**Лабораторна робота 5**

РОЗРОБКА ПРОСТИХ НЕЙРОННИХ МЕРЕЖ

**Мета*:*** використовуючи спеціалізовані бібліотеки та мову програмування Python навчитися створювати та застосовуватипрості нейронні мережі**.**

Хід роботи

**Завдання 2.1. Створити простий нейрон**

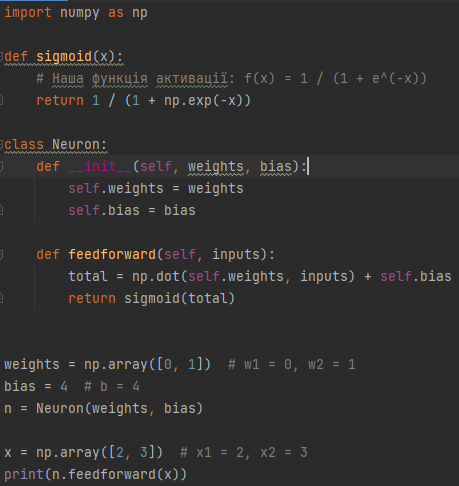


Рис. 1 Лістинг програми



Рис. 2 Результат виконання програми

**Завдання 2.2. Створити просту нейронну мережу для передбачення статі людини**

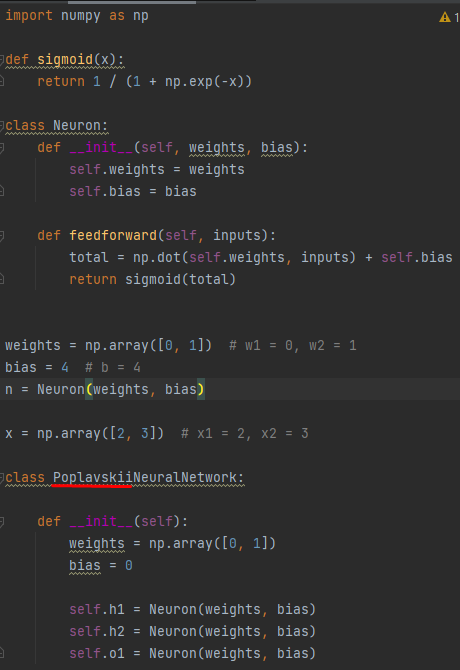


Рис. 3 Лістинг програми

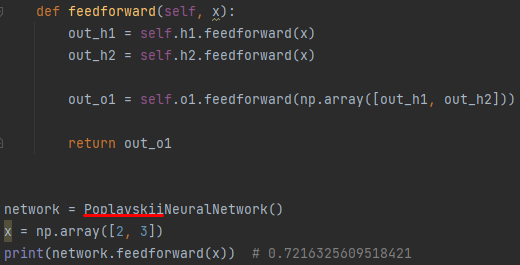


Рис. 4 Лістинг програми



Рис. 5 Результат виконання програми

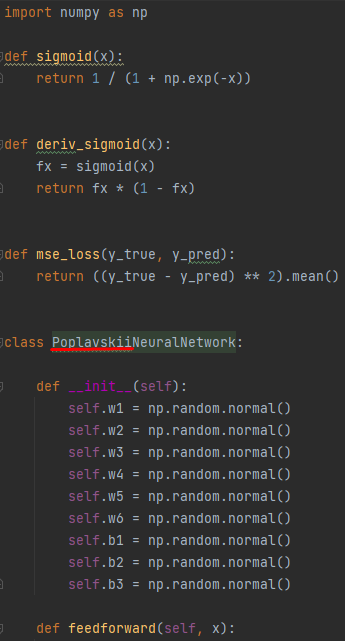


Рис. 6 Лістинг програми

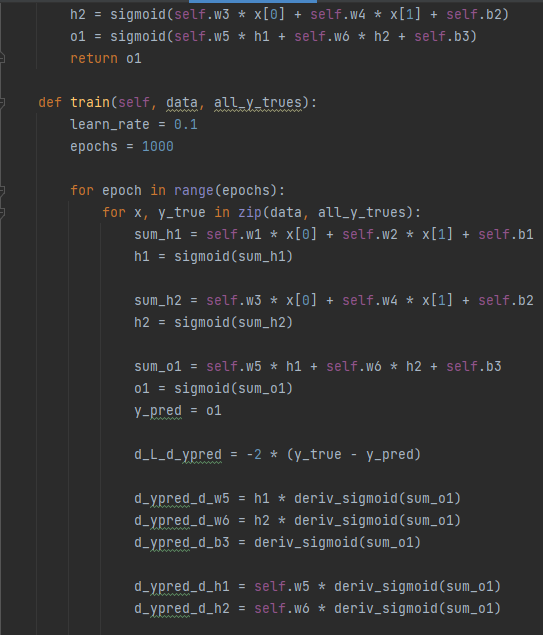


Рис. 7 Лістинг програми

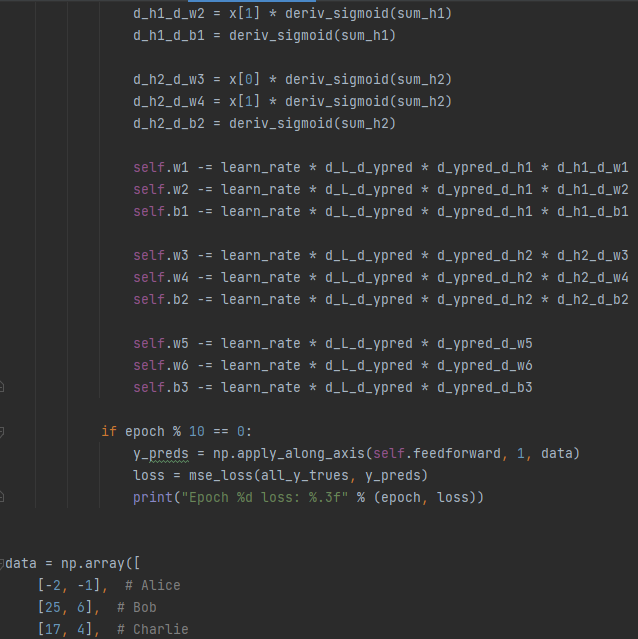


Рис. 8 Лістинг програми

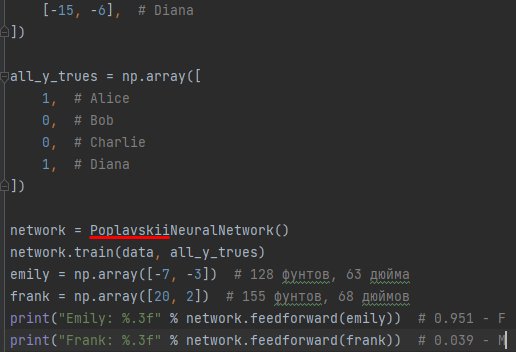


Рис. 9 Лістинг програми

Висновок: Функція активації використовується для підключення незв'язаних вхідних даних із виходом, у якого проста та передбачувана форма. Як правило, як функція активації найбільш часто використовується функція сигмоїди. Можливості нейроннних мереж прямого поширення полягають в тому, що сигнали поширюються в одному напрямку, починаючи від вхідного шару нейронів, через приховані шари до вихідного шару і на вихідних нейронах отримується результат опрацювання сигналу. В мережах такого виду немає зворотніх зв’язків.

Нейронні мережі прямого поширення знаходять своє застосування в задачах комп'ютерного бачення та розпізнаванні мовлення, де класифікація цільових класів ускладнюється. Такі типи нейронних мереж добре справляються із зашумленими даними.

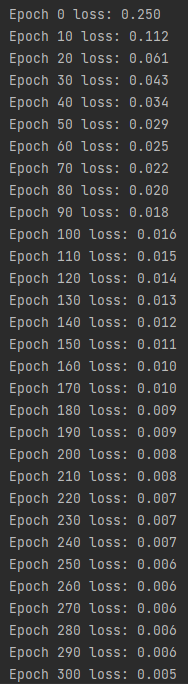


Рис. 10 Результат виконання програми

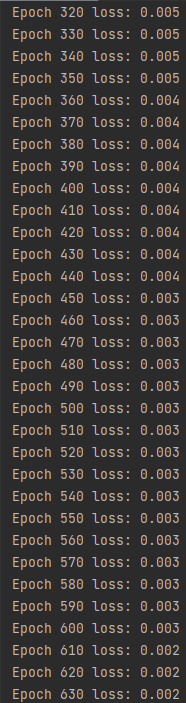


Рис. 11 Вмконання програми

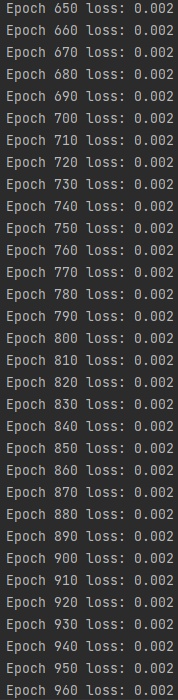


Рис. 12 Результат виконання програми

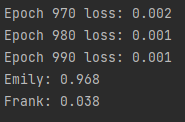


Рис. 13 Результат виконання програми

**Завдання 2.3. Класифікатор на основі перцептрону з використанням бібліотеки NeuroLab**

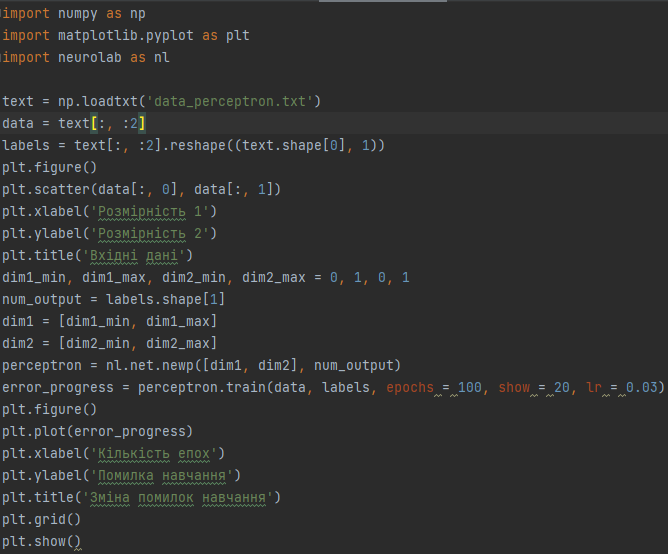


Рис. 14 Лістинг програми

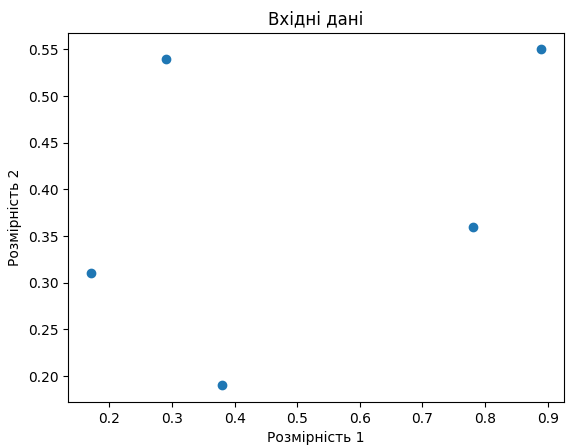


Рис. 15 Графік вхідних даних



Рис. 16 Графік процесу навчання

Висновок: На другому графіку я відобразив процес навчання, використовуючи метрику помилки. Під час першої епохи відбулося від 1.0 до 1.5 помилок, під час наступних двох епох відбулось 1.5 помилок. Потім під час 4 епохи помилки почались зменшувтась, тому що ми навчили перцептрон за допомогою тренувальних даних.

**Завдання 2.4. Побудова одношарової нейронної мережі**

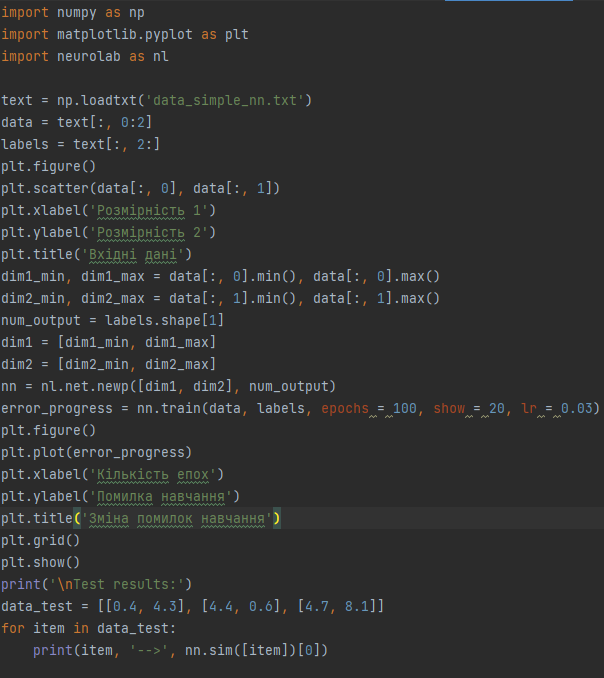


Рис. 17 Лістинг програми

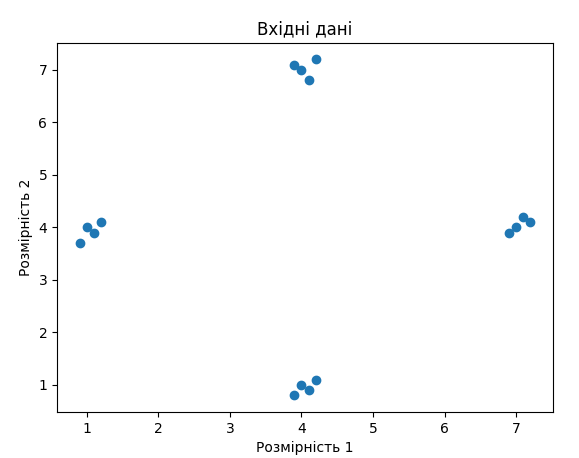


Рис. 18 Графік вхідних даних

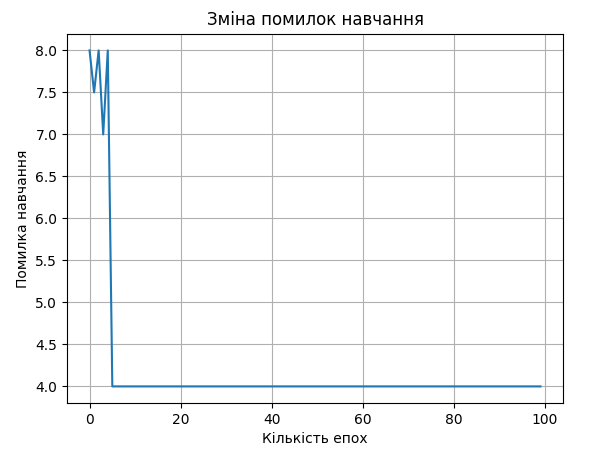


Рис. 19 Графік просування процесу навчання

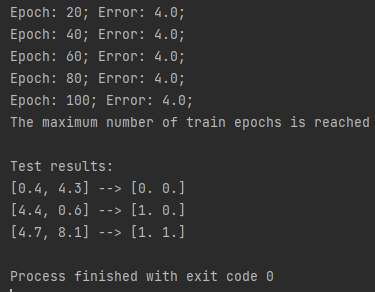


Рис. 20 Результат виконання програми

Висновок: На рис. 20 зображено процес навчання мережі. На 20 епосі відбулось 4 помилки, аналогічно на 40, 60, 80 та 100. Потім вивелось повідомлення, що ми досягли максимальної кількості епох для тренування. Ми вирішили визначити вибіркові тестові точки даних та запустили для них нейронну мережу. І це його результат.

**Завдання 2.5. Побудова багатошарової нейронної мережі**

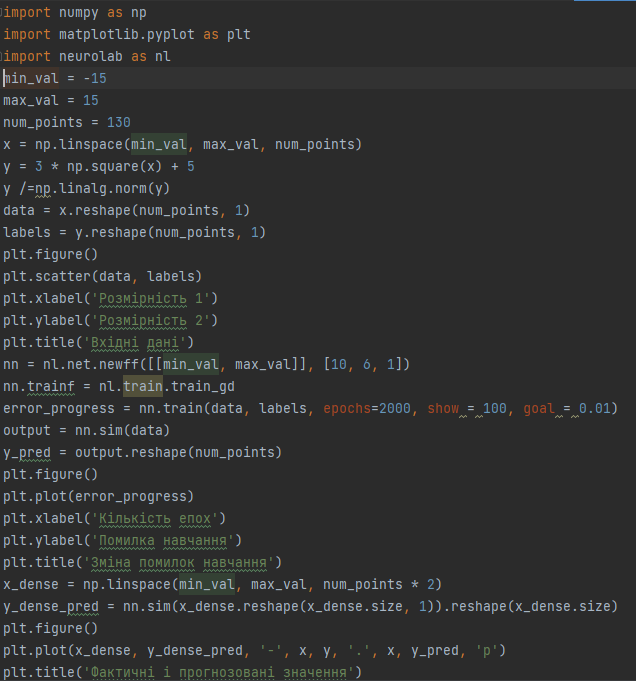


Рис. 21 Лістинг програми

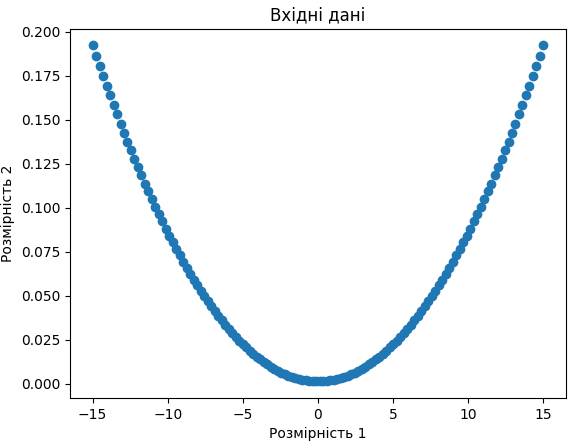


Рис. 22 Результат виконання програми

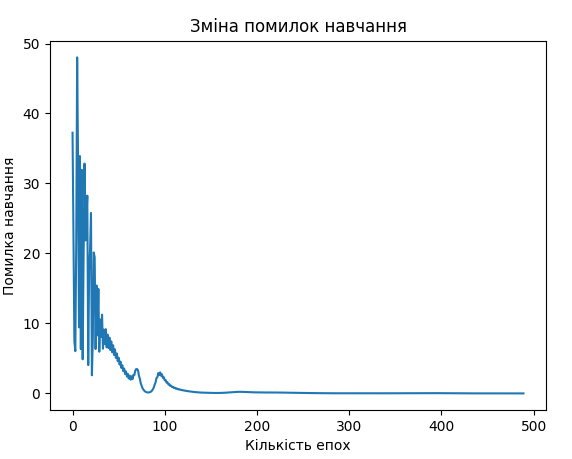


Рис. 23 Результат виконання програми

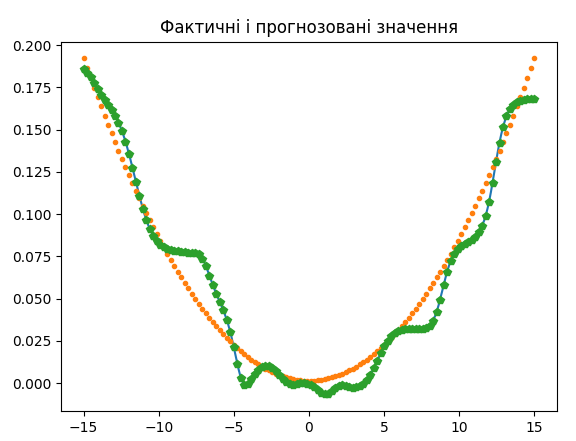


Рис. 24 Результат виконання програми

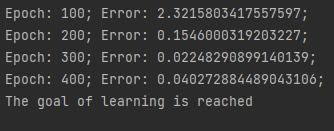
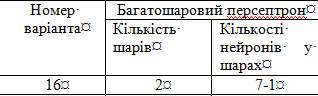


Рис. 25 Результат виконання програми

Висновок: На рис. 25 зображено процес навчання мережі. На 100 епосі відбулось 2.32 помилки, на 200 епосі відбулось 0.15 помилки, на 300 епосі відбулось 0.02 помилки, на 400 епосі відбулось 0.04 помилки,. Потім вивелось повідомлення, що ми досягли цілі навчання.

**Завдання 2.6. Побудова багатошарової нейронної мережі для свого варіанту**

****

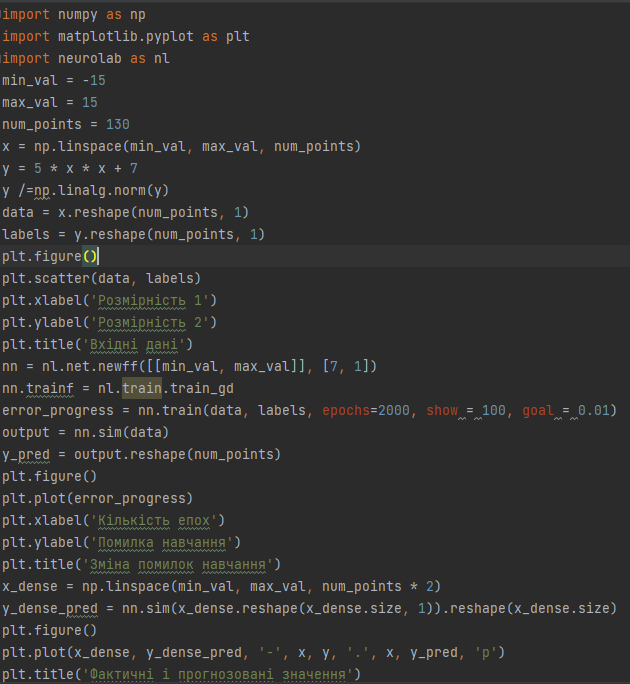


Рис. 25 Лістинг програми

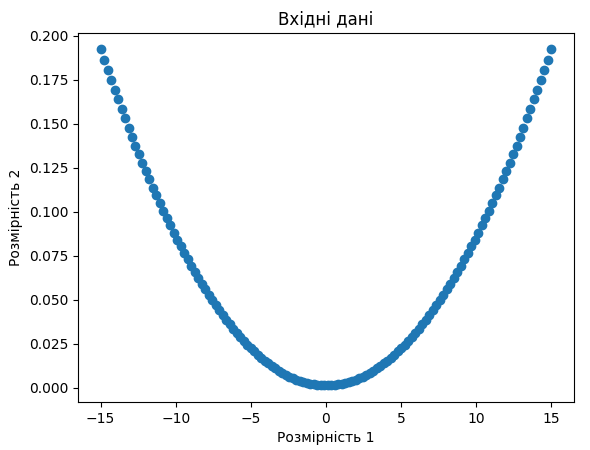


Рис. 26 Результат виконання програми

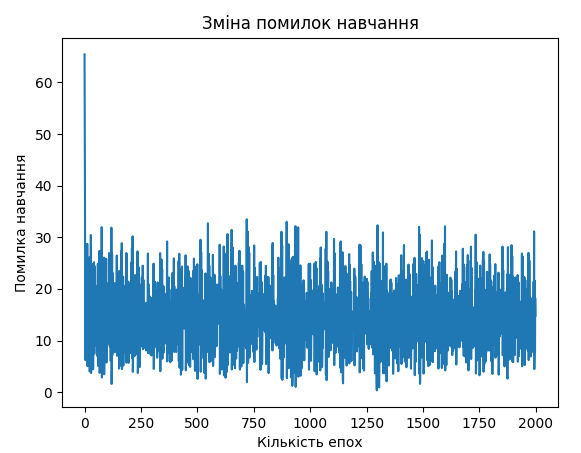


Рис. 27 Результат виконання програми

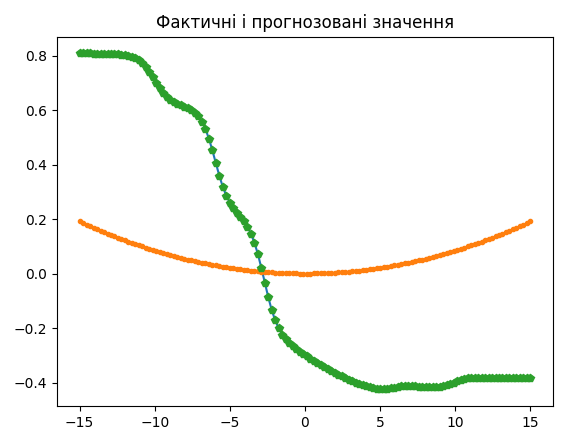


Рис. 28 Результат виконання програми

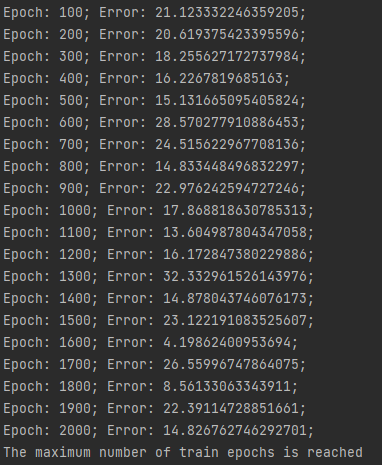


Рис. 29 Результат виконання програми

На рис. 29 зображено процес навчання мережі. На 100 епосі відбулось 21.12 помилки, на 200 епосі відбулось 20.62 помилки, на 300 епосі відбулось 18.26 помилки і так далі, на 2000 епосі відбулось 14.83 помилки,. Потім вивелось повідомлення, що ми досягли максимальної кількості епох для тренування.

**Завдання 2.7. Побудова нейронної мережі на основі карти Кохонена, що самоорганізується**

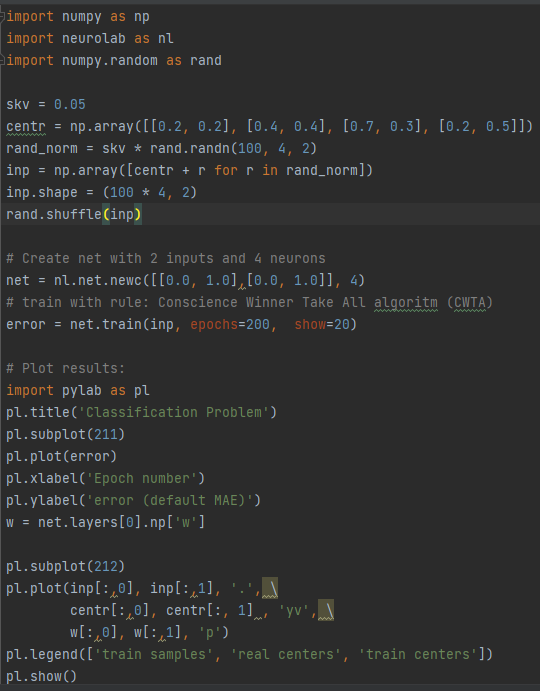


Рис. 30 Лістинг програми

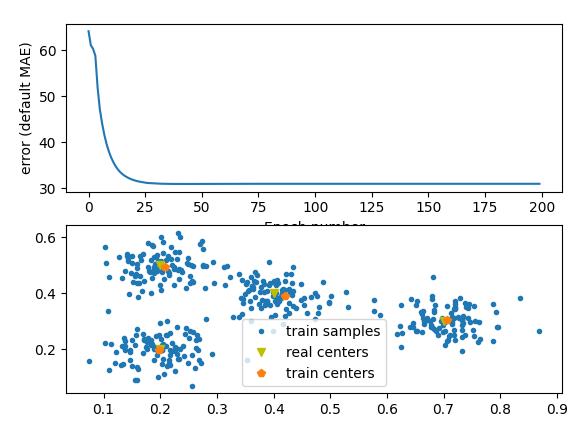


Рис. 31 Результат виконання програми

Помилка MAE - [Средня абсолютна помилка (Mean Absolute Error). Середньою абсолютною похибкою називають середнє арифметичне з абсолютних похибок усіх вимірювань.](https://wiki.loginom.ru/articles/mae.html)

**Завдання 2.8. Дослідження нейронної мережі на основі карти Кохонена, що само організується**

****

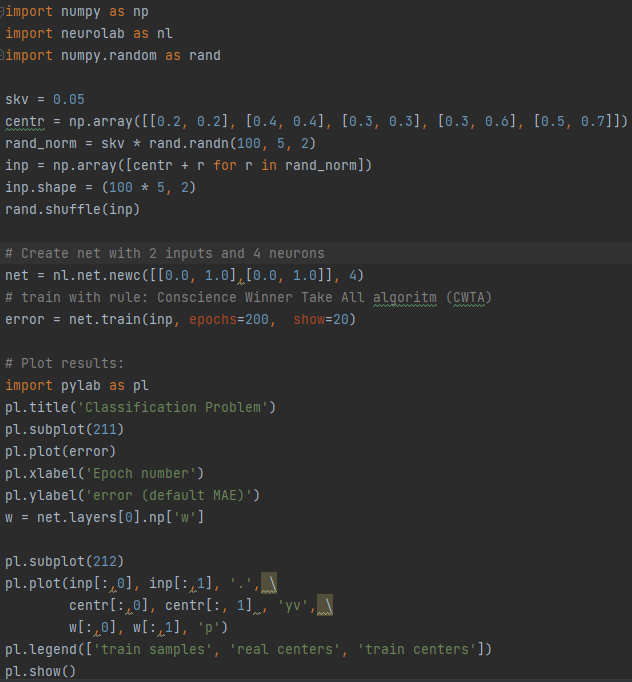


Рис. 32 Лістинг програми

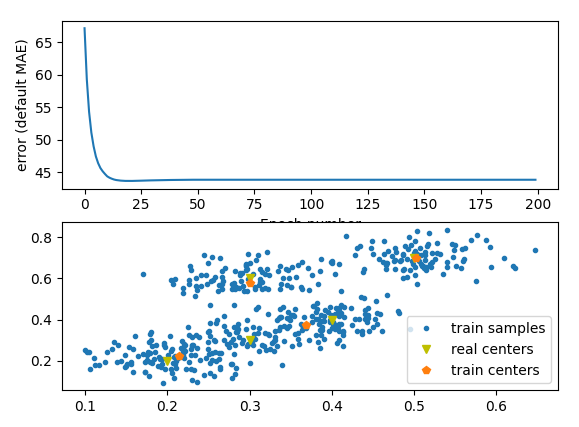


Рис. 33 Результат виконання програми

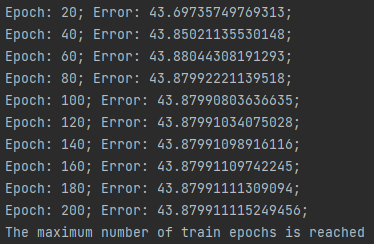


Рис. 34 Результат виконання програми

На рис. 34 зображено процес навчання мережі. На 20 епосі відбулось 43.7 помилки, на 40 епосі відбулось 43.85 помилки, на 60 епосі відбулось 43.88 помилки і так далі, на 200 епосі відбулось 43.88 помилки,. Потім вивелось повідомлення, що ми досягли максимальної кількості епох для тренування.



Рис. 35 Лістинг програми

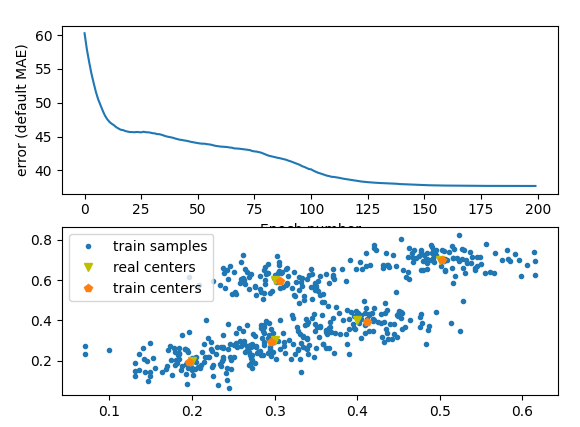


Рис. 36 Результат виконання програми

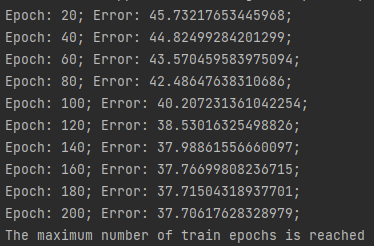


Рис. 37 Результат виконання програми

На рис. 37 зображено процес навчання мережі. На 20 епосі відбулось 45.73 помилки, на 40 епосі відбулось 44.82 помилки, на 60 епосі відбулось 43.57 помилки і так далі, на 200 епосі відбулось 37.7 помилки,. Потім вивелось повідомлення, що ми досягли максимальної кількості епох для тренування.

Якщо порівнювати нейронну мережу Кохонена з 4 нейронами та 5 нейронами, можна зробити такі висновки. При 4 нейронах Помилка МAE повільніше зменшується, ніж з 5 нейронами,також з 5 нейронами ця помилка нижча. З 5 нейронами обоє центрів збігаються майже в одні точці. Число нейронів в шарі Кохонена має відповідати числу класів вхідних сигналів. Тобто в нашому випадку нам давалось 5 вхідних сигналів, значить у нас має бути 5 нейронів, а не 4.Отже, невірний вибір кількості нейронів числу кластерів впливає на величину помилки ускладнюючи навчання мережі і швидкості, тому на рис. 33 набагато гірші результати, ніж на рис. 36

**ВИСНОВОК:** під час виконання лабараторної роботи, використовуючи спеціалізовані бібліотеки та мову програмування Python навчитися створювати та застосовуватипрості нейронні мережі**.**

**Репозиторій:**