**Лабораторна робота №1**

**Пархомчук Іван ІПЗ-21-5**

**Мета:** Дослідити математичну модель нейрона.

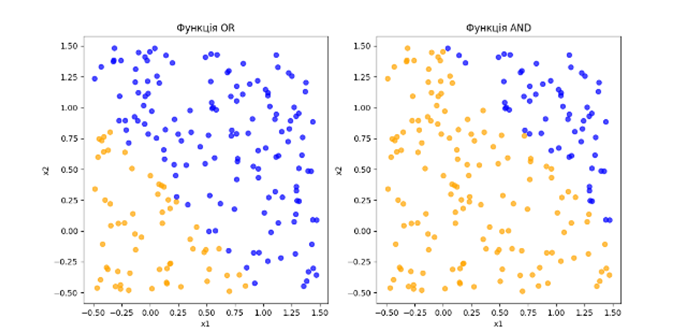
**Завдання 1.** Реалізувати обчислювальний алгоритм для or(x1, x2) і and(x1, x2) в програмному середовищі (С++, Python, та ін.). Для реалізації обчислювальних алгоритмів рекомендується використання онлайн середовищ тестування (наприклад repl.it, trinket, і.т.д.).

**Лістинг коду**

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)  
  
# 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  
y\_random = np.random.rand(num\_points) \* 2 - 0.5 # Генерація значень від -0.5 до 1.5  
  
# Класифікація точок за допомогою персептронів OR та AND  
or\_results = np.array([or\_perceptron(x, y) for x, y in zip(x\_random, y\_random)])  
and\_results = np.array([and\_perceptron(x, y) for x, y in zip(x\_random, y\_random)])  
  
# Встановлення кольорів: синій для 1, оранжевий для 0  
colors\_or = ['blue' if result == 1 else 'orange' for result in or\_results]  
colors\_and = ['blue' if result == 1 else 'orange' for result in and\_results]  
  
# Створення графіків  
fig, ax = plt.subplots(1, 2, figsize=(12, 6))

# Графік для OR  
ax[0].scatter(x\_random, y\_random, c=colors\_or, alpha=0.7)  
ax[0].set\_title('Функція OR')  
ax[0].set\_xlabel('x1')  
ax[0].set\_ylabel('x2')  
  
# Графік для AND  
ax[1].scatter(x\_random, y\_random, c=colors\_and, alpha=0.7)  
ax[1].set\_title('Функція AND')  
ax[1].set\_xlabel('x1')  
ax[1].set\_ylabel('x2')  
  
plt.show()

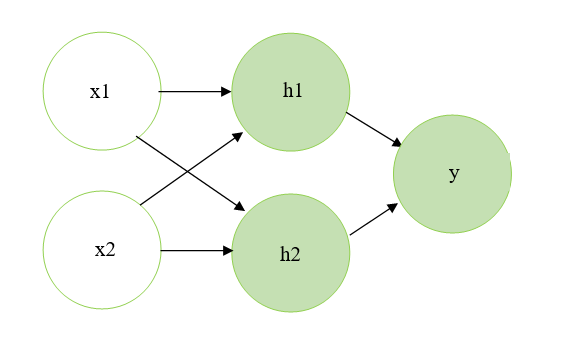
**Результат виконання коду**



Код генерує випадкові числа в діапазоні від 0 до 1, які масштабуються до інтервалу від -0.5 до 1.5. Початкове значення генератора фіксоване, що забезпечує відтворюваність даних. Аналізується персептрон для функцій OR та AND: задаються ваги і поріг, обчислюється зважена сума вхідних даних із врахуванням порогу, а результат визначається через функцію активації. Усі результати візуалізуються.

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

XOR(x1,x2) = OR(AND(x1,NOT(x2)),(AND(NOT(x1),x2))



Персептрон функції XOR.

Функція XOR не може бути розділена на два класи однією прямою, як у випадку з функціями AND чи OR, оскільки вона не є лінійно роздільною. Проте двошарова нейронна мережа здатна побудувати дві розділові лінії, які виділяють області для істинних і хибних значень XOR.

Розглянемо першу лінію для h1:

h1 = σ((-1) \* x1 + x2 + 1)

Лінійне рівняння для цієї межі можна виразити як:

−x1 + x2 + 1 = 0 або x2 = x1 − 1.

Визначаємо другу пряму для h2:

h2 = σ(x1 + (−1) \* x2 + 1)

Лінійне рівняння цієї межі можна записати:

x1 − x2 + 1 = 0 або x2 = x1 + 1.

**Висновок:** в ході виконання лабораторної роботи, дослідив математичну модель нейрона.