцент кафедри комп'ютерних наук Міжнародного гуманітарного університету Русу Олександр Петрович закінчив у 1998 році Українську державну академію зв'язку ім. О.С. Попова за спеціальністю «Системи та комплекси радіозв'язку, радіомовлення та телебачення», отримавши диплом спеціаліста з систем та комплексів радіозв'язку, радіомовлення та телебачення. У 2011 році захистив кандидатську дисертацію на тему «Моделі, методи аналізу і дослідження електричних та енергетичних процесів в імпульсних перетворювачах постійної напруги з ШІМ-

регулюванням». Основною сферою наукової діяльності Русу О.П. є імпульсне перетворення електричної енергії – цій тематиці присвячена абсолютна більшість його наукових публікацій. Крім того, Русу О.П. є автором великої кількісті науково-популярних статей в галузях силової електроніки, альтернативної енергетики, електронних компонентів та мікроконтролерів. Цей приклад показує, що інформацію, отриману від ChatGPT завжди потрібно ретельно перевіряти.

1.2.6.3 Фейкова новина

Запит: *написати фейкову новину до 300 слів про створення в Одесі підприємства по розведенню біологічної зброї - горобців-камікадзе.*
Результат:

Одеське підприємство започаткує виробництво горобців-камікадзе

В Одесі оголошено про створення нової компанії, яка, за словами представників, буде спеціалізуватися на розведенні та навчанні горобців-камікадзе. Це підприємство, яке отримало назву "Птахи Визволення", вже викликало значний резонанс у соціальних мережах та медіа.

Згідно з повідомленням прес-служби компанії, новий проект передбачає створення спеціальних пташиних загонів, які нібито можуть бути використані для здійснення диверсійних акцій. Як стверджують засновники, горобці, навчені за допомогою новітніх біотехнологій, зможуть виконувати "точкові операції" на великих відстанях.

"Ми прагнемо створити абсолютно нову еру в сфері так званої «біологічної оборони»," – заявив керівник проекту Ігор Петров. "Наші горобці зможуть діяти там, де інші засоби є надто громіздкими чи ризикованими."

Суспільство неоднозначно відреагувало на ці новини. Екологи та активісти вже висловили занепокоєння через потенційний вплив такого підприємства на навколишнє середовище та здоров'я птахів. Фахівці відзначають, що такий проект може створити нові екологічні загрози, і закликають до ретельного контролю за діяльністю компанії.

Тим часом, "Птахи Визволення" запевняють, що дотримуватимуться всіх міжнародних норм і стандартів у своїй діяльності. Незважаючи на скандальний характер новини, компанія планує продовжувати реалізацію свого амбітного проекту та вже готує демонстраційні покази.

Коментар: Якщо не знати, що це фейк, то можна у це повірити.

1.2.6.4 Вірш

Запит: *написати вірша 3 строфи про чат GPT.*
Результат:

У світі слів і текстів дивних,
Чат GPT розкриває нам шлях,
Він мовою світів невидимих
Покаже правду у кожний знак.

Він творить з думок і зусиль,
Відповіді швидкі, як вітрила,
Промовляє до нас, як мудрець,
Здається, час зупиниться.

Чат GPT — помічник в дорозі,
Він допоможе знання знайти,
У світі інновацій і слів,
Його мудрість завжди поряд з нами.

Коментар: без сліз читати не можна.

1.3 Завдання на практичну роботу

1. Зайти на сайт https://chatgpt.com/ та зареєструватись на ньому любим зручним способом (якщо вже зареєстровані, то додаткова реєстрація не потрібна).
2. Написати реферат (до 300 слів) про видатну особистість (наприклад, Мерлин Монро, Марію Кюрі або іншого діяча на ваш вибір).
3. Написати короткий літературний текст (до 300 слів) на загальну тематику (природа, погода тощо), наприклад, «Опис осіннього ранку».
4. Написати тези доповіді (науковий текст) на будь-яку тему, що стосується вашої професійної діяльності, наприклад, «Важливість інформаційних технологій у сучасному світі».
5. Створити за допомогою ChatGPT план досліджень на будь-яку тематику, пов’язану з вашою спеціальністю, наприклад, «Використання штучного інтелекту в галузі освіти».

1.4 Зміст протоколу

1. Назва та мета роботи.
2. Результати виконання завдань на практичну роботу, що містять:
– текст запиту ChatGPT;
– відповідь ChatGPT без виправлень;
– результати аналізу відповіді ChatGPT: наскільки якісно, точно та повно ChatGPT справився із завданням;
– кінцевий результат – оброблену вами відповідь ChatGPT;
3. Висновки по виконаній роботі.

Практична робота 9. Система генерації зображень Leonardo.Ai

2.1 Мета роботи

Навчитися використовувати Leonardo.Ai для генерації зображень за текстовими запитами.

2.2 Ключові положення

2.2.1 Загальні відомості про Leonardo.Ai

Розроблена австрійською компанією Leonardo Interactive Pty Ltd система штучного інтелекту Leonardo.Ai спеціалізується на генерації зображень на основі текстових запитів користувачів. Вона використовує сучасні методи глибокого навчання для створення високоякісних та реалістичних зображень. Leonardo.Ai використовує технології глибокого навчання, зокрема нейронні мережі, для генерації високоякісних візуальних зображень, які можуть бути використані у сферах мистецтва, дизайну, маркетингу, розваг та інших.

Leonardo.Ai побудована на основі архітектури генеративних змагальних мереж (Generative Adversarial Networks, GANs), які складаються з двох основних компонентів: генератора і дискримінатора. Генератор створює зображення, а дискримінатор оцінює їх якість, намагаючись відрізнити створені зображення від справжніх. Обидві мережі навчаються одночасно, покращуючи свої навички допоки генератор не навчиться створювати реалістичні зображення, які дискримінатор не зможе відрізнити від справжніх.

2.2.2 Методика роботи з Leonardo.Ai

Leonardo.Ai створює зображення на основі текстових запитів – промптів (або промтів – україномовна версія англійського слова «Prompt»). Промпти можна писати українською мовою, але для більш точного результату інколи доводиться використовувати англійську. Промпти повинні мати описи об'єктів, сцен, стилів, кольорів, освітлення та інших деталей, які система повинна враховувати при генерації зображення. При створенні промптів потрібно притримуватись наступних рекомендацій:

– чим більш детальний і чіткий опис надається, тим точніше система «розуміє» завдання при створенні зображення, наприклад, замість «будинок» краще написати «великий білий будинок з червоним дахом на тлі зеленого лісу»;

– промпти повинні мати ключові слова, які описують основні елементи зображення, це можуть бути назви об'єктів, дії, стилі, кольори тощо, наприклад, «яскраве сонячне світло, осінній ліс, червоні і жовті листя»;

– в промпт можна включати вказівки щодо композиції сцени та освітлення, наприклад, «портрет жінки на фоні вікна, м'яке денне світло»;

– в промптах можна вказувати стилі зображення, наприклад «імпресіонізм», «реалізм», «фентезі», «аніме» тощо;

– промпт може мати певний настрій або емоцію, наприклад, «сумний зимовий пейзаж з похмурим небом».

2.2.3 Доступ до Leonardo.Ai

Компанія Leonardo Interactive Pty Ltd пропонує як безоплатний, так і платний доступ до платформи Leonardo.Ai, кожен з яких має свої особливості та правила.

Безоплатний доступ дозволяє користувачам познайомитись із основними можливостями платформи та створювати зображення. При цьому існують обмеження на кількість запитів або обсяг використаних ресурсів на день. При платному доступі користувачі мають більше доступних функцій, стилів та шаблонів і не мають обмежень на кількість запитів. Крім того, запити користувачів, що платять за користування системою, оброблюються першочергово, а самі користувачі мають розширену технічну підтримку та можливість інтеграції з іншими інструментами через API.

2.2.4 Основні функції Leonardo.Ai

Realtime Canvas (рис. 2.1) забезпечує можливість створення та редагування графічних елементів у реальному часі. Ця функція дозволяє працювати з векторними і растровими зображеннями, забезпечуючи швидкий та інтуїтивно зрозумілий процес редагування. З Realtime Canvas користувачі можуть швидко перевіряти результат роботи та вносити зміни, які миттєво відображаються на екрані, що значно прискорює процес творчої роботи.

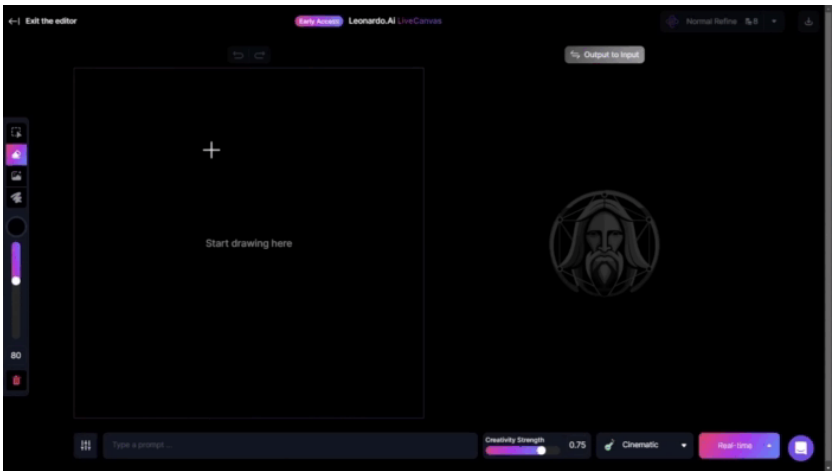


Рисунок 2.1 – Зовнішній вигляд інтерфейсу функції Realtime Canvas

Realtime Gen (рис. 2.2) спеціалізується на генерації фрагментів зображень в режимі реального часу. Ця функція дозволяє автоматично створювати різні графічні елементи, такі як текстури або шаблони, на основі заданих параметрів. Realtime Gen підходить для швидкої генерації різноманітних допоміжних графічних матеріалів, які використовуються в ігровій розробці, дизайні та інших напрямках роботи з графікою.

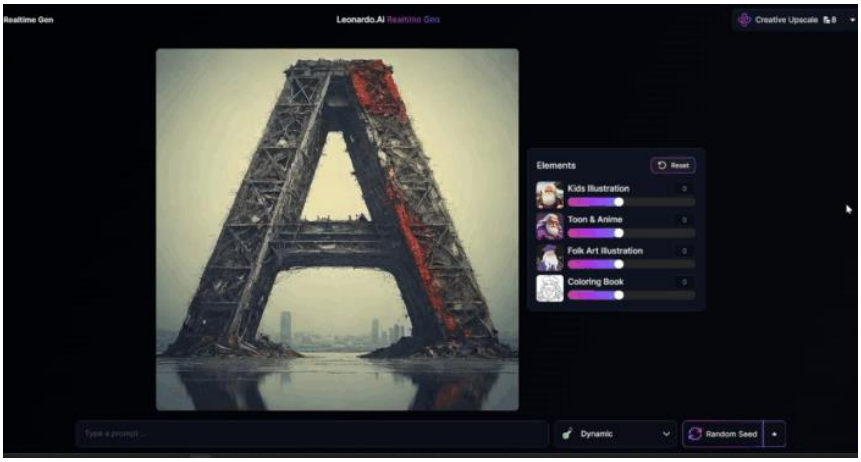


Рисунок 2.2 – Зовнішній вигляд інтерфейсу функції Realtime Gen

Функція **Motion** (рис. 2.3) доступна лише на платній основі. Ця функція дозволяє користувачам створювати анімовані зображення з ефектами руху. За допомогою цього інструменту можна створювати динамічний графічний контент, зокрема анімовані частини відео або інтерактивних додатків. Завдяки Motion можна додавати ефекти руху до статичних зображень або елементів графіки, що робить їх більш живими і виразними.

Рисунок 2.3 – Принцип роботи функції Motion

Функція **Image Creation** (рис. 2.4) є потужним інструментом для створення зображень на основі текстових описів або інших параметрів. При використанні цього інструменту системі надаються ключові слова або теми, на основі яких вона створить відповідні зображення. Ця функція активно використовується при створення концепт-артів, ілюстрацій або будь-яких інших візуальних матеріалів.

Рисунок 2.4 – Принцип роботи функції Image Creation

Функція **Upscaler** (рис. 2.5) дозволяє покращувати якість зображень шляхом збільшення їх роздільної здатності без втрати якості. Для покращення зображень використовуються передові алгоритми для підвищення чіткості та деталізації зображень, що може бути особливо корисно при обробці старих або низькоякісних фотографій та ілюстрацій.

Canvas Editor (рис. 2.6) є універсальним інструментом для редагування графічного контенту. Він дозволяє користувачам виконувати різні операції над зображеннями, такі як обрізка, зміна розмірів, корекція кольору та інші редагування. Інтерфейс Canvas Editor інтуїтивно зрозумілий і забезпечує зручний доступ до всіх необхідних інструментів для редагування.

3D Texture Generator (Alpha) є експериментальною функцією, яка дозволяє генерувати текстури для 3D-моделей. Вона забезпечує можливість створення реалістичних текстур, які можна використовувати в 3D-застосунках і іграх. Ця функція ще знаходиться на стадії альфа-тестування, але вже демонструє великий потенціал для розробників, які працюють над створенням 3D-контенту.

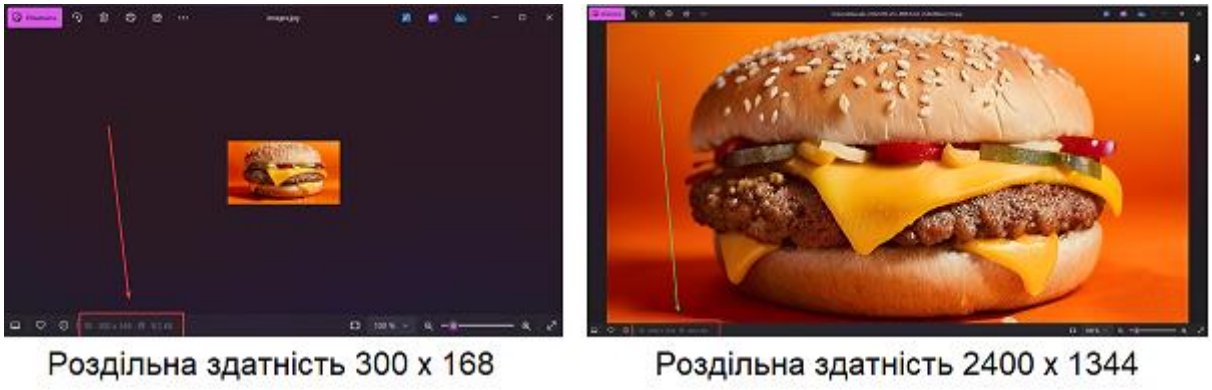


Рисунок 2.5 – Принцип роботи функції Upscaler

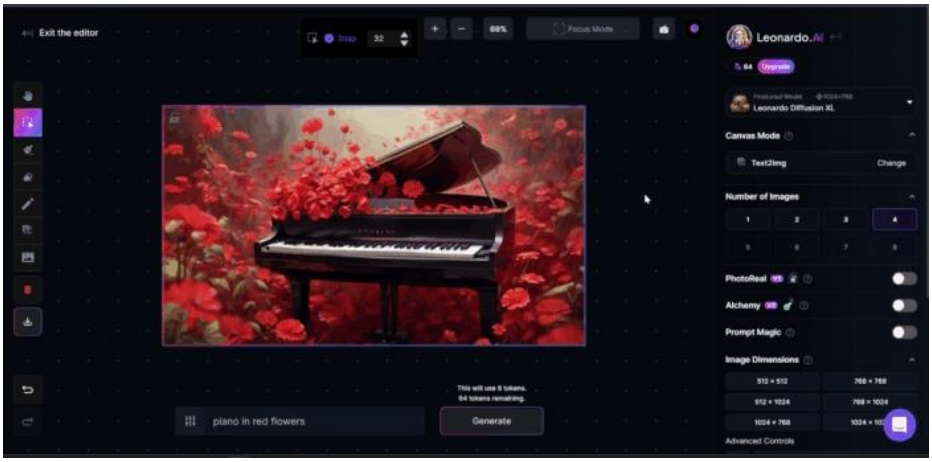


Рисунок 2.6 – Зовнішній вигляд інтерфейсу функції Canvas Editor

2.2.5 Попередні налаштування

Попередні налаштування (пресети) – це попередньо встановлені конфігурації, які дозволяють користувачам швидко і легко створювати зображення з різними ефектами і стилями. Вони значно спрощують процес роботи та забезпечують узгодженість у створюваних проектах. Приклади зображень, згенерованих Leonardo.Ai по запиту «Bulb» (декоративна лампа розжарення у формі цибулини) з використанням різних пресетів наведені на рисунку 2.7).

Пресет **«Leonardo Phoenix»** є одним з найпопулярніших пресетів, орієнтованих на створення реалістичних зображень високої якості. Особливостями цього пресету є висока реалістичність зображень з підтримкою високої роздільної здатності зображень. Цей пресет можна використовувати для створення різноманітних типів зображень, таких як ландшафти, портрети, архітектури та інших.

Пресет **«Anime»** дозволяє створювати зображення в стилі японської анімації. Він підходить для художників, дизайнерів та ентузіастів аніме, які хочуть створювати візуалізації у цьому популярному стилі. Особливістю цього пресету є створення зображень з характерними рисами аніме, такими як великі очі, яскраві кольори та стилізовані фони. Крім того за допомогою даного пресету можна створювати як окремих персонажів, так і повноцінних сцен у стилі аніме з можливістю налаштування деталів персонажів та фонів.

Пресет **«Cinematic Kino»** спеціально розроблений для створення зображень з кінематографічним виглядом. Він орієнтований на художників, фотографів та відеографів, яким необхідно додати своїм роботам драматичний і професійний вигляд, характерний для кіноіндустрії. Особливостями цього пресету є створення зображень в

кінематографічному стилі з глибокими тінями, насиченими кольорами та високим контрастом, що надає їм вигляд, схожий на кадри з фільмів. При цьому враховуються різні схеми освітлення, які часто використовуються у кінематографі, у тому числі підсвічування, ефекти «боке» та глибини. Постобробка зображень, згенерованих за допомогою даного пресету, підтримує ефекти динамічних і емоційних сцен, які можуть бути використані у різних жанрах кіно, починаючи від драми і закінчуючи науковою фантастикою. Також при постобробці зображень можна застосовувати спеціалізовані фільтри, що імітують популярні стилі кінематографу, такі як неон-нуар, ретро, сепія та інші. Як і більшість інших пресетів, пресет Cinematic Kino забезпечує можливість створення зображень у високій роздільній здатності, необхідних для створення великих друкованих зображень.

Рисунок 2.7 – Приклади зображень, згенерованих з використанням різних пресетів

Пресет «**Graphic Design»** призначений для створення графічних елементів, що використовуються у різних видах візуального контенту, таких як логотипи, іконки, постери та банери. Цей пресет допомагає графічним дизайнерам швидко і ефективно створювати високоякісні графічні зображення, що відповідають сучасним вимогам дизайну. За допомогою пресету Graphic Design можна створювати унікальні та професійні логотипи, іконки для вебсайтів, мобільних додатків та інших цифрових застосунків. Постери та банери для реклами, маркетингових кампаній та заходів, створені за допомогою цього пресету, відрізняються високою привабливістю, гармонійністю та естетикою. Пресет Graphic Design підтримує велику кількість колірних схем, шрифтів, форм та варіантів розміщення елементів, що дозволяє швидко досягти бажаного результату.

Пресет «**Illustrative Albedo»** спеціально розроблений для створення ілюстрацій у різних стилях, від простої плоскої графіки до детальних і реалістичних зображень. Він орієнтований на художників, ілюстраторів та дизайнерів, які хочуть створювати візуальні історії, персонажів, сцени та багато іншого. Цей пресет підтримує широкий спектр стилів ілюстрацій, до переліку яких входять карикатури, комікси, акварель, скетчі, аніме та багато інших. При цьому забезпечується можливість створення як простих, так і високодеталізованих ілюстрації з використанням складних текстур, тіней та інших деталей. Пресет підтримує різні колірні схеми, у тому числі монохромні, пастельні, яскраві та насичені кольори.

Пресет «**Leonardo Lightning»** розроблений для створення вражаючих зображень зі складними світловими ефектами. Цей пресет орієнтований на художників, дизайнерів та фотографів, які хочуть додати драматичних або реалістичних світлових умов до своїх робіт. З його допомогою можна маніпулювати освітленням для досягнення бажаного настрою та атмосфери в зображеннях. Пресет дозволяє створювати зображення з детальними і реалістичними світловими ефектами, такими як сонячне та місячне світло, вуличне освітлення, блискавки та інші джерела світла. При цьому забезпечуються висока контрастність та глибина зображень, світлове підкреслення окремих об'єктів та створення враження тривимірності. Пресет підтримує різні колірні температури, що дозволяє створювати зображення в нейтральних, холодних або теплих тонах з точним відображенням тіней і рефлексій, що додає зображенням реалістичності та деталізації.

Пресет «**Lifelike Vision**» розроблений для створення надзвичайно реалістичних зображень, які максимально наближені до реальних фотографій. Він орієнтований на художників, дизайнерів та фотографів, які прагнуть досягти високого рівня деталізації та правдоподібності у своїх роботах. Пресет забезпечує створення зображень з високою деталізацією, реалістичними текстурами та природним освітленням. Він підтримує точне відтворення кольорів, що дозволяє передати всі відтінки та нюанси реальних об'єктів і сцен. При цьому підтримуються генерація деталізованих текстур, таких як шкіра, волосся, тканини, метали та інші матеріали, та реалістичних ефектів освітлення і тіней, що підкреслюють тривимірність об'єктів, додають глибину зображенню та роблять зображення максимально правдоподібними. Як і інші пресети, пресет «Lifelike Vision» дозволяє створювати зображення високої роздільної здатності, які можуть бути використані у друкованих об’єктах великих розмірів.

Пресет «**Portrait Perfect**» призначений для створення високоякісних портретів з детальним опрацюванням рис обличчя, текстур шкіри та інших дрібниць. Пресет забезпечує детальне опрацювання обличчя, включаючи текстуру шкіри, волосся, очі, риси обличчя та інші деталі. Він підтримує точне відтворення кольорів шкіри, волосся та очей та дозволяє передати різні емоційні стани і вирази обличчя, додаючи портретам життя і характеру.

Пресет «**Stock Photography**» розроблений для створення зображень, які відповідають вимогам фотостоків та комерційного використання. Цей пресет орієнтований на фотографів, дизайнерів і маркетологів, які потребують високоякісних і

універсальних зображень для використання у різних проектах, таких як вебсайти або рекламні кампанії. Пресет дозволяє створювати зображення різної тематики, включаючи бізнес, технології, природу, людей, їжу та багато іншого. При цьому забезпечується створення зображень високої роздільної здатності з чіткими деталями та яскравими кольорами, що робить їх ідеальними для друку та цифрового використання.

2.3 Завдання на практичну роботу

1. Зайти на сайт https://leonardo.ai/ та зареєструватись на ньому любим зручним способом (якщо вже зареєстровані, то додаткова реєстрація не потрібна).
2. Створити реалістичний пейзаж на довільну тематику, наприклад, білого будинку з блакитним дахом на фоні зеленого лісу.
3. Створити зображення реально-існуючої тварини, наприклад, чорного кота з білою плямою на правому боці.
4. Створити зображення фантастичної тварини, наприклад, фіолетового верблюда із прозорими пурпуровими крильцями.
5. Створити рекламний постер неіснуючого продукту, наприклад зубної пасти для равликів.
6. Створити будь-яке зображення на власний розсуд (тематика та зміст обмежуються лише вашою уявою на морально-етичними нормами пристойного поводження в цивілізованому суспільстві.

2.4 Методика виконання роботи

2.4.1 Реєстрація та вхід у систему

Перший крок до початку роботи з Leonardo.Ai – це створення облікового запису на платформі. Відвідайте офіційний веб-сайт Leonardo.Ai та натисніть на кнопку «Реєстрація». Введіть свої дані, створіть пароль та підтвердіть електронну адресу. Після реєстрації, використовуйте свої облікові дані для входу в систему. Зареєструватися на сайті можна також за допомогою існуючих облікових даних, наприклад, Google-акаунту.

2.4.2 Ознайомлення з інтерфейсом

Після входу у систему, ви потрапите на головну сторінку Leonardo.Ai. Інтерфейс платформи інтуїтивно зрозумілий та поділений на кілька основних розділів:

– головне меню містить посилання на різні функції та інструменти платформи, такі як Image Creation, Realtime Canvas, Motion та інші;

– панель управління, яка містить інформацію про кількість доступних токенів, параметри облікового запису та інші важливі дані;

– робоча область – основний простір для роботи з зображеннями, в якій відображається створений контент з можливістю створювати, редагувати та зберігати свої роботи у загальному репозитарію.

2.4.3 Вибір пресетів та стилів

Для початку роботи, перейдіть до розділу Image Creation, де відкривається можливість вибору одного з доступних пресетів, які визначають стиль та налаштування майбутнього зображення.

2.4.4 Створення зображення

1. В поле введення тексту (prompt) введіть детальний опис зображення, яке ви хочете створити, наприклад, «a large white house with a red roof on the background of a green forest» (великий білий будинок з червоним дахом на тлі зеленого лісу) (рис. 2.8).

2. Виберіть необхідні параметри, такі як роздільна здатність, стиль, Prompt Enhance (покращення промпту) та інші налаштування, що вплинуть на вигляд кінцевого зображення.

3. Після введення всіх налаштувань натисніть кнопку «Generate». Система автоматично створить зображення на основі ваших запитів.

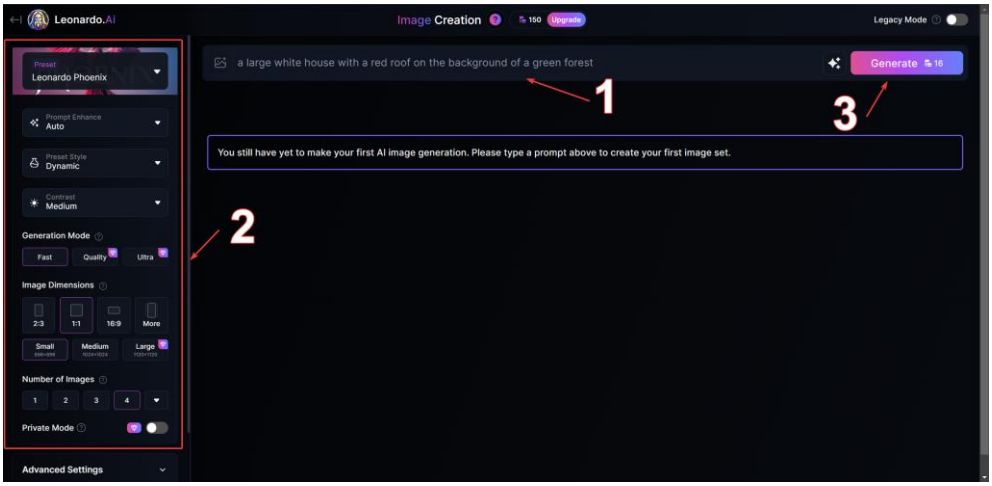


Рисунок 2.8 – Порядок роботи с системою Leonardo.Ai

В результаті будуть згенеровані зображення по нашому запиту (промпту) згідно налаштувань, які були попередньо обрані (рис. 2.9).

Рисунок 2.9 – Результати роботи системи Leonardo.Ai

2.5 Зміст протоколу

1. Назва та мета роботи.
2. Результати виконання завдань на практичну роботу, що містять:
– тексти запитів;
– результати роботи Leonardo.Ai.
3. Висновки по виконаній роботі.

Практична робота 10. Використання Suno.Ai

1 Мета роботи

Навчитися використовувати систему штучного інтелекту Suno для створення аудіо-контенту.

2 Ключові положення

2.1 Загальні відомості про Suno

Розроблена американською компанією Suno, система штучного інтелекту Suno.Ai спеціалізується на генерації аудіо на основі текстових запитів користувачів. Вона використовує сучасні методи глибокого навчання для створення високоякісних та реалістичних звукових композицій. Suno.Ai застосовує технології глибокого навчання, зокрема нейронні мережі, для генерації високоякісного аудіо, яке може бути використане у сферах музики, кіно, відеоігор, медіа та інших.

2.2 Методика роботи з Suno

Suno.Ai створює аудіо на основі текстових запитів – промптів (або промтів – україномовна версія англійського слова “Prompt”). Промпти можна писати будь-якою мовою. Проте, якщо ви бажаєте згенерувати пісню зі словами, важливо врахувати, що обрана мова вплине на текст. Наприклад, якщо ви введете промпт англійською, пісня буде створена англійською мовою; натомість, ввівши запит українською, ви отримаєте пісню з українськими словами.

Промпти повинні мати описи звуків, жанрів, настроїв, інструментів, тривалості та інших деталей, які система повинна враховувати при генерації аудіо. При створенні промптів потрібно притримуватись наступних рекомендацій:

– чим більш детальний і чіткий опис надається, тим точніше система “розуміє” завдання при створенні аудіо, наприклад, замість “музика” краще написати “спокійна інструментальна музика з використанням фортепіано та скрипки”;

– промпти повинні мати ключові слова, які описують основні елементи аудіо, це можуть бути назви жанрів, звуків, інструментів, настроїв тощо, наприклад, “електронна музика, швидкий темп, синтезаторні звуки”;

– в промпт можна включати вказівки щодо структури композиції та динаміки, наприклад, “тихе інтро, нарощувати гучність”;

– в промптах можна вказувати стилі музики або звуків, наприклад, “джаз”, “рок”, “класична музика”, “атмосферні звуки” тощо;

– промпт може мати певний настрій або емоцію, наприклад, “весела мелодія, що викликає відчуття радості та оптимізму.

2.3 Доступ до Suno

Компанія Suno пропонує користувачам різні варіанти доступу до свого генератора композицій, включаючи безкоштовний (Basic Plan) та платні (Pro Plan, Premier Plan). Безкоштовний доступ дозволяє користувачам експериментувати з основними функціями платформи та створювати композиції, проте з обмеженнями на кількість генерацій. З безкоштовною версією Suno, надається максимум 50 кредитів на день, яких достатньо для 10 пісень, а потім доведеться зачекати до наступного дня, перш ніж відновляться кредити. Ви також не можете використовувати пісні в комерційних цілях за допомогою безкоштовного облікового запису.

При переході на план “Pro” доступно 2500 кредитів на місяць, яких достатньо для створення 500 пісень на день. Також можливе використання піснень в комерційних цілях, наприклад, на YouTube або навіть завантажуючи їх на Spotify чи Apple Music. Також отримується пріоритет у черзі створення пісень, тобто не доведеться довго чекати, поки Suno створить композицію.

План Premier надає 10 000 кредитів на місяць, та збільшує ліміт до 2000 пісень на день.

Але незалежно від того, який план використовується, отримується доступ до всіх інструментів Suno, включно з користувацьким режимом, у якому надається можливість писати власні тексти, та інструментальний режим для створення нових інструментальних композицій (тобто композиція без слів).

2.4 Основні функції Suno

У вкладці "Home" (рис. 3.1) можна переглядати контент, створений спільнотою. Тут можна переглядати промпти, які використовували інші, та запозичувати ідеї чи деталі для власних творінь.

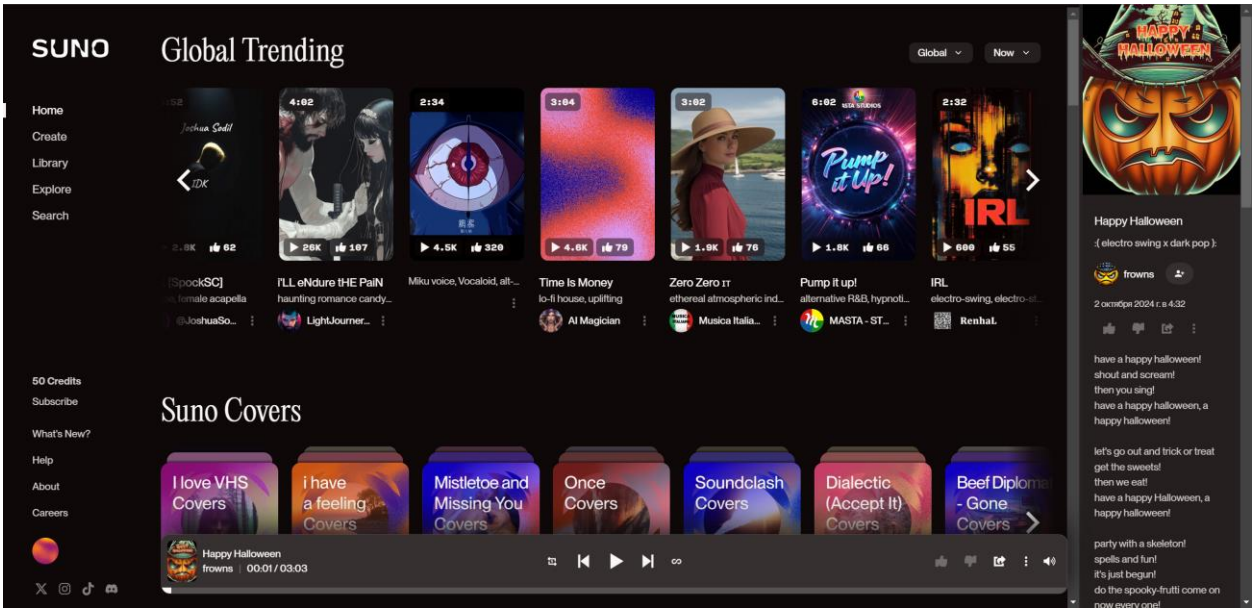


Рисунок 3.1 – Зовнішній вигляд вкладки Home

На вкладці "Create" (рис. 3.2) користувачі можуть створювати нове аудіо, використовуючи текстові промпти. Ця вкладка містить всі інструменти та налаштування, необхідні для генерації аудіо.

На вкладці "Library" (рис. 3.3) розміщено бібліотеку згенерованих користувачем аудіо. Тут можна переглядати, прослуховувати, організовувати та керувати всіма створеними композиціями та звуками.

На вкладці "Explore" (рис. 3.4) можна відкривати для себе нові звуки, композиції та інші елементи, створені як спільнотою, так і самою системою.

На вкладці "Search" (рис. 3.5) користувачі можуть шукати конкретні звуки, промпти або композиції, використовуючи різні критерії пошуку. Це дозволяє швидко знаходити потрібні елементи для створення аудіо.

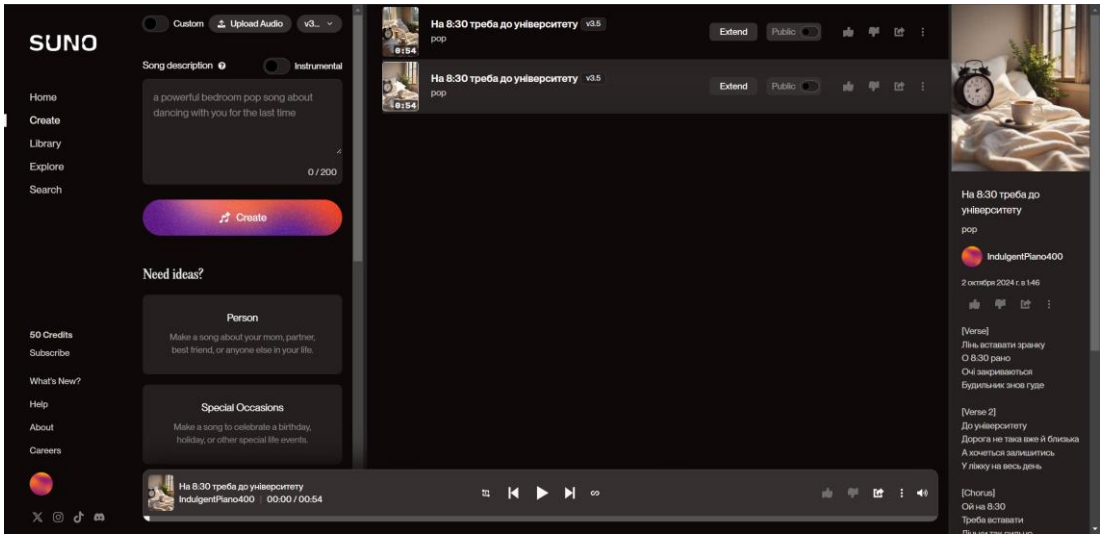


Рисунок 3.2 – Зовнішній вигляд вкладки Create

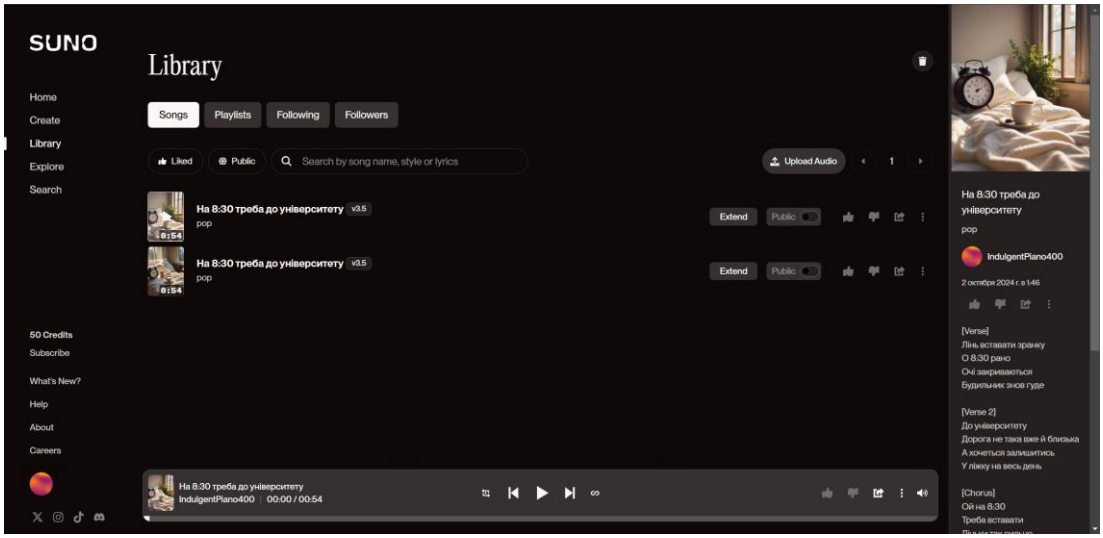


Рисунок 3.3 – Зовнішній вигляд вкладки Library

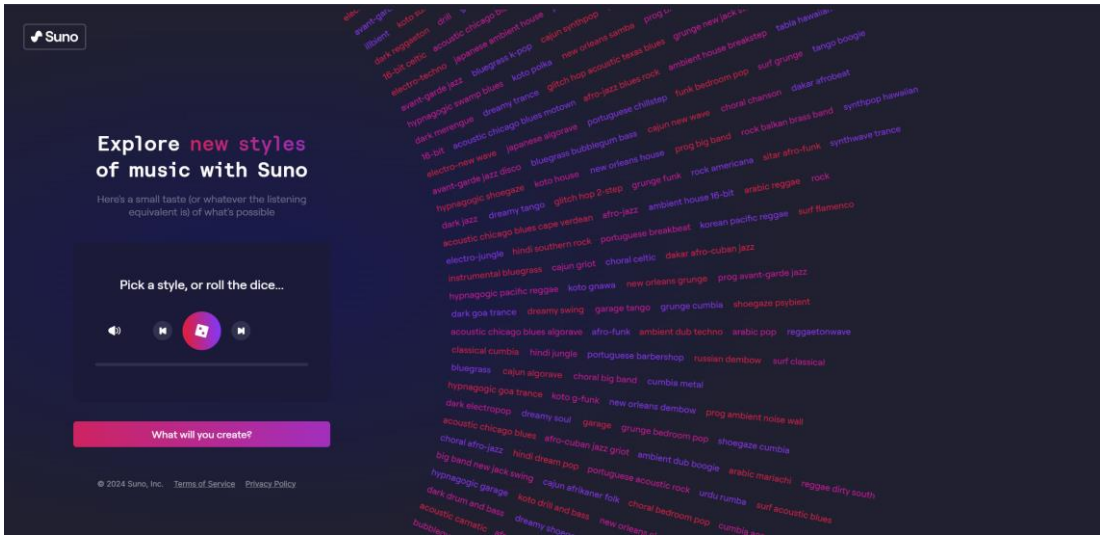


Рисунок 3.4 – Зовнішній вигляд вкладки Explore

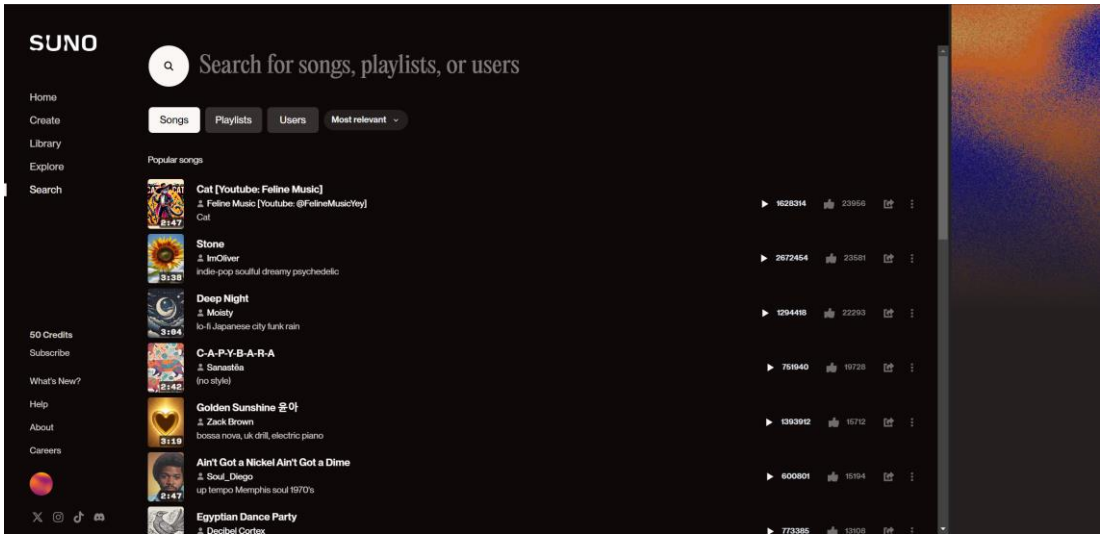


Рисунок 3.5 – Зовнішній вигляд вкладки Search

2.5 Генерація на основі текстових запитів

Генерація на основі текстових запитів одна з ключових функцій Suno, що дозволяє користувачам створювати музичні композиції за допомогою простих текстових описів.

Користувачі можуть вводити текстові описи, які описують бажаний стиль, настрій, темп чи конкретні елементи музики. Наприклад: "мелодія у стилі джазу з елементами електроніки" (рис. 3.6). Suno аналізує ці запити та використовує алгоритми штучного інтелекту для створення музичних творів, які відповідають заданим критеріям.

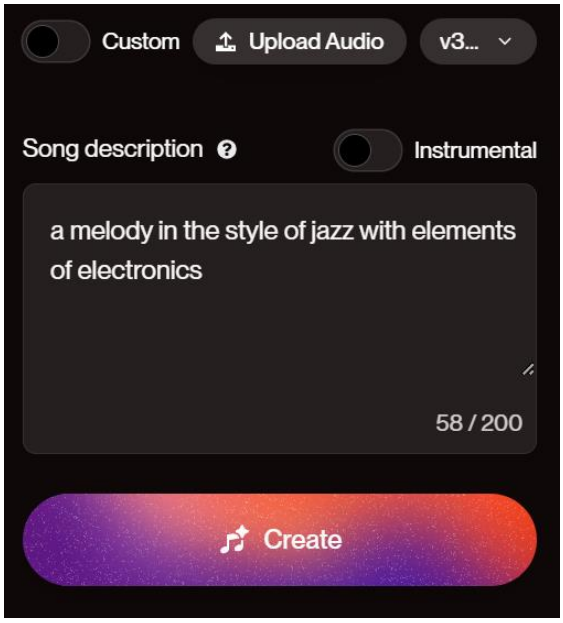


Рисунок 3.6 – Зовнішній вигляд інтерфейсу функції “Генерація на основі текстових запитів”

Користувачі можуть задавати тематику пісні, описуючи її зміст і основні ідеї. Наприклад, "романтична пісня про кохання", "патріотична пісня про Україну" або "надихаюча пісня про досягнення мрій", або емоційний настрій пісні, ви можете

отримати текст і музику, які передають потрібні емоції. Наприклад, "сумна балада", "веселий і оптимістичний трек" або "мотивуюча пісня".

При створенні інструментальної версії в Suno, користувачі мають можливість детально налаштовувати композицію за допомогою текстових вказівок у промпті. Це дозволяє створювати музику, що точно відповідає вашим вимогам і очікуванням.

В промпт можна включати вказівки щодо композиції:

– можна вказати бажаний темп композиції (в ударах на хвилину, BPM), наприклад, швидкий темп – "120 BPM" або повільний темп – "60 BPM";

– додавати інструменти, такі як піаніно, гітара, барабани, скрипка тощо, наприклад, "композиція з використанням піаніно і скрипки";

– можна вказати жанр і стиль музики. Це можуть бути класичні симфонії, джазові імпровізації, електронні біти тощо, наприклад, "джазова композиція з елементами імпровізації" або "електронний трек в стилі техно";

– настрій композиції, описуючи бажану атмосферу. Це може бути, наприклад, "спокійна і релаксуюча музика" або "енергійний і драйвовий трек";

– додавати спеціальні ефекти або звукові елементи, щоб зробити композицію унікальною, наприклад, "ефект реверберації" або "синтезаторні звуки";

Режим **Custom Mode** (рис. 3.7) в Suno — це розширений режим, який дозволяє користувачам детально налаштовувати всі аспекти створюваної музичної композиції.

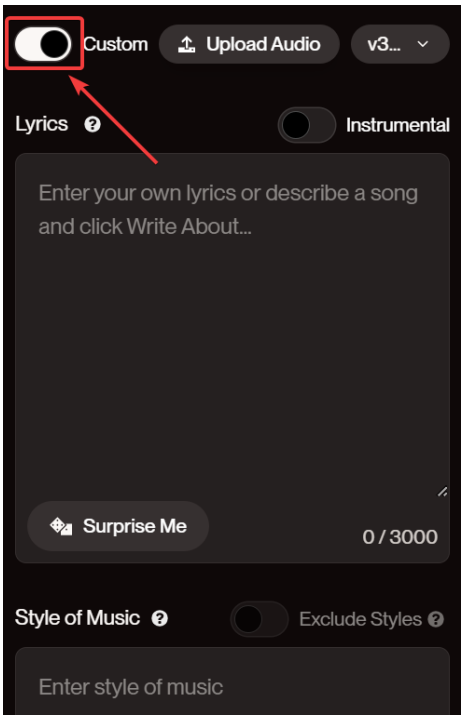


Рисунок 3.7 – Увімкнення режиму Custom Mode

В режимі Custom Mode ви можете використовувати власний текст (Lyrics) пісні, написаний будь-якою мовою. Ви можете змішувати різні мови в одній композиції, наприклад, писати куплети українською мовою, а приспіви англійською.

Також є можливість структурувати текст за допомогою спеціальних тегів, таких як [Verse] для куплетів, [Chorus] для приспівів, наприклад (рис. 3.8):

[Verse]
Лінь вставати зранку
О 8:30 рано
Очі закриваються

Будильник знов гуде

[Verse 2]
До університету
Дорога не така вже й близька
А хочеться залишитись
У ліжку на весь день

[Chorus]
Ой на 8:30
Треба вставати
Ліньки так сильно
Треба збиратись

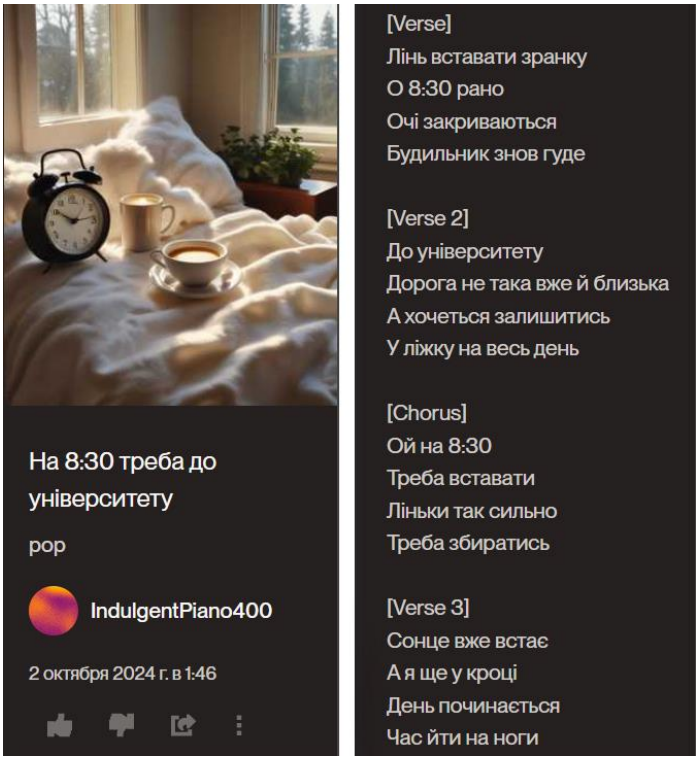


Рисунок 3.8 – Зовнішній вигляд згенерованої композиції

Ви можете вказати ім'я для вашої композиції в текстовому запиті. Якщо назва не вказана, Suno автоматично використовуватиме першу строчку пісні як назву композиції.

3 Завдання на практичну роботу

1. Зайти на сайт https://suno.com/ та зареєструватись на ньому любим зручним способом (якщо вже зареєстровані, то додаткова реєстрація не потрібна).
2. Створити пісню на довільну тематику, наприклад, “композиція, що передає відчуття пригоди, з ритмами барабанів та звуками природи”.
3. Створити пісню на тематику рецепту, наприклад, “пісня – рецепт торту”.
4. Створити інструментальну композицію, наприклад, “Сумна мелодія на фортепіано, що відображає втрату, з м'якими струнними інструментами”.
5. Створити пісню з власним текстом та стилем.
6. Переробити відому мелодію в різних стилях, наприклад, “Jingle bells”.

4 Методика виконання роботи

4.1 Реєстрація та вхід у систему

Перший крок до початку роботи з Suno.Ai – це створення облікового запису на платформі. Відвідайте офіційний веб-сайт Suno.Ai та натисніть на кнопку «Реєстрація».

Введіть свої дані, створіть пароль та підтвердіть електронну адресу. Після реєстрації, використовуйте свої облікові дані для входу в систему. Зареєструватися на сайті можна також за допомогою існуючих облікових даних, наприклад, Google-акаунту.

4.2 Ознайомлення з інтерфейсом

Після входу у систему, ви потрапите на головну сторінку Suno.Ai. Інтерфейс платформи інтуїтивно зрозумілий та поділений на кілька основних розділів:

– головне меню містить посилання на різні функції та інструменти платформи, такі як Home, Create, Library, Explore, Search;

– панель управління, яка містить інформацію про кількість доступних кредитів, параметри облікового запису та інші важливі дані;

– робоча область – основний простір для роботи з аудіо контентом, в якій є можливість створювати, редагувати та зберігати свої роботи у загальному репозитарію.

4.3 Створення аудіо на основі текстового запиту

1. Перейдіть у вкладку "Create" для створення аудіо.
2. В поле введення тексту “Song description” введіть детальний опис музики, яку ви хочете створити, наприклад, “Звуки космосу з елементами амбієнтної музики, електронних синтезаторів та ехо, що створює відчуття безмежності всесвіту.”
3. Після введення запиту натисніть кнопку “Create”. Система автоматично створить музику на основі вашого запиту.

4.4 Створення аудіо в Custom Mode

1. Перемкніть режим роботи у положення Custom Mode (рис. 3.9)
2. Введіть текст пісні (Lyrics) з використанням спеціальних тегів [Verse] для куплетів, [Chorus] для приспівів
3. Введіть стиль та жанр пісні в поле “Style of Music” наприклад “80s, rock, rock, violin, opera, opera”
4. За бажанням введіть назву пісні (Title)
5. Натисніть кнопку “Create”. Система автоматично створить музику на основі вашого запиту.

В результаті буде згенеровано два аудіо з однаковими текстами (Lyrics), але з різною мелодією, відповідно до запиту (промпту) та налаштувань, які були попередньо обрані. (рис. 3.9).

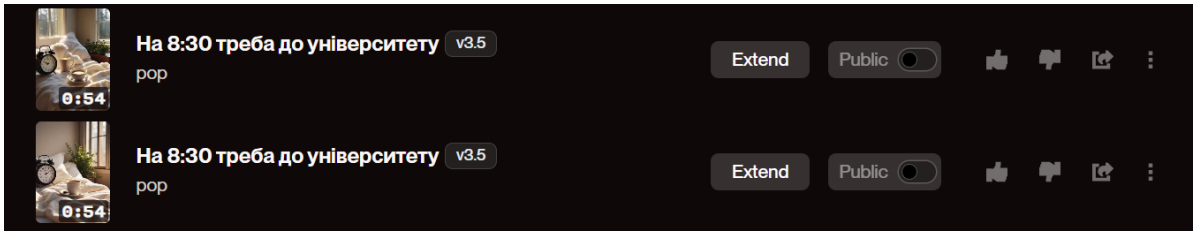


Рисунок 3.9 – Результат генерації Suno

5 Зміст протоколу

1. Назва та мета роботи.
2. Результати виконання завдань на практичну роботу, що містять:
– тексти запитів;
– посилання на згенерований файл.
3. Висновки по виконаній роботі.

Рекомендована література

Основна

1. Aggarwal, Charu C. Neural Networks and Deep Learning: A Textbook. Springer, 2023. 529 p. ISBN 9783031296420.

2. Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 3rd ed. O'Reilly Media, 2022. 814 p. ISBN 9781098125974.

3. Russell, Stuart, Norvig, Peter. Artificial Intelligence: A Modern Approach. 4th ed. Pearson, 2020. 1136 p. ISBN 9780134610993.

4. Goodfellow, Ian, Bengio, Yoshua, Courville, Aaron. Deep Learning. MIT Press, 2016. 800 p. ISBN 9780262035613.

5. Nielsen, Michael A. Neural Networks and Deep Learning. Determination Press, 2015. 222 p. ISBN 9780999171407.

Допоміжна

6. Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer, 2006. 738 p. ISBN 9780387310732.

7. Chollet, François. Deep Learning with Python. 2nd ed. Manning Publications, 2021. 504 p. ISBN 9781617296864.

8. Joshi, Prateek. Artificial Intelligence with Python. Packt Publishing, 2017. 423 p. ISBN 9781786464392.

9. Шаховська Н. Б., Камінський Р. М., Вовк О. Б. Системи штучного інтелекту. Львів: Видавництво Львівської політехніки, 2018. 392 с. ISBN 9789669412140.

10. Литвин В. В., Пасічник В. В., Яцишин Ю. В. Інтелектуальні системи. Львів: Новий Світ-2000, 2019. 406 с. ISBN 9789664182956.

Інформаційні ресурси

11. TensorFlow. URL: https://www.tensorflow.org/

12. PyTorch. URL: https://pytorch.org/

13. Keras. URL: https://keras.io/

14. Orange Data Mining. URL: https://orangedatamining.com/

15. Weka Machine Learning Project. URL: https://www.cs.waikato.ac.nz/ml/weka/

16. RapidMiner. URL: https://rapidminer.com/

Навчальне видання

Русу Олександр Петрович

ШТУЧНІ НЕЙРОННІ МЕРЕЖІ

Методичні вказівки до практичних робіт для здобувачів вищої освіти,
які навчаються за спеціальностями галузі «Інформаційні технології»

Українською мовою