A drawing of a building

Description automatically generated

**Національний технічний університет України**

**“Київський політехнічний інститут ім. Ігоря Сікорського”**

**Факультет прикладної математики**

**Кафедра системного програмування і спеціалізованих комп’ютерних систем**

**Розрахунково-графічна робота**

***з дисципліни***

***“Системи штучного інтелекту”***

**Група: КВ-41мн**

**Виконав: Пилипченко Б. О.**

**Київ – 2024**

Зміст

[1. Завдання на роботу 3](#_Toc186510017)

[2. Про реалізацію 3](#_Toc186510018)

[3. Тестування 4](#_Toc186510019)

[3.1 8 класів 4](#_Toc186510020)

[3.2 16 класів 8](#_Toc186510021)

[Висновки 14](#_Toc186510022)

# Завдання на роботу

Написати програму реалізації нейронної мережі Кохонена. Згенерувати Q (10 - 25) бінарних векторів, що належати до різних з М (8 - 16) класів. Розмір кожного з векторів N (25 - 50) елементів. Наприклад,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, //class 0

1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, //class 0

0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, //class 1

0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, //class 2

0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, //class 1

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, //class 2

Згенерувати декілька датасетів із різною кількістю класів. Для кожного датасету створити окрему мережу і навчити її, дослідити залежність кількості хибно класифікованих векторів від загальної кількості навчальних епох.

# Про реалізацію

Мережа Кохонена була реалізована мовою програмування C#.

Посилання на гітхаб репозиторій: [Bohdan628318ylypchenko/Kohonen-Net](https://github.com/Bohdan628318ylypchenko/Kohonen-Net)

У програмі мережа представляється у вигляді матриці, колонки матриці є ваговими коефіцієнтами відповідних кластерів (клас KNet). Класифікація зразка виконується наступним чином:

1. Вектор-зразок множиться на матрицю вагових коефіцієнтів мережі.
2. Клас зразка вважається рівним індексу найбільшого елемента вектору – результату множення зразка на вагові коефіцієнти.

Навчання мережі на датасеті реалізоване методом Train (клас KNet):

public void Train(Dataset dataset, int epochCount, double l)

{

var shuffledSamples = new Vector<double>[dataset.SampleCount];

Array.Copy(dataset.Samples, shuffledSamples, dataset.SampleCount);

for (var i = 0; i < epochCount; i++)

{

RandomSingleton.Generator.Shuffle(shuffledSamples);

foreach (var sample in shuffledSamples)

{

int indexOfGroup = PredictAsInternal(sample);

var columnOfGroup = weights.Column(indexOfGroup);

weights.SetColumn(indexOfGroup, columnOfGroup + (sample - columnOfGroup).Multiply(l));

}

}

UpdateGroupOrder();

}

Параметрами навчання є:

* Навчальний датасет
* Кількість епох
* Коефіцієнт спотворення образу L

Критерій зупинки навчання – виконання заданої кількості епох.

Епохою є проходження по усім зразкам датасету із корегуванням відповідної колонки матриці вагових коефіцієнтів для кожного зразка за формулою:

, де:

* – поточний зразок,
* – ідентифікатор (індекс) класу, до якого було віднесено зразок s,
* - нове значення вагових коефіцієнтів і-того класу,
* - поточне значення вагових коефіцієнтів і-того класу,
* – коефіцієнт спотворення образу L.

Варто звернути увагу, що перед проходженням зразків датасет перемішується випадковим чином. Перемішування відбувається окремо для кожної епохи. Початкова реалізація мережі не містила перемішування датасету. Спроби навчити мережу без перемішування зразків призводили до нездатності мережі розрізняти зразки деяких класів (зазвичай 2 класів). Додавання перемішування значно пришвидшило навчання мережі за однакову кількість епох, дозволило отримувати мережі, які правильно розпізнають зразків ВСІХ класів із датасету.

# Тестування

## 8 класів

Параметри датасету:

розмірність векторів – 32

кількість класів – 8

кількість векторів - 25

A black screen with white text

Description automatically generated

Згенерований датасет:

A screenshot of a computer screen

Description automatically generated

(Алгоритм генерації датасету реалізовано класом DatasetGenerator, див. репозиторій).

Навчання полягало у послідовному виконанні 2 експериментів із оцінкою навченої мережі після кожного експерименту:

A screen shot of a computer program

Description automatically generated

(Нульовий експеримент у масиві experiments не навчає мережу, оскільки L = 0. Перший експеримент лише оцінює мережу до навчання).

Вагові коефіцієнти початкової мережі та оцінка мережі на датасеті:

A screenshot of a computer

Description automatically generated

Бачимо, що початкова мережа хибно розпізнала 15 векторів із 25.

Вагові коефіцієнти мережі та оцінка мережі після першого експерименту (10 епох, L = 0.05):

A screenshot of a computer

Description automatically generated

Мережа хибно розпізнала лише 2 зразки із 25. Маємо суттєвий прогрес за лише 10 ітерацій!

Імовірно, значення вагових коефіцієнтів мережі близькі до таких, що забезпечать правильне розпізнавання всіх зразків датасету. Зменшимо значення L до 0.01 у другому експерименті. При цьому, якщо мережа знайде оптимальні вагові коефіцієнти, власне різниця між зразками і колонками класів прямуватиме до 0, та множитиметься на 0.01. Тобто можемо дозволити компенсувати мізерність змін коефіцієнтів впродовж однієї епохи збільшенням кількості епох з 10 до 20.

Маємо:

A screenshot of a computer

Description automatically generated

Мережа успішно розпізнає всі зразки датасету. Для знаходження оптимальних значень вагових коефіцієнтів всього знадобилось 30 епох.

## 16 класів

Згенеруємо новий датасет із більшою кількістю класів – 16, інші параметри датасету залишимо без змін:

A black screen with white text

Description automatically generated

Згенерований датасет:

A screenshot of a computer screen

Description automatically generated

Повна послідовність експериментів:

A screen shot of a computer code

Description automatically generated

Оцінимо початкову мережу:

A screen shot of a computer

Description automatically generated

Із 25 зразків мережа хибно розпізнала 17 зразків.

В якості першого експерименту виконаємо той самий експеримент, що і для мережі з 8 класами: кількість епох – 10, L = 0.05:

A black screen with white dots

Description automatically generated

Як і у випадку з 8 класами, маємо значне покращення результатів: мережа розпізнала хибно лише 3 зразки із 25.

В якості другого експерименту оберемо ті самі параметри, що і для 8 класів: кількість епох – 20, L = 0.01:

A black and white screen with many white squares

Description automatically generated

Маємо лише 1 хибно класифікований вектор.

Для експерименту 3 оберемо ті ж параметри, що і для експерименту 2:

A black and white screen with many white squares

Description automatically generated with medium confidence  
Мережа навчилась, всі 25 зразків класифіковано вірно.

# Висновки

Було створено і навчено 2 мережі Кохонена:

* мережа для 8 класів
* мережа для 16 класів

Для того, щоб мережа навчилась безпомилково розпізнавати 8 класів, знадобилось 30 епох. Для безпомилкового розпізнавання 16 класів знадобилось 50 епох.

Кількість епох очікувано зростає із кількість класів. Збільшення кількості класів призводить до збільшення кількості вагових коефіцієнтів, а отже зростає «час» пошуку оптимальних коефіцієнтів. При цьому в обох випадках мережа демонструє швидке покращення результатів вже через 10 епох.