МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №5 з дисципліни

«Методи та технології штучного інтелекту»

«Моделювання нейронної мережі Хебба»

Перевірив:
Шимкович В.М.

Виконав:
студент 3 курсу
групи ІП-11 ФІОТ
Прищепа В.С.

Київ-2023

Лабораторна робота №5

Дослідження алгоритму нечіткої кластеризації

Мета роботи: Промоделювати та дослідити нейронну мережу Хебба.

Завдання:
1. Розробіть структуру мережі Хебба, яка здатна розпізнавати чотири різні літери вашого імені або прізвища.
2. Розробіть алгоритм і програму, що моделює мережу Хебба. При цьому в алгоритмі обов'язково передбачте можливість виникнення ситуацій з нерозв'язними проблемами адаптації ваг зв'язків нейромережі.
3. Навчіть нейронну мережу Хебба розпізнаванню чотирьох заданих букв вашого імені або прізвища.
4. Продемонструйте працездатність мережі при пред'явленні навчальних зображень і зображень, що містять помилки.
5. Оформіть звіт по лабораторній роботі.

Хід роботи:
Створимо 2 функції для реалізації мережі Хебба, одна з яких буде створювати ваги зв’язків, а інша розпізнавати літери. Використаємо літери мого імені V, L, A, Y, створимо до кожної з них зображення і для L та A створимо некоректні зображення для тестування розпізнання. На коректних навчимо мережу Хебба, а потім на некоректних перевіримо, чи розпізнає вона зображення літер правильно.

Лістинг:
```
def hebb_network(letters, expected_result, neurons_number):
    for index in range(neurons_number):
        letters[index] = [1] + letters[index]
    weights = [[0] * len(letters[0]) for _ in range(neurons_number)]
    # calculate weights
    for index_of_letter in range(neurons_number):
        for index_of_neuron in range(neurons_number):
            for index_of_weight in range(len(weights[index_of_neuron])):
                weights[index_of_neuron][index_of_weight] += letters[index_of_letter]
[index_of_weight] * expected_result[index_of_letter][index_of_neuron]
    # calculate output
    actual_result = calculate_output(letters, weights, neurons_number)
    # check ending rule
    if actual_result == expected_result:
```

```
        return weights
    raise Exception('There is unsolvable problem of weight adaptation. Weights: ' +
str(weights))

def calculate_output(letters, weights, neurons_number):
    actual_result = []
    for index_of_letter in range(len(letters)):
      letter_result = []
      for index_of_neuron in range(neurons_number):
        s = 0
        for index_of_weight in range(len(weights[index_of_neuron])):
            s += weights[index_of_neuron][index_of_weight] * letters[index_of_letter]
[index_of_weight]
        if s > 0:
            letter_result += [1]
        else:
            letter_result += [-1]
      actual_result += [letter_result]
    return actual_result

# letters
v = [ 1, -1, 1,
 1, -1, 1,
 -1, 1, -1]
l = [ 1, -1, -1,
 1, -1, -1,
 1, 1, 1]
a = [ -1, 1, -1,
 1, 1, 1,
 1, -1, 1]
y = [1, -1, 1,
 -1, 1, -1,
 -1, 1, -1]

# expected result
expected_result = [[ 1, -1, -1, -1],
 [-1, 1, -1, -1],
 [-1, -1, 1, -1],
 [-1, -1, -1, 1]]

# train network
train_letters = [v, l, a, y]
number_of_neurons = len(train_letters)
final_weights = hebb_network(train_letters, expected_result, number_of_neurons)
print("Weights:")
```

```
for weights in final_weights:
    print(weights)

# test network
mistake1 = [ -1, 1, -1,
 1, 1, 1,
 1, 1, 1] # a with mistake
mistake2 = [ 1, -1, -1,
 1, -1, -1,
 1, -1, 1] # l with mistake

letters_with_mistakes = [v, l, a, y, mistake1, mistake2]
# add x0 = 1 to every letter
for index in range(len(letters_with_mistakes)):
     letters_with_mistakes[index] = [1] + letters_with_mistakes[index]
actual_result = calculate_output(letters_with_mistakes, final_weights,
number_of_neurons)
print("\nResult (V, L, A, Y, A with mistake, L with mistake):")
for res in actual_result:
    print(res)
```

Виконання:

Спочатку розробимо тренувальні зображення обраних літер, проведемо на них тренування, створимо зображення з помилками і виконаємо тестування.

Правильні зображення:

	V				L	
1	-1	1		1	-1	-1
1	-1	1		1	-1	-1
-1	1	-1		1	1	1
	A				Y	
-1	1	-1		1	-1	1
1	1	1		-1	1	-1
1	-1	1		-1	1	-1

Зображення з помилками:

	L				A	
1	-1	-1		-1	1	-1
1	-1	-1		1	1	1
1	-1	1		1	1	1

Результат мусить бути наступним:

	V	L	A	Y
V	1	-1	-1	-1
L	-1	1	-1	-1
A	-1	-1	1	-1
Y	-1	-1	-1	1
A з помилками	-1	-1	1	-1
L з помилками	-1	1	-1	-1

Результати тренування моделі у вигляді обчислених ваг та тестування розпізнавання:

```
Weights:
[-2, 0, 0, 2, 0, -2, 2, -2, 0, -2]
[-2, 0, 0, -2, 0, -2, -2, 2, 0, 2]
[-2, -4, 4, -2, 0, 2, 2, 2, -4, 2]
[-2, 0, 0, 2, -4, 2, -2, -2, 0, -2]

Result (V, L, A, Y, A with mistake, L with mistake):
[1, -1, -1, -1]
[-1, 1, -1, -1]
[-1, -1, 1, -1]
[-1, -1, -1, 1]
[-1, -1, 1, -1]
[-1, 1, -1, -1]
```

Висновок:

Під час виконання лабораторної роботи я дослідив та промоделював нейронну мережу Хебба з 4 нейронами. Навчив мережу розпізнавати літери V, L, A, Y. Також прогнав через мережу зображення літер L та A з помилками і мережа розпізнала ці зображення коректно (як L і A відповідно). Результати виконання лабораторної наведені вище.