

ЛАБОРАТОРНА РОБОТА № 5

РОЗРОБКА ПРОСТИХ НЕЙРОННИХ МЕРЕЖ

Посилання на GitHub: <https://github.com/FrancIwanicki/OAI.git>

Мета роботи: використовуючи спеціалізовані бібліотеки та мову програмування Python навчитися створювати та застосовувати прості нейронні мережі.

2. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ ТА МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ЙОГО ВИКОНАННЯ

Завдання 2.1. Створити простий нейрон

Лістинг програми:

```
import numpy as np
import self

def sigmoid(x):
    # Наша функція активації:  $f(x) = 1 / (1 + e^{-x})$ 
    return 1 / (1 + np.exp(-x))

class Neuron:
    def __init__(self, weights, bias):
        self.weights = weights
        self.bias = bias

    # Вхідні дані про вагу, додавання зміщення і подальше використання функції ак-
    # тивації
    def feedforward(self, inputs):
        total = np.dot(self.weights, inputs) + self.bias
        return sigmoid(total)

weights = np.array([0, 1]) #w1 = 0, w2 = 1
bias = 4 #b = 4
n = Neuron(weights, bias)
x = np.array([2, 3]) #x1 = 2, x2 = 3
print(n.feedforward(x))
```

Результат виконання:

```
C:\Users\franc\OneDrive\Desktop\labsOAI\lab1\lab\Scripts\python.exe C:/Users/franc/OneDrive/Desktop/labsOAI/lab5/task1.py
0.9990889488055994
```

					ДУ «Житомирська політехніка».23.121.8.000 – Лр5			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Іваницький Ф.А.			Звіт з лабораторної роботи	Літ.	Арк.	Аркушів
Перевір.		Голенко М. Ю.					1	
Керівник						ФІКТ Гр. ІПЗ-20-3		
Н. контр.								
Зав. каф.								

Завдання 2.2. Створити просту нейронну мережу для передбачення статі людини

Лістинг програми:

```
import numpy as np
import self

def sigmoid(x):
    # Наша функція активації: f(x) = 1 / (1 + e^(-x))
    return 1 / (1 + np.exp(-x))

class Neuron:
    def __init__(self, weights, bias):
        self.weights = weights
        self.bias = bias

    # Вхідні дані про вагу, додавання зміщення і подальше використання функції ак-
    # тивації
    def feedforward(self, inputs):
        total = np.dot(self.weights, inputs) + self.bias
        return sigmoid(total)

weights = np.array([0, 1]) #w1 = 0, w2 = 1
bias = 4 #b = 4
n = Neuron(weights, bias)
x = np.array([2, 3]) #x1 = 2, x2 = 3
print(n.feedforward(x))

class IvanytskyiNeuralNetwork:
    def __init__(self):
        weights = np.array([0, 1])
        bias = 0
        self.h1 = Neuron(weights, bias)
        self.h2 = Neuron(weights, bias)
        self.o1 = Neuron(weights, bias)

    def feedforward(self, x):
        out_h1 = self.h1.feedforward(x)
        out_h2 = self.h2.feedforward(x)
        out_o1 = self.o1.feedforward(np.array([out_h1, out_h2]))
        return out_o1

network = IvanytskyiNeuralNetwork()
x= np.array([2, 3])
print(network.feedforward(x))
```

Результат виконання:

```
sys.path.extend(['C:\\Users\\franc\\OneDrive\\Desktop\\labs0AI\\lab5', 'C:/Users/franc/OneDrive/Desktop/labs0AI/lab5'])

Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)]
0.9990889488055994
0.7216325609518421
```

Висновки:

Функція Sigmoid – перетворює ваговані суми вхідних сигналів в діапазон значень між 0 і 1.

Mean Squared Error - визначає середньоквадратичну помилку між прогнозованими значеннями та фактичними значеннями вихідної змінної.

		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. Ю.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Можливості нейронних мереж прямого поширення:

1. Вони можуть бути використані для вирішення багатьох типів завдань, включаючи класифікацію, регресію, розпізнавання образів та багато інших.
2. Вони можуть навчатися на прикладах і вдосконалювати свої параметри, щоб наближати вихід до бажаного результату (наприклад, мінімізувати втрати).
3. Нейронні мережі можуть автоматично визначати ваги та зміщення для вирішення конкретних завдань, що робить їх потужними і універсальними інструментами для багатьох додатків.

Завдання 2.3. Класифікатор на основі перцептрону з використанням бібліотеки NeuroLab

Лістинг програми:

```
import numpy as np
import matplotlib.pyplot as plt
import neurolab as nl

# Завантаження вхідних даних
text = np.loadtxt('data_perceptron.txt')

# Поділ точок даних та міток
data = text[:, : 2]
labels = text[:, 2]. reshape((text. shape[0],1))

plt. figure()
plt.scatter(data[:, 0], data[:, 1])
plt. xlabel('Размерность 1 ')
plt.ylabel('Размерность 2')
plt. title('Входные данные')

# Визначення максимального та мінімального значень для кожного виміру
dim1_min,dim1_max,dim2_min,dim2_max = 0,1,0,1

# Кількість нейронів у вихідному шарі
num_output = labels.shape[1]

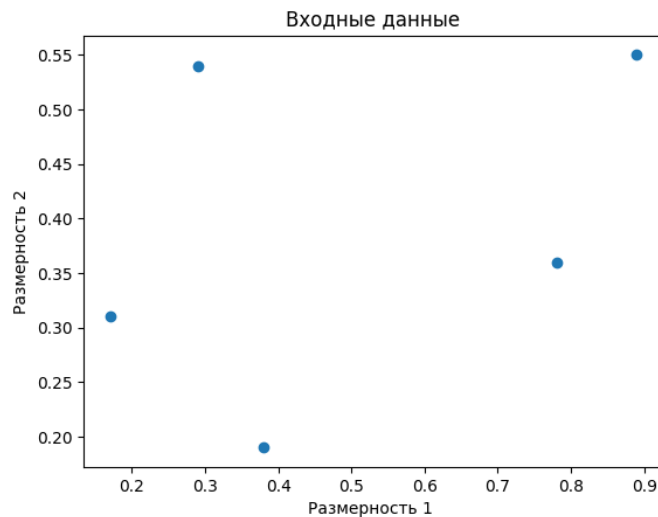
# Визначення перцептрону з двома вхідними нейронами (оскільки Вхідні дані - двовимірні)
dim1 = [dim1_min,dim1_max]
dim2 = [dim2_min,dim2_max]
perceptron = nl.net.newp([dim1,dim2],num_output)

# Тренування перцептрону з використанням наших даних
error_progress = perceptron.train(data, labels, epochs=100, show=20, lr=0.03)

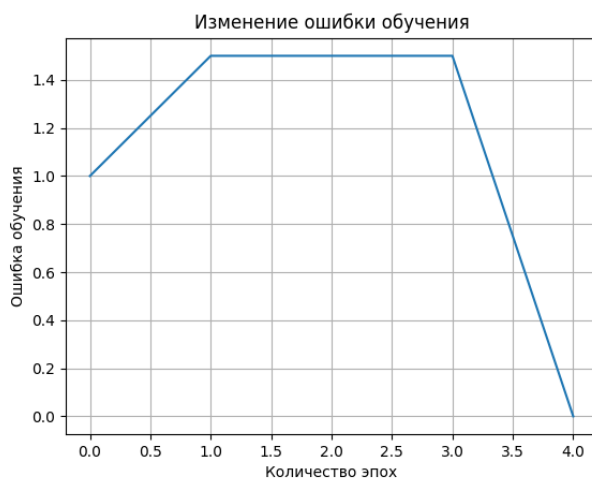
# Побудова графіка процесу навчання
plt.figure()
plt.plot(error_progress)
plt.xlabel('Количество эпох')
plt.ylabel('Ошибка обучения')
plt.title('Изменение ошибки обучения')
plt.grid()
plt.show()
```

		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. Ю.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання:



Графік №1. Вхідні дані.



Графік №2. Класифікування вхідних даних.

Висновок: Другий графік показує зміну помилки навчання впродовж епох під час навчання перцептрону. З часом помилка навчання зменшується і стає близькою до нуля. Це означає, що перцептрон зміг навчитися правильно класифікувати вхідні дані.

		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. Ю.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

Завдання 2.4. Побудова одношарової нейронної мережі

Створіть одношарову нейронну мережу, що складається з незалежних нейронів, для вхідного файлу `data_simple_nn.txt`.

Лістинг програми:

```
import numpy as np
import matplotlib.pyplot as plt
import neurolab as nl

# Завантаження вхідних даних
text = np.loadtxt('data_simple_nn.txt')

# Поділ точок даних та міток
data = text[:, 0:2]
labels = text[:, 2:]

# Побудова графіка вхідних даних
plt.figure()
plt.scatter(data[:, 0], data[:, 1])
plt.xlabel('Размерность 1 ')
plt.ylabel('Размерность 2')
plt.title('Входные данные')

# Мінімальне та максимальне значення для кожного виміру
dim1_min, dim1_max = data[:, 0].min(), data[:, 0].max()
dim2_min, dim2_max = data[:, 1].min(), data[:, 1].max()

# Визначення кількості нейронів у вихідному шарі
num_output = labels.shape[1]

# Визначення одношарової нейронної мережі
dim1 = [dim1_min, dim1_max]
dim2 = [dim2_min, dim2_max]
nn = nl.net.newp([dim1, dim2], num_output)

# Навчимо мережу на тренувальних даних
error_progress = nn.train(data, labels, epochs=100, show=20, lr=0.03)

# Побудова графіка просування процесу навчання
plt.figure()
plt.plot(error_progress)
plt.xlabel('Количество эпох')
plt.ylabel('Ошибка обучения')
plt.title('Изменение ошибки обучения')
plt.grid()
plt.show()

# Виконання класифікатора на тестових точках даних
print('\n Test results:')
data_test = [[0.4, 4.3], [4.4, 0.6], [4.7, 8.1]]
for item in data_test:
    print(item, '-->', nn.sim([item])[0])
```

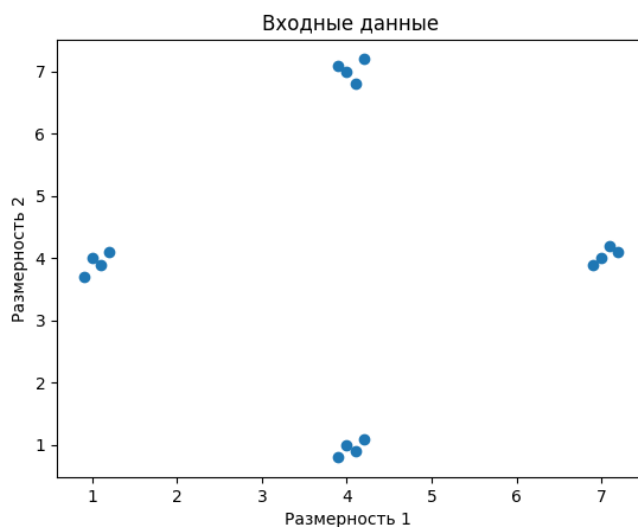
		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. Ю.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання:

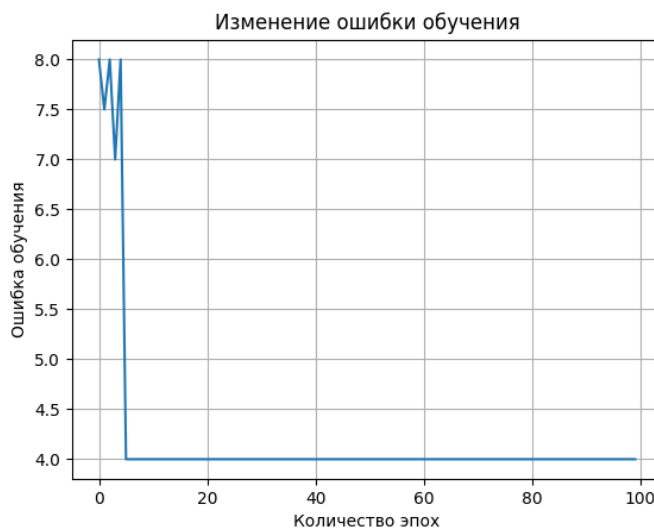
```
sys.path.extend(['C:\\Users\\franc\\OneDrive\\Desktop\\labs0AI\\lab5', 'C:/Users/franc/OneDrive/Desktop/labs0AI/lab5'])

Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)]
Epoch: 20; Error: 4.0;
Epoch: 40; Error: 4.0;
Epoch: 60; Error: 4.0;
Epoch: 80; Error: 4.0;
Epoch: 100; Error: 4.0;
The maximum number of train epochs is reached

Test results:
[0.4, 4.3] --> [0. 0.]
[4.4, 0.6] --> [1. 0.]
[4.7, 8.1] --> [1. 1.]
```



Графік №1. Вхідні дані.



Графік №2. Класифікування вхідних даних.

Висновок: Другий графік відображає зміну помилки під час навчання нейронної мережі. Рівень помилки зменшився з 8.0 до 4.0 (приблизно за 5 епох), але значення 4.0 не змінилось за 100 епох, можна зробити висновок, що мережа

		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. Ю.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

здатна вдосконалити своє рішення під час навчання, але вона не досягла задовільної точності класифікації даних.

З тестових результатів можна зробити висновок, що мережа видає виведення [0, 0] для першої тестової точки, [1, 0] для другої точки і [1, 1] для третьої точки. Ці виведення вказують на класифікацію точок, але вони є некоректними, оскільки вони не відповідають очікуваним класам.

Завдання 2.5. Побудова багатошарової нейронної мережі

Лістинг програми:

```
import numpy as np
import matplotlib.pyplot as plt
import neurolab as nl

# Генерація тренувальних даних
min_val = -15
max_val = 15
num_points = 130
x = np.linspace(min_val, max_val, num_points)
y = 3 * np.square(x) + 5
y /= np.linalg.norm(y)

# Створення даних та міток
data = x.reshape(num_points, 1)
labels = y.reshape(num_points, 1)

# Побудуємо графік вхідних даних.
plt.figure()
plt.scatter(data, labels)
plt.xlabel('Размерность 1 ')
plt.ylabel('Размерность 2')
plt.title('Входные данные')

# Вихідний шар складається з одного нейрона.
nn = nl.net.newff([[min_val, max_val]], [10, 6, 1])

# Завдання градієнтного спуску як навчального алгоритму
nn.trainf = nl.train.train_gd

# Тренування нейронної мережі
error_progress = nn.train(data, labels, epochs=2000, show=100, goal=0.01)

# Виконання нейронної мережі на тренувальних даних
output = nn.sim(data)
y_pred = output.reshape(num_points)

# Побудова графіка помилки навчання
plt.figure()
plt.plot(error_progress)
plt.xlabel('Количество эпох')
plt.ylabel('Ошибка обучения')
plt.title('Изменение ошибки обучения')

# Побудова графіка результатів
x_dense = np.linspace(min_val, max_val, num_points * 2)
y_dense_pred = nn.sim(x_dense.reshape(x_dense.size, 1)).reshape(x_dense.size)
plt.figure()
plt.plot(x_dense, y_dense_pred, '-', x, y, '.', x, y_pred, 'p')
plt.title('фактические и прогнозные значения')
plt.show()
```

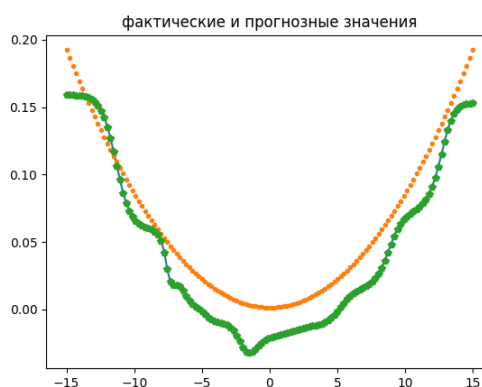
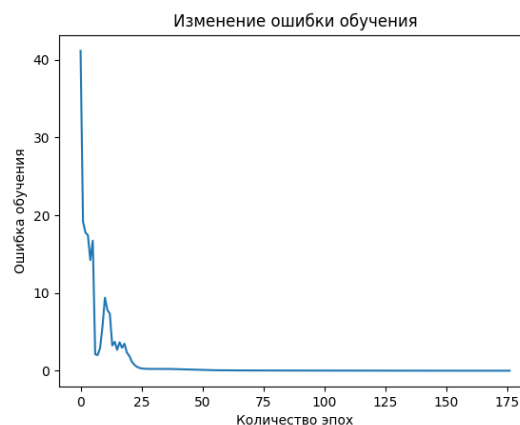
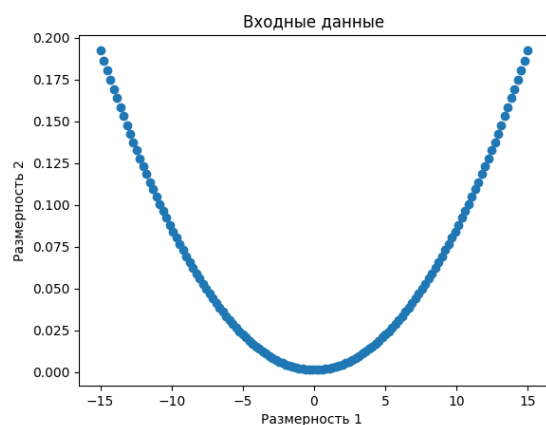
		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. Ю.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

Результат виконання:

```
sys.path.extend(['C:\\Users\\franc\\OneDrive\\Desktop\\labs0AI\\lab5', 'C:/Users/franc/OneDrive/Desktop/labs0AI/lab5'])
```

```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)]
```

```
Epoch: 100; Error: 2.1608077624110766;
Epoch: 200; Error: 0.6988729212574726;
Epoch: 300; Error: 0.3716567937626848;
Epoch: 400; Error: 0.2762075249388611;
Epoch: 500; Error: 0.2538630673921808;
Epoch: 600; Error: 0.2545974714762639;
Epoch: 700; Error: 0.2567640572766321;
Epoch: 800; Error: 0.2530655592766472;
Epoch: 900; Error: 0.2440132712083514;
Epoch: 1000; Error: 0.23124132798518812;
Epoch: 1100; Error: 0.21533678649478175;
Epoch: 1200; Error: 0.1960101406072532;
Epoch: 1300; Error: 0.17284614753493005;
Epoch: 1400; Error: 0.14626215306340254;
Epoch: 1500; Error: 0.11822214110280041;
Epoch: 1600; Error: 0.09182120801932978;
Epoch: 1700; Error: 0.06972884290848076;
Epoch: 1800; Error: 0.052990189563879075;
Epoch: 1900; Error: 0.04115408549587276;
Epoch: 2000; Error: 0.033100694757080634;
The maximum number of train epochs is reached
```



Висновок: В терміналі було показано навчання мережі(номер епохи та значення її помилки),навчання відбулось протягом 2000 епох, найкраща досягнута помилка становила приблизно 0.0331(при початковій – 2.16).

		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. Ю.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2.6. Побудова багат шарової нейронної мережі для свого варіанту

№ варіанта	Тестові дані
Варіант 8	$y = 3x^2 + 8$

Номер варіанта	Багат шаровий перцептрон	
	Кількість шарів	Кількості нейронів у шарах
8	3	5-5-1

Лістинг програми:

```
import numpy as np
import matplotlib.pyplot as plt
import neurolab as nl

# Генерація тренувальних даних
min_val = -15
max_val = 15
num_points = 130
x = np.linspace(min_val, max_val, num_points)
y = 3 * np.square(x) + 8
y /= np.linalg.norm(y)

# Створення даних та міток
data = x.reshape(num_points,1)
labels = y.reshape(num_points,1)

#Побудуємо графік вхідних даних.
plt.figure()
plt.scatter(data,labels)
plt.xlabel('Размерность 1 ')
plt.ylabel('Размерность 2')
plt.title('Входные данные')

# Вихідний шар складається з одного нейрона.
nn = nl.net.newff([[min_val, max_val]], [5,5,1])

# Завдання градієнтного спуску як навчального алгоритму
nn.trainf = nl.train.train_gd

# Тренування нейронної мереж
error_progress = nn.train(data, labels, epochs=2000, show=100, goal=0.01)

# Виконання нейронної мережі на тренувальних даних
output = nn.sim(data)
y_pred = output.reshape(num_points)

# Побудова графіка помилки навчання
plt.figure()
plt.plot(error_progress)
plt.xlabel('Количество эпох')
plt.ylabel('Ошибка обучения')
plt.title('Изменение ошибки обучения')

# Побудова графіка результатів
x_dense = np.linspace(min_val, max_val, num_points * 2)
```

		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. Ю.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

y_dense_pred = nn.sim(x_dense.reshape(x_dense.size, 1)).reshape(x_dense.size)
plt.figure()
plt.plot(x_dense, y_dense_pred, '-', x, y, '.', x, y_pred, 'p')
plt.title('фактические и прогнозные значения')
plt.show()

```

Результат виконання:

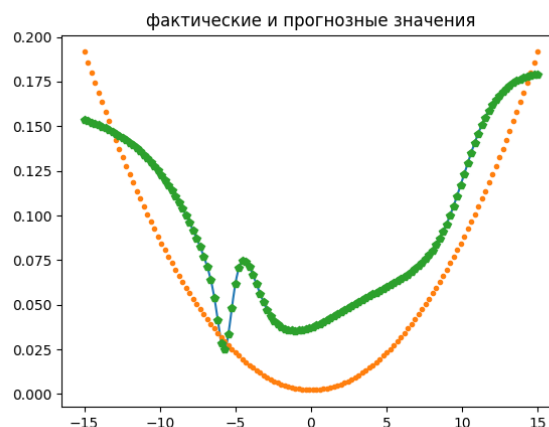
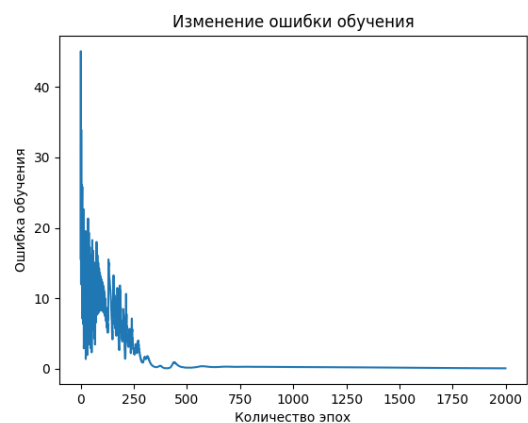
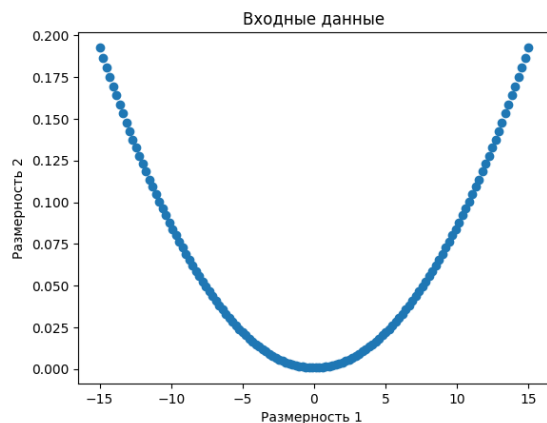
```
sys.path.extend(['C:\\Users\\franc\\OneDrive\\Desktop\\labs0AI\\lab5', 'C:/Users/franc/OneDrive/Desktop/labs0AI/lab5'])
```

```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)]
```

```

Epoch: 100; Error: 12.624911357218284;
Epoch: 200; Error: 6.19794469838179;
Epoch: 300; Error: 1.4843917056795104;
Epoch: 400; Error: 0.08242515595537847;
Epoch: 500; Error: 0.16194151930675726;
Epoch: 600; Error: 0.30117328305325347;
Epoch: 700; Error: 0.30285187636694416;
Epoch: 800; Error: 0.28988842570953566;
Epoch: 900; Error: 0.2740793551988143;
Epoch: 1000; Error: 0.25655495793776634;
Epoch: 1100; Error: 0.24114451211721338;
Epoch: 1200; Error: 0.22656685294209553;
Epoch: 1300; Error: 0.2100430894894728;
Epoch: 1400; Error: 0.1905609255138166;
Epoch: 1500; Error: 0.16842099291820653;
Epoch: 1600; Error: 0.14484886917387724;
Epoch: 1700; Error: 0.12178145070189894;
Epoch: 1800; Error: 0.10115887456887097;
Epoch: 1900; Error: 0.0841197426459148;
Epoch: 2000; Error: 0.07081787104948606;
The maximum number of train epochs is reached

```



		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. ІО.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновок: В терміналі було показано навчання мережі(номер епохи та значення її помилки),навчання відбулось протягом 2000 епох, найкраща досягнута помилка становила приблизно 0.070(при початковій – 12.62).

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

Лістинг програми:

```
import numpy as np
import neurolab as nl
import numpy.random as rand
import pylab as pl
skv = 0.05
centr = np.array([[0.2, 0.2], [0.4, 0.4], [0.7, 0.3], [0.2, 0.5]])
rand_norm = skv * rand.randn(100, 4, 2)
inp = np.array([centr + r for r in rand_norm])
inp.shape = (100 * 4, 2)
rand.shuffle(inp)

#Create net with 2 inputs and 4 neurons
net = nl.net.newc([[0.0, 1.0],[0.0, 1.0]], 4)
# train with rule: Conscience Winner Take All algorithm (CWTA)
error = net.train(inp, epochs=200, show=20)

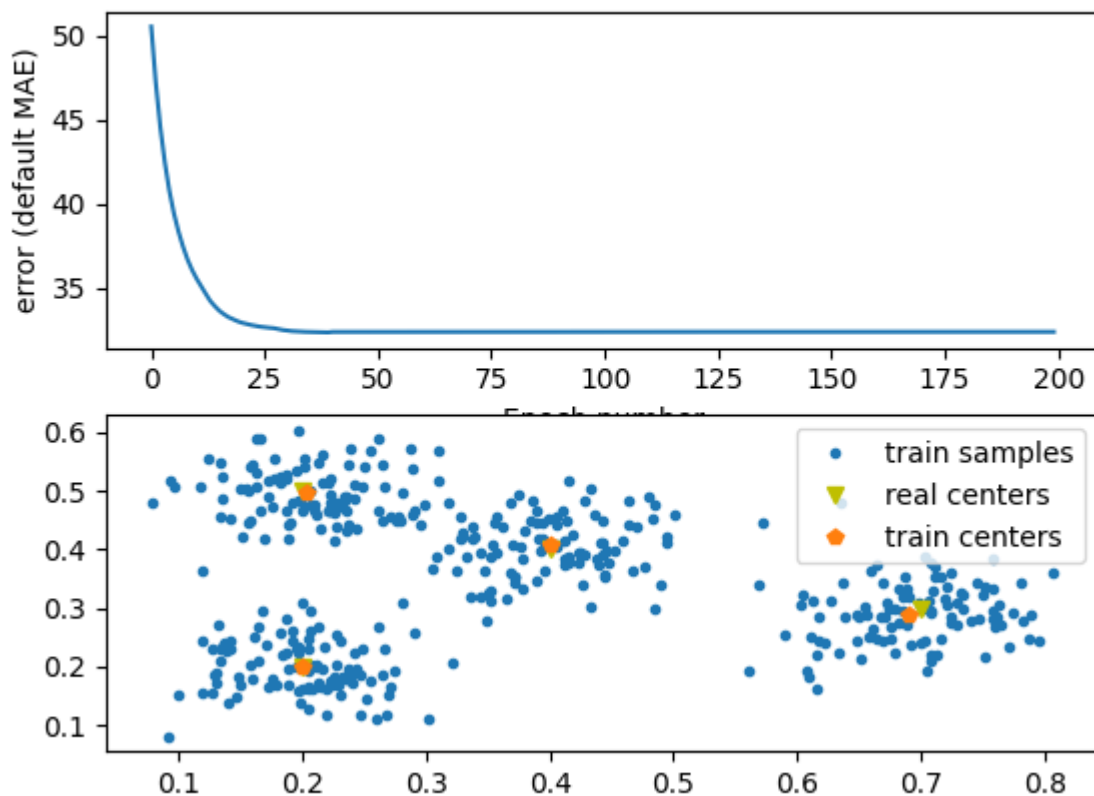
# Plot results:
pl.title('Classification Problem')
pl.subplot(211)
pl.plot(error)
pl.xlabel('Epoch number')
pl.ylabel('error (default MAE)')
w = net.layers[0].np['w']
pl.subplot(212)
pl.plot(inp[:,0], inp[:,1], '.',\
        centr[:,0], centr[:,1], 'yv',\
        w[:,0], w[:,1], 'p')
pl.legend(['train samples', 'real centers', 'train centers'])
pl.show()
```

Результат виконання:

```
sys.path.extend(['C:\\Users\\franc\\OneDrive\\Desktop\\\\labs0AI\\lab5', 'C:/Users/franc/OneDrive/Desktop/labs0AI/lab5'])

Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)]
Epoch: 20; Error: 33.032468604201384;
Epoch: 40; Error: 32.366723868744685;
Epoch: 60; Error: 32.39803155095098;
Epoch: 80; Error: 32.39935103246758;
Epoch: 100; Error: 32.399630796638476;
Epoch: 120; Error: 32.39968370122053;
Epoch: 140; Error: 32.39969313155501;
Epoch: 160; Error: 32.399694753931925;
Epoch: 180; Error: 32.39969502665073;
Epoch: 200; Error: 32.39969507176634;
The maximum number of train epochs is reached
```

		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. Ю.				11
Змн.	Арк.	№ докум.	Підпис	Дата		



Висновок: Ми побудували нейронну мережу на основі карти Кохонена, модель не справилась з завданням (не покращувала свою точність), значення через 200 епох зменшилось на 0.633. Помилка MAE – вимірює наскільки середньоквадратична похибка змінюється під час навчання нейронної мережі.

Завдання 2.8. Дослідження нейронної мережі на основі карти Кохонена, що самоорганізується

№ варіанту	Центри кластера	skv
Варіант 8	[0.1, 0.2], [0.4, 0.3], [0.7, 0.3], [0.2, 0.5], [0.5, 0.3]	0,04

Створіть нейронну мережу Кохонена з 2 входами та 4 нейронами

Лістинг програми:

```
import numpy as np
import neurolab as nl
import numpy.random as rand
import pylab as pl
skv = 0.05
centr = np.array([[0.1, 0.2], [0.4, 0.3], [0.7, 0.3], [0.2, 0.5], [0.5, 0.3]])
rand_norm = skv * rand.randn(100, 5, 2)
inp = np.array([centr + r for r in rand_norm])
inp.shape = (100 * 5, 2)
rand.shuffle(inp)

#Create net with 2 inputs and 4 neurons
```

		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. Ю.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

```

net = nl.net.newc([[0.0, 1.0],[0.0, 1.0]], 4)
# train with rule: Conscience Winner Take All algorithm (CWTA)
error = net.train(inp, epochs=200, show=20)

# Plot results:
pl.title('Classification Problem')
pl.subplot(211)
pl.plot(error)
pl.xlabel('Epoch number')
pl.ylabel('error (default MAE)')
w = net.layers[0].np['w']
pl.subplot(212)
pl.plot(inp[:,0], inp[:,1], '.', \
        centr[:,0], centr[:, 1] , 'yv', \
        w[:,0], w[:,1], 'p')
pl.legend(['train samples', 'real centers', 'train centers'])
pl.show()

```

Результат виконання:

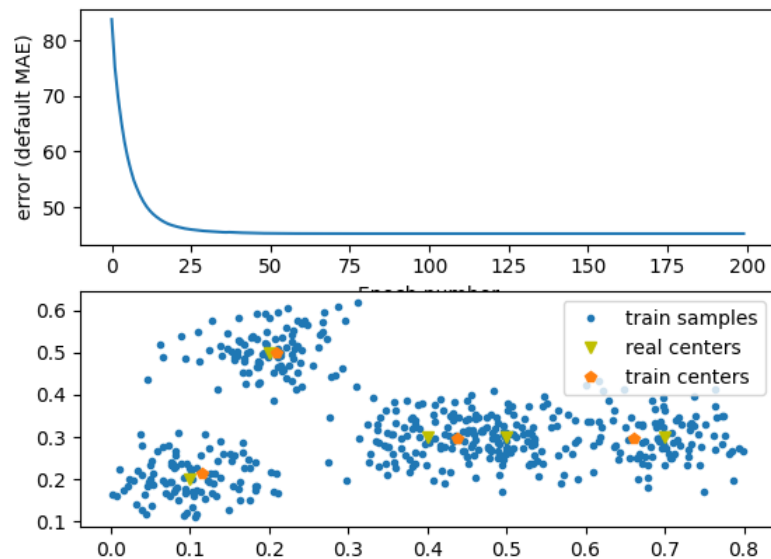
```
sys.path.extend(['C:\\Users\\franc\\OneDrive\\Desktop\\labs0AI\\lab5', 'C:/Users/franc/OneDrive/Desktop/labs0AI/lab5'])
```

```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)]
```

```

Epoch: 20; Error: 46.70276655408858;
Epoch: 40; Error: 45.403123444208845;
Epoch: 60; Error: 45.224667991013334;
Epoch: 80; Error: 45.20893760867882;
Epoch: 100; Error: 45.207459848889506;
Epoch: 120; Error: 45.2073273970846;
Epoch: 140; Error: 45.20731559665046;
Epoch: 160; Error: 45.20731452520573;
Epoch: 180; Error: 45.207314426797076;
Epoch: 200; Error: 45.20731441760479;
The maximum number of train epochs is reached

```



		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. ІО.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Створіть нейронну мережу Кохонена з 2 входами та 5 нейронами

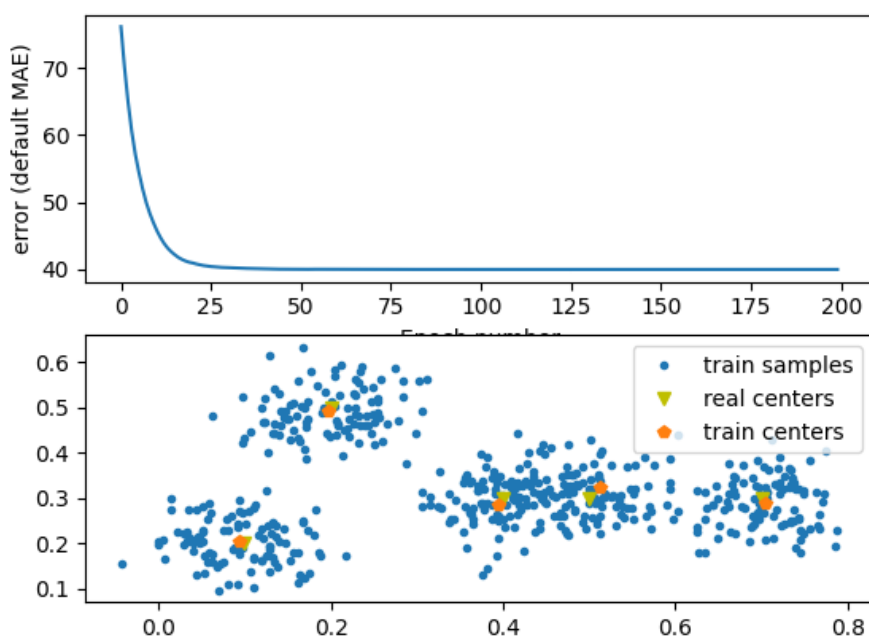
Результат виконання:

```
sys.path.extend(['C:\\Users\\franc\\OneDrive\\Desktop\\labs0AI\\lab5', 'C:/Users/franc/OneDrive/Desktop/labs0AI/lab5'])
```

```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)]
```

```
Epoch: 20; Error: 41.01342382632121;  
Epoch: 40; Error: 40.02608979111954;  
Epoch: 60; Error: 39.96627777405012;  
Epoch: 80; Error: 39.94555762244039;  
Epoch: 100; Error: 39.941471451801156;  
Epoch: 120; Error: 39.940162569365654;  
Epoch: 140; Error: 39.94107584922769;  
Epoch: 160; Error: 39.94048278250279;  
Epoch: 180; Error: 39.94000419459332;  
Epoch: 200; Error: 39.94105042823992;
```

```
The maximum number of train epochs is reached
```



Висновок: Зменшення кількості кластерів при незмінній кількості нейронів може поліпшити точність моделі, оскільки відображає кількість класів або кластерів, які модель намагається розрізнити. У другому випадку (5 нейронів і 4 кластери), зменшення кількості кластерів знизило помилку MAE. Таким чином, вибір кількості нейронів і кластерів повинен враховувати природу даних і завдання, яке ви намагаєтеся вирішити. При 5 кластерах і 5 нейронах було отримано значення MAE – 39.94, при 5 кластерах і 4 нейронах – 45,20. Якщо порівнювати з попереднім завданням, то в цьому нейронна мережа відпрацювала краще, т.к., значення помилки значно зменшилось.

		Іваницький Ф.А.			ДУ «Житомирська політехніка».23.121.8.000 – Лр5	Арк.
		Голенко М. Ю.				14
Змн.	Арк.	№ докум.	Підпис	Дата		