

## ЛАБОРАТОРНА РОБОТА №1

Тема: 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:

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

# Функція активації
def step_function(x):
    return 1 if x >= 0 else 0

# Персептрон для функції OR
def or_perceptron(x1, x2):
    weights = np.array([1, 1]) # Ваги
    threshold = -0.5 # Поріг
    inputs = np.array([x1, x2]) # Вхідні дані
    linear_combination = np.dot(weights, inputs) + threshold # Лінійна комбінація
    return step_function(linear_combination)

# Персептрон для функції AND
def and_perceptron(x1, x2):
    weights = np.array([1, 1]) # Ваги
    threshold = -1.5 # Поріг
    inputs = np.array([x1, x2]) # Вхідні дані
    linear_combination = np.dot(weights, inputs) + threshold # Лінійна комбінація
    return step_function(linear_combination)

# Персептрон для функції XOR через OR та AND
def xor_perceptron(x1, x2):
    or_result = or_perceptron(x1, x2)
    and_result = and_perceptron(x1, x2)
    return step_function(or_result - and_result) # XOR через OR та NOT AND

# Генерація випадкових точок
np.random.seed(42) # Для відтворюваності результатів
num_points = 200 # Кількість точок
x_random = np.random.rand(num_points) * 2 - 0.5 # Генерація значень від -0.5 до 1.5
```

					ДУ «Житомирська політехніка».24.121.16.000 – Лр1			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Некритий В.Ю.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Іванов Д.А.						1
Керівник							Аркушів	
Н. контр.							4	
Зав. каф.							ФІКТ Гр. ІПЗ-21-5	

```

y_random = np.random.rand(num_points) * 2 - 0.5 # Генерація значень від -0.5 до 1.5

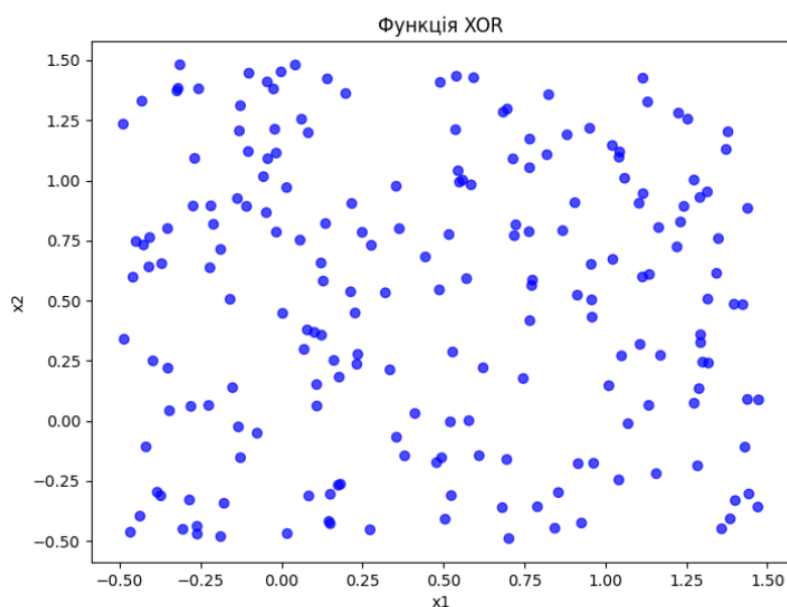
# Класифікація точок за допомогою XOR
xor_results = np.array([xor_perceptron(x, y) for x, y in zip(x_random, y_random)])

# Встановлення кольорів: синій для 1, оранжевий для 0
colors_xor = ['blue' if result == 1 else 'orange' for result in xor_results]

# Побудова графіку
plt.figure(figsize=(8, 6))
plt.scatter(x_random, y_random, c=colors_xor, alpha=0.7)
plt.title('Функція XOR')
plt.xlabel('x1')
plt.ylabel('x2')
plt.show()

```

Результат виконання програми:



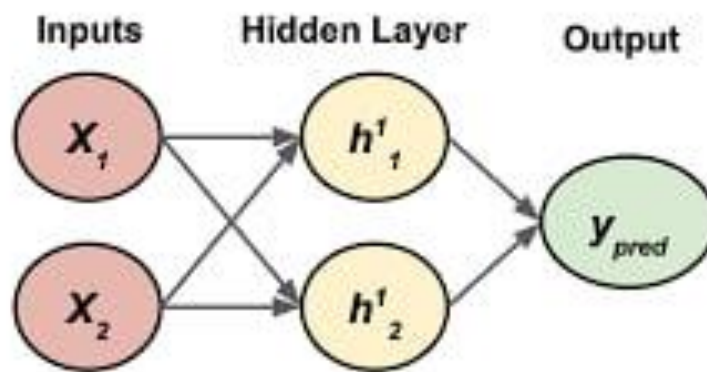
Розглянемо код детальніше: у коді реалізовано обчислювальний алгоритм для функції XOR, що базується на комбінації логічних функцій OR і AND. Спочатку генеруються випадкові точки у двовимірному просторі, де координати  $x_1$  і  $x_2$  знаходяться в діапазоні від -0.5 до 1.5. Початкові значення генератора випадкових чисел зафіксовано для забезпечення відтворюваності результатів. Далі для кожної точки обчислюються значення функцій OR і AND, після чого реалізується функція XOR шляхом комбінації  $OR(x_1, x_2)$  та  $NOT AND(x_1, x_2)$ . Для візуалізації роботи алгоритму точки класифікуються: синім кольором позначаються точки, які належать до класу 1 (істина), а оранжевим — точки класу 0 (хибність). Результати класифікації представлені на графіку, що відображає роботу функції XOR у двовимірному просторі. Таким чином, завдання виконано з демонстрацією роботи алгоритму через класифікацію точок та їх візуалізацію.

		Некритий В.Ю.			ДУ «Житомирська політехніка».24.121.16.000 – Пр1	Арк.
		Іванов Д.А.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Завдання 2. Зобразити двохшаровий персептрон для функції хог( $x_1, x_2$ ) та складіть відповідне рівняння розділяючої прямої, використовуючи теоретичний матеріал даної лабораторної роботи.

Функцію XOR не можна розділити однією прямою у просторі двох змінних, як це робиться для OR або AND, оскільки точки істинності та хибності розташовані нестандартно. Лінійний класифікатор не зможе виконати поділ класів для всіх комбінацій. Для вирішення використовується двохшаровий персептрон. У прихованому шарі обчислюються проміжні значення через функції AND і OR, що дозволяє створити дві розділяючі лінії. Ці лінії визначають області, відповідні істинним та хибним значенням.

$$\text{XOR}(x_1, x_2) = \text{OR}(\text{AND}(x_1, \text{NOT}(x_2)), (\text{AND}(\text{NOT}(x_1), x_2)))$$



Персептрон функції XOR.

У результаті прихований шар формує дві прямі для поділу простору, а вихідний шар поєднує ці результати, отримуючи функцію XOR. Це дозволяє правильно класифікувати всі комбінації вхідних даних.

Розглянемо першу лінію для  $h_1$ :

$$h_1 = \sigma((-1) * x_1 + x_2 + 1)$$

Лінійне рівняння для цієї межі (після перетворення сигмоїдної функції для порогу активації 0.5) можна виразити як:

$$-x_1 + x_2 + 1 = 0$$

або

		Некритий В.Ю.			ДУ «Житомирська політехніка».24.121.16.000 – Пр1	Арк.
		Іванов Д.А.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

$$x_2 = x_1 - 1.$$

Тепер визначимо другу пряму для  $h_2$ :

$$h_2 = \sigma(x_1 + (-1) * x_2 + 1)$$

Лінійне рівняння цієї межі можна записати так:

$$x_1 - x_2 + 1 = 0$$

або

$$x_2 = x_1 + 1.$$

**Висновки:** дізнався, що логічні функції AND і OR можна реалізувати за допомогою простого перцептрона, а для функції XOR потрібна двошарова нейронна мережа. Це показує, як приховані шари допомагають розв'язувати задачі з нелінійними залежностями.

		Некритий В.Ю.			ДУ «Житомирська політехніка».24.121.16.000 – Лр1	Арк.
		Іванов Д.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		4