

## Лабораторна робота №1

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

*Мета:* Дослідити математичну модель нейрона.

Github - <https://github.com/Oleksi89/ai-systems>

#### 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 (Python).

*Лістинг програми:*

```
import numpy as np

def step_function(x):
    """Функція активації: 1 якщо x > 0, інакше 0"""
    return 1 if x > 0 else 0

def or_neuron(x1, x2):
    """
    Реалізація функції OR.
    Розділяюча лінія:  $x1 + x2 = 0.5$ 
    Ваги:  $w1=1, w2=1, \text{bias}=-0.5$ 
    """
    weights = np.array([1, 1])
    inputs = np.array([x1, x2])
    bias = -0.5

    # Суматор:  $w1*x1 + w2*x2 + \text{bias}$ 
    total = np.dot(weights, inputs) + bias
    return step_function(total)

def and_neuron(x1, x2):
    """
    Реалізація функції AND.
    Розділяюча лінія:  $x1 + x2 = 1.5$ 
    Ваги:  $w1=1, w2=1, \text{bias}=-1.5$ 
    """
    weights = np.array([1, 1])
    inputs = np.array([x1, x2])
    bias = -1.5

    total = np.dot(weights, inputs) + bias
    return step_function(total)
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.5.000 – Лр.1		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Васянович О.А.			Звіт з лабораторної роботи №1	Літ.	Арк.
Перевір.		Маєвський О.Є.					Аркушів
Реценз.							1
Н. Контр.						ФІКТ, гр. ІПЗ-22-3	
Зав.каф.							

```
def xor_neuron(x1, x2):
    """
    Реалізація XOR через OR та AND.
    Вхідні дані: y1 (від OR) та y2 (від AND).
    Розділяюча лінія (y1, y2): y1 - y2 = 0.5
    Ваги: w_or=1, w_and=-1, bias=-0.5
    """

    # Отримуємо сигнали від попереднього шару
    y1 = or_neuron(x1, x2)
    y2 = and_neuron(x1, x2)

    # Обчислюємо вихід XOR
    inputs = np.array([y1, y2])
    weights = np.array([-1, 1]) # Вага -1 для OR, 1 для AND
    bias = 0.5

    total = np.dot(weights, inputs) + bias
    return step_function(total)

# --- Тестування ---
print(f'{"x1":<5} {"x2":<5} | {"OR":<5} {"AND":<5} | {"XOR (Result)":<10}')
print("-" * 45)

test_cases = [(0, 0), (0, 1), (1, 0), (1, 1)]

for x1, x2 in test_cases:
    res_or = or_neuron(x1, x2)
    res_and = and_neuron(x1, x2)
    res_xor = xor_neuron(x1, x2)

    print(f'{"x1":<5} {"x2":<5} | {res_or:<5} {res_and:<5} | {res_xor:<10}')
```

*Результати:*

...	x1	x2	OR	AND	XOR (Result)
	0	0	0	0	0
	0	1	1	0	1
	1	0	1	0	1
	1	1	1	1	0

Рис.1

**Завдання 2.** Зобразити двохслойний персептрон для функції  $\text{xor}(x1, x2)$  та скласти відповідне рівняння розділяючої прямої, використовуючи теоретичний матеріал даної лабораторної роботи.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.5.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

### Лістинг програми:

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

# Дані: вхідні вектори та очікуваний результат
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
y = [0, 1, 1, 0]

plt.figure(figsize=(8, 6))
x = np.linspace(-0.5, 1.5, 100)

# 1. Розділяючі прямі
# Рівняння OR:  $x_1 + x_2 = 0.5 \rightarrow x_2 = 0.5 - x_1$ 
plt.plot(x, 0.5 - x, 'g-', linewidth=2, label='OR boundary')

# Рівняння AND:  $x_1 + x_2 = 1.5 \rightarrow x_2 = 1.5 - x_1$ 
plt.plot(x, 1.5 - x, 'r-', linewidth=2, label='AND boundary')

# 2. Зафарбовування областей класів
# Клас 1 (XOR=1): між прямими
plt.fill_between(x, 0.5 - x, 1.5 - x, color='lightgreen', alpha=0.5, label='Class 1(XOR = 1)')
# Клас 0 (XOR=0): зовні
plt.fill_between(x, -1, 0.5 - x, color='salmon', alpha=0.2, label='Class 0(XOR = 0)')
plt.fill_between(x, 1.5 - x, 2.5, color='salmon', alpha=0.2)

# 3. Відображення точок
for i, point in enumerate(X):
    color = 'green' if y[i] == 1 else 'red'
    marker = 's' if y[i] == 1 else 'o' # квадрат для 1, коло для 0

    plt.scatter(point[0], point[1], c=color, marker=marker, s=150, edgecolors='k', zorder=5)
    plt.text(point[0] + 0.05, point[1], f"({point[0]}, {point[1]})", fontsize=10)

# Оформлення
plt.xlim(-0.5, 1.5)
plt.ylim(-0.5, 1.5)
plt.xlabel('x1')
plt.ylabel('x2')
plt.title('XOR Decision Boundaries')
plt.grid(True)
plt.legend(loc='upper right')
plt.show()
```

### Результати:

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.5.000 – Лр.1	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

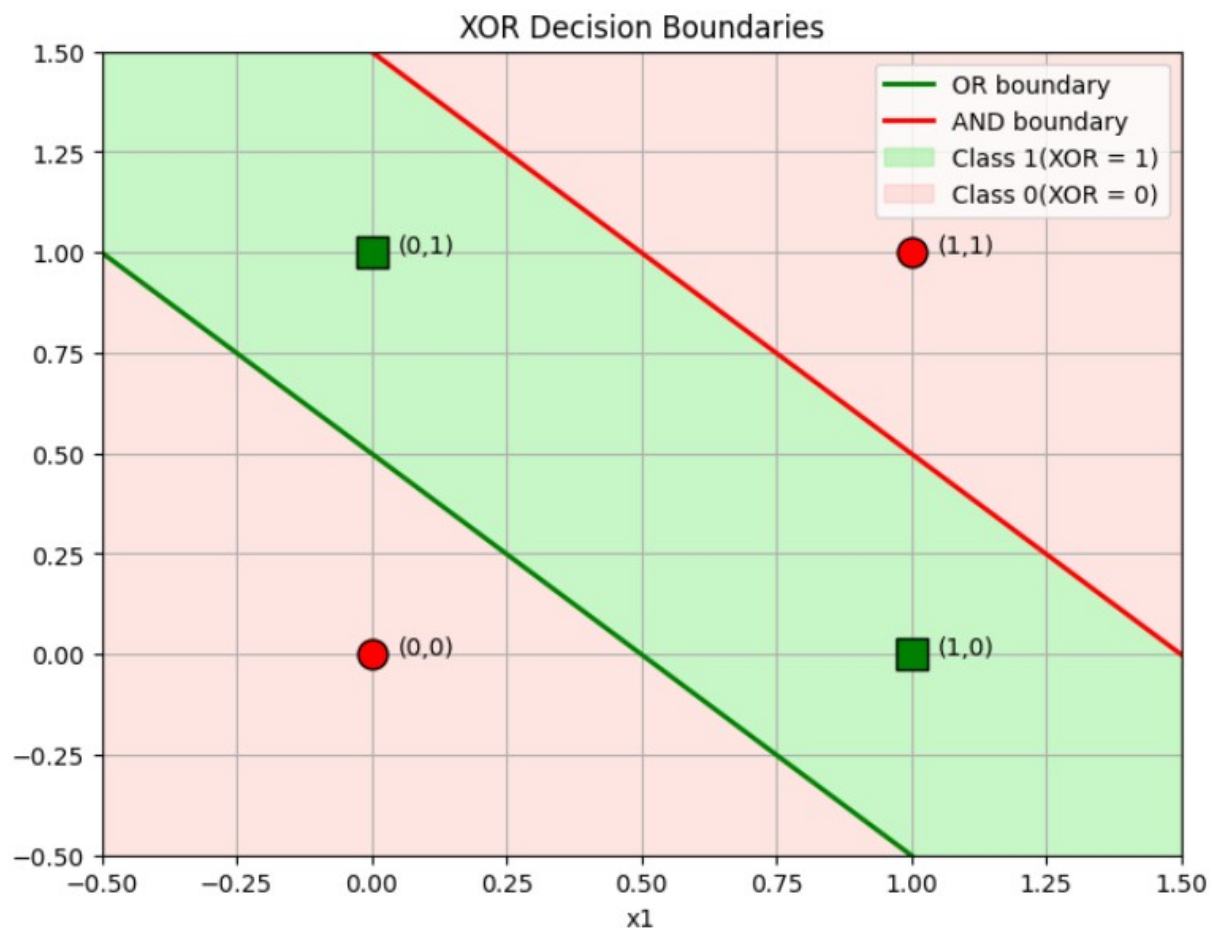


Рис.2

Згідно з теоретичними відомостями, нейрон описується дискримінантною функцією:

$$g(x) = \sum_{i=1}^n W_i \cdot x_i + W_0$$

та пороговою функцією активації:

$$f(v) = \begin{cases} 0, & \text{при } v < 0; \\ 1, & \text{при } v > 0. \end{cases}$$

Для вирішення проблеми XOR використовується двошаровий персептрон, що формує дві розділяючі лінії у вхідному просторі ( $x_1, x_2$ ):

Нейрон OR ( $y_1$ ): Реалізує нижню межу.

Ваги:  $W_1 = 1, W_2 = 1$ .

Поріг:  $W_0 = -0.5$ .

Рівняння прямої:  $x_1 + x_2 - 0.5 = 0 \Rightarrow x_1 + x_2 = 0.5$

Нейрон NAND (y2): Реалізує верхню межу (інверсія AND)

Ваги:  $W1 = -1$ ,  $W2 = -1$

Поріг:  $W0 = 1.5$

Рівняння прямої:  $-x1 - x2 + 1.5 = 0 \Rightarrow x1 + x2 = 1.5$

Вихідний нейрон (NAND) об'єднує сигнали прихованого шару. Простір перетворюється з координат (x1, x2) у координати (y1, y2).

Таблиця 1

x1	x2	y2(NAND)	Клас $\Omega$ (XOR)
0	0	1	$\Omega_0(0)$
0	1	1	$\Omega_1(1)$
1	0	1	$\Omega_1(1)$
1	1	0	$\Omega_0(0)$

У просторі виходів (y1, y2) клас  $\Omega_1$  (де XOR=1) відповідає точці (1, 1).

Розділяюча функція вихідного нейрона має вигляд:

$$g = 1*y1 + 1*y2 - 1.5$$

Це означає, що нейрон активується лише тоді, коли виконуються обидві умови першого шару:

$y1 = 1$  (точка вище прямої OR)

$y2 = 1$  (точка нижче прямої NAND)

Геометрично це відповідає області (“коридору”) між прямими  $x1 + x2 = 0.5$  та  $x1 + x2 = 1.5$ , що зображено на графіку(рис.2).

Висновок: у цій лабораторній роботі я дослідив модель нейрона, реалізацію функцій xor, and, nand, or.