

## Лабораторна робота №1

### Нейронна реалізація логічних функцій AND, OR, XOR

**Мета:** дослідити математичну модель нейрона.

#### Хід роботи:

**Завдання 1.** Реалізувати обчислювальний алгоритм для функції  $\text{xor}(x1, x2)$  через функції  $\text{or}(x1, x2)$  і  $\text{and}(x1, x2)$  в програмному середовищі (C++, Python, та ін.). Для реалізації обчислювальних алгоритмів рекомендується використання онлайн середовищ тестування (наприклад [repl.it](https://repl.it), [trinket](https://trinket.io), і т.д.).

Лістинг програми (реалізація  $\text{xor}$  в площині X):

```
import numpy as np
from matplotlib import pyplot as plt

k = 50
a = 0
b = 1
step = (b-a)/k

# OR:  $x1 + x2 - 0.5 = 0 \rightarrow x2 = -x1 + 0.5$ 
def line_or(x):
    return -x + 0.5

# AND:  $x1 + x2 - 1.5 = 0 \rightarrow x2 = -x1 + 1.5$ 
def line_and(x):
    return -x + 1.5

t = a
mass_X1 = [a]
for i in range(k):
    if t <= b:
        t = t + step
        mass_X1.append(t)
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр1						
Змн.	Арк.	№ докум.	Підпис	Дата	<div>Звіт з лабораторної роботи №1</div>						
Розроб.		Козлик С.О.									
Перевір.		Маєвський О.В.									
Керівник											
Н. контр.											
Зав. каф.											
					Літ.	Арк.	Аркушів				
							1	7			
					ФІКТ, гр. ІПЗ-22-2						

## Лістинг програми (продовження):

```
# OR лінія
mass_X2_or = []
mass_X1_0_or = []
for j in range(k):
    f = line_or(mass_X1[j])
    if f >= 0:
        mass_X2_or.append(f)
        mass_X1_0_or.append(mass_X1[j])

# AND лінія
mass_X2_and = []
mass_X1_0_and = []
for j in range(k):
    f = line_and(mass_X1[j])
    if f >= 0:
        mass_X2_and.append(f)
        mass_X1_0_and.append(mass_X1[j])

# Побудова двох прямих
plt.plot(mass_X1_0_or, mass_X2_or, c='red', marker='*', label="OR boundary")
plt.plot(mass_X1_0_and, mass_X2_and, c='red', marker='*', label="AND boundary")

mass_test_date = []
len_mass_test = 700
long = 1
for i in range(len_mass_test):
    x = 0
    vnutr_mass = []
    while x <= long:
        g = np.random.random()
        vnutr_mass.append(g)
        x += 1
    mass_test_date.append(vnutr_mass)

def AND(x1, x2):
    return 1 if x1 + x2 - 1.5 > 0 else 0

def OR(x1, x2):
    return 1 if x1 + x2 - 0.5 > 0 else 0

# XOR через OR і AND
def classify(x1, x2):
    y1 = OR(x1, x2)
    y2 = AND(x1, x2)
    return AND(y1, 1 - y2)

for x, y in mass_test_date:
    color = 'green' if classify(x, y) == 1 else 'blue'
```

		Козлик С.О.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр1	Арк.
		Маєвський О.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```
plt.scatter(x, y, c=color, alpha=0.5)

plt.title("Дискримінантна пряма XOR та тестові точки")
plt.xlabel("x1")
plt.ylabel("x2")
plt.grid(True)
plt.show()
```

Результат виконання програми:

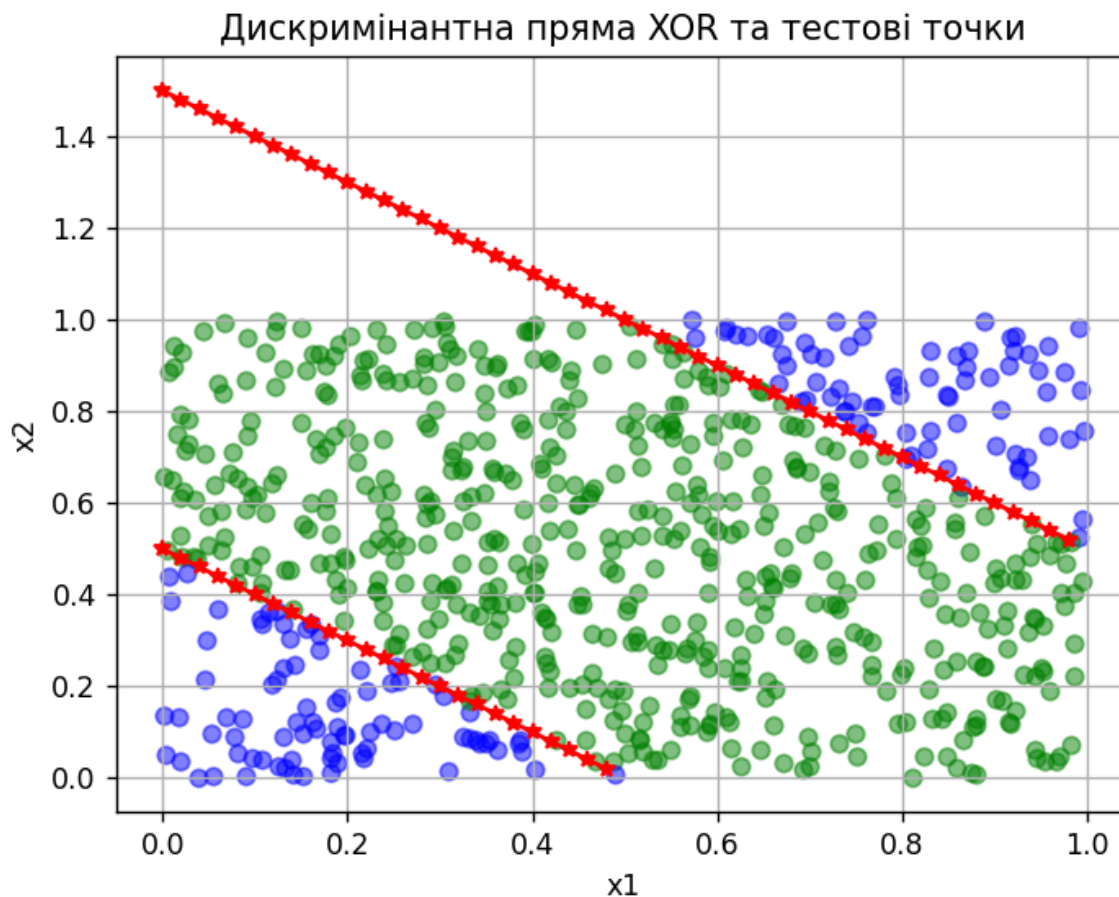


Рис. 1. Результат виконання програми

Лістинг програми (реалізація хог в площині Y):

```
import numpy as np
from matplotlib import pyplot as plt

k = 50
a = 0
b = 1
step = (b-a)/k

# OR:  $x_1 + x_2 - 0.5 = 0 \rightarrow x_2 = -x_1 + 0.5$ 
def line_or(x):
    return -x + 0.5

# AND:  $x_1 + x_2 - 1.5 = 0 \rightarrow x_2 = -x_1 + 1.5$ 
def line_and(x):
```

		Козлик С.О.		
		Маєвський О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```

        return -x + 1.5

# AND:  $y1 - y2 - 0.5 = 0 \rightarrow y2 = y1 - 0.5$ 
def line_xor(y):
    return y - 0.5

t = a
mass_X1 = [a]
for i in range(k):
    if t <= b:
        t = t + step
        mass_X1.append(t)

# OR лінія
mass_X2_or = []
mass_X1_0_or = []
for j in range(k):
    f = line_or(mass_X1[j])
    if f >= 0:
        mass_X2_or.append(f)
        mass_X1_0_or.append(mass_X1[j])

# AND лінія
mass_X2_and = []
mass_X1_0_and = []
for j in range(k):
    f = line_and(mass_X1[j])
    if f >= 0:
        mass_X2_and.append(f)
        mass_X1_0_and.append(mass_X1[j])

mass_X2_xor = []
mass_X1_0_xor = []
for j in range(k):
    f = line_xor(mass_X1[j])
    if f >= 0:
        mass_X2_xor.append(f)
        mass_X1_0_xor.append(mass_X1[j])

# Побудова прямої
plt.plot(mass_X1_0_xor, mass_X2_xor, c='red', marker='*', label="XOR boundary")

mass_test_date = []
len_mass_test = 700
long = 1
for i in range(len_mass_test):
    x = 0
    vnutr_mass = []
    while x <= long:
        g = np.random.random()

```

		Козлик С.О.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр1	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

    vnutr_mass.append(g)
    x += 1
    mass_test_date.append(vnutr_mass)

def AND(x1, x2):
    return 1 if x1 + x2 - 1.5 > 0 else 0

def OR(x1, x2):
    return 1 if x1 + x2 - 0.5 > 0 else 0

def XOR(y1, y2):
    return 1 if y1 - y2 - 0.5 > 0 else 0

# XOR через OR і AND
def classify(x1, x2):
    y1 = OR(x1, x2)
    y2 = AND(x1, x2)
    return AND(y1, 1 - y2)

mass_X1_class_A = []
mass_X2_class_A = []
mass_X1_class_B = []
mass_X2_class_B = []

def Klassification(x1, x2):
    if classify(x1, x2):
        mass_X1_class_A.append(x1)
        mass_X2_class_A.append(x2)
    else:
        mass_X1_class_B.append(x1)
        mass_X2_class_B.append(x2)

for x1, x2 in mass_test_date:
    Klassification(x1, x2)

# Візуалізація
for x, y in zip(mass_X1_class_A, mass_X2_class_A):
    color = 'green' if XOR(x, y) == 1 else 'blue'
    plt.scatter(x, y, c=color, alpha=0.5)

for x, y in zip(mass_X1_class_B, mass_X2_class_B):
    plt.scatter(x, y, c='blue', alpha=0.5)

plt.title("Дискримінантна пряма XOR та тестові точки")
plt.xlabel("y1")
plt.ylabel("y2")
plt.grid(True)
plt.show()

```

		Козлик С.О.			ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.13.000 – Лр1	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

Результат виконання програми:

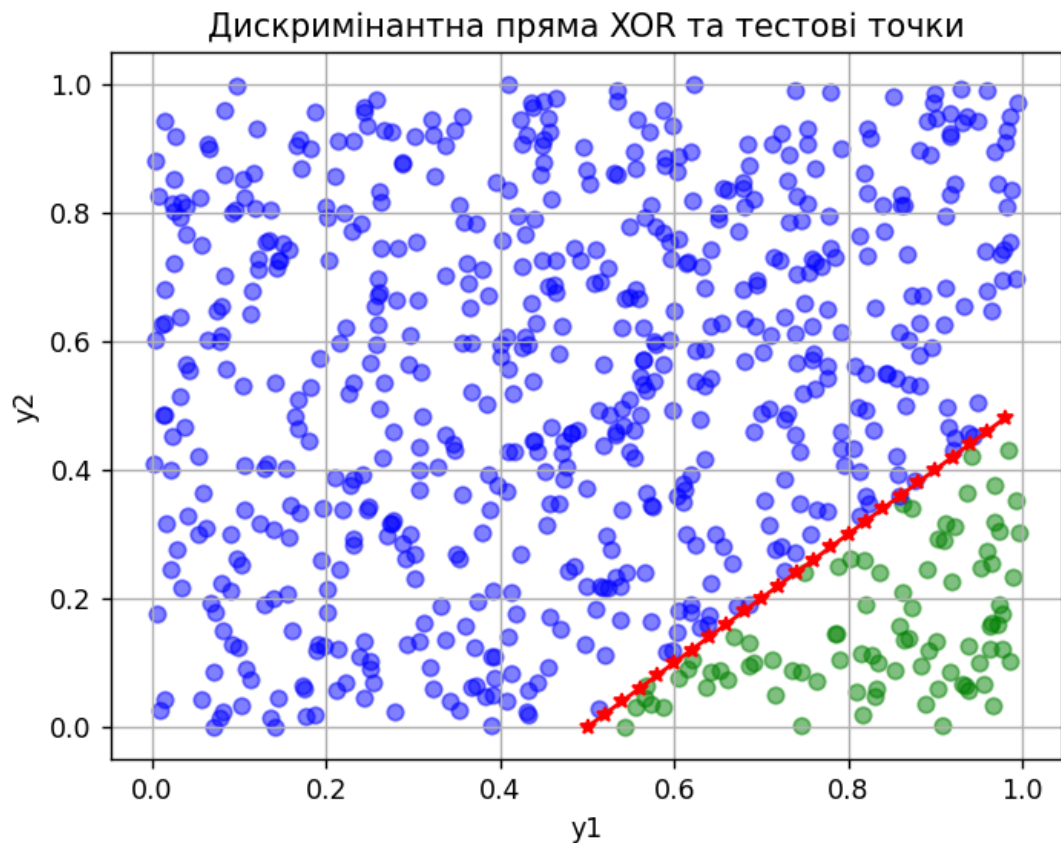


Рис. 2. Результат виконання програми

**Завдання 2.** Зобразити двошлойний персептрон для функції  $\text{xor}(x_1, x_2)$  та скласти відповідне рівняння розділяючої прямої, використовуючи теоретичний матеріал даної лабораторної роботи.

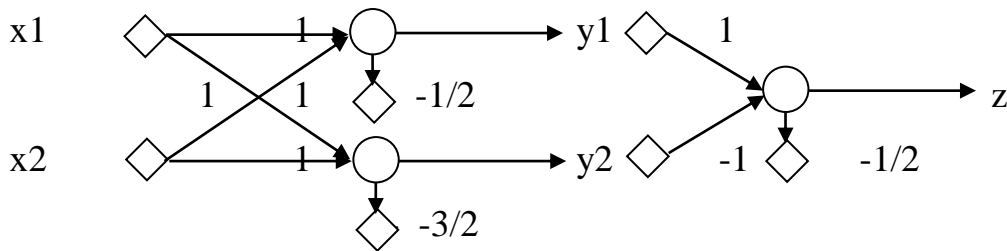
Побудова нелінійного класифікатора функції  $\text{xor}(x_1, x_2)$ . Нехай на виході персептрона для функції  $\text{or}(x_1, x_2) - y_1$ , а на виході персептрона  $\text{and}(x_1, x_2) - y_2$ .

Таблиця істинності для  $\text{xor}$ :

$x_1$	$x_2$	OR ( $y_1$ )	AND ( $y_2$ )	XOR
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

З таблиці істинності можна визначити XOR через OR і AND наступним чином:

$XOR = OR - AND$ , тобто  $XOR = y1 - y2 = x1 + x2 - 0,5 - (x1 + x2 - 1,5) = x1 + x2 - 0,5 - x1 - x2 + 1,5 = 1$ . Нормалізуємо різницю двох рівнянь меж (від OR до AND), поділивши її на 2. Тоді розподільчою лінією є  $y1 - y2 = 1/2$ . Відповідний двохслойний персептрон має вигляд:



Висновок: в ході виконання лабораторної роботи ми дослідили математичну модель нейрона.