

Лабораторна №1

Нейронна реалізація логічних функцій AND, OR, XOR

Мета: Дослідити математичну модель нейрона.

Хід роботи

Завдання №1:

Реалізувати обчислювальний алгоритм для функції $\text{хор}(x_1, x_2)$ через функції $\text{or}(x_1, x_2)$ і $\text{and}(x_1, x_2)$ в програмному середовищі (C++, Python, та ін.).

Для реалізації обчислювальних алгоритмів рекомендується використання онлайн середовищ тестування (наприклад repl.it, trinket, і.т.д.).

Завдання №2:

Зобразити двохслойний персепtron для функції $\text{хор}(x_1, x_2)$ та скласти відповідне рівняння розділяючої прямої, використовуючи теоретичний матеріал даної лабораторної роботи.

Захист лабораторної роботи передбачає виконання практичних завдань поставлених в роботі, та виконання завдань теоретичного характеру.

1. Програмна реалізація

Було написано скрипт мовою Python у середовищі VS Code. Створено клас `Neuron`, який моделює роботу штучного нейрона за формулою

$$f(x) = \begin{cases} 1, & \text{якщо } \sum w_i x_i + bias > 0 \\ 0, & \text{інакше} \end{cases}$$

Для реалізації функції XOR використано архітектуру з трьох нейронів:

1. OR — визначає наявність хоча б однієї одиниці.
2. AND — визначає наявність двох одиниць одночасно.
3. XOR — віднімає результат AND від результату OR.

Результати тестування програми: Таблиця істинності підтверджує правильність роботи (0, 1, 1, 0).

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Пр.1		
Розроб.	Чайковський А.В					Lіт.	Арк.
Перевір.	Маєвський О					1	7
Реценз.							
Н. Контр.							
Зав.каф.							
<i>Звіт з лабораторної роботи №1</i>					<i>ФІКТ, гр. ІПЗ-22-3</i>		

Код

```

class Neuron:
    def __init__(self, w1, w2, bias):
        self.w1 = w1
        self.w2 = w2
        self.bias = bias

    def feed_forward(self, x1, x2):
        total = (x1 * self.w1) + (x2 * self.w2) + self.bias
        return 1 if total > 0 else 0

neuron_OR = Neuron(w1=1, w2=1, bias=-0.5)
neuron_AND = Neuron(w1=1, w2=1, bias=-1.5)
neuron_XOR = Neuron(w1=1, w2=-1, bias=-0.5)

inputs = [
    [0, 0],
    [0, 1],
    [1, 0],
    [1, 1]
]

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

for x1, x2 in inputs:
    res_or = neuron_OR.feed_forward(x1, x2)
    res_and = neuron_AND.feed_forward(x1, x2)
    res_final = neuron_XOR.feed_forward(res_or, res_and)

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

print("-" * 45)

```

PS C:\Users\chaik\source\repos\ConsoleApp> & C:/Users/chaik/AppData/Local/Programs/Python/Python313/python.exe c:/Users/chaik/source/repos/ConsoleApp/UML/lab01.py

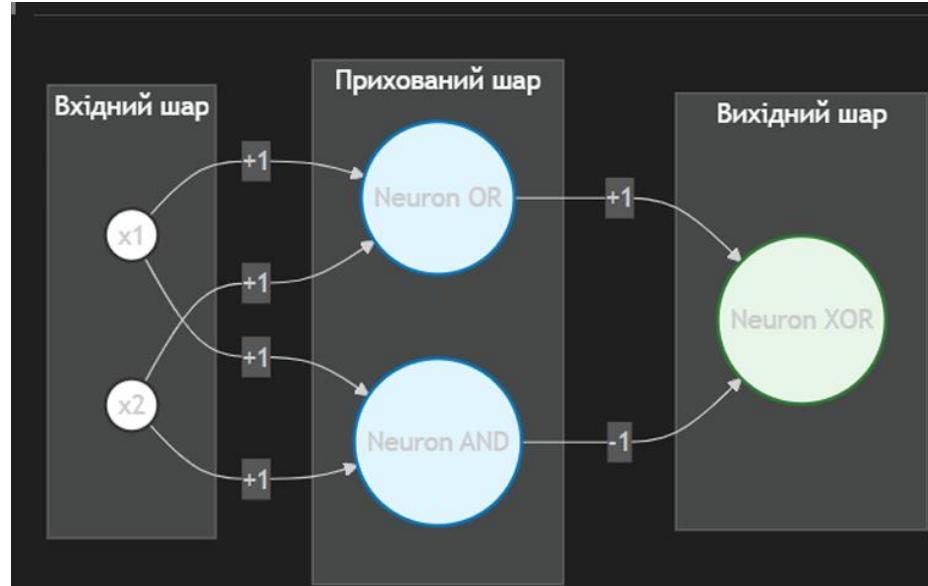
Вхід x1	Вхід x2	OR	AND	XOR (Фінал)
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

PS C:\Users\chaik\source\repos\ConsoleApp> █

2. Двошаровий персепtron Графічно змодельовано структуру мережі.

- Перший шар складається з нейронів OR та AND.
 - Другий шар приймає сигнали від першого шару.

		Чайковський А.В				Арк.
		Маєвський О				
Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Пр.1	2



Рівняння розділяючої прямої: Робота мережі описується системою нерівностей. Фінальне рішення приймається за формулою:

$$Y_{XOR} = H(1 \cdot Y_{OR} - 1 \cdot Y_{AND} - 0.5)$$

Де H — порогова функція Гевісайда. Це дозволяє розділити лінійно нерозрідільні класи, використовуючи дві прямі замість однієї.

Висновок: На лабораторній роботі я реалізував найпростішу нейронну мережу на Python. Я з'ясував, що один нейрон не може вирішити задачу XOR, тому використав дворівневу архітектуру. Отримані результати повністю відповідають таблиці істинності функції XOR

Github [AntonChaikovskyi/AI-systems](https://github.com/AntonChaikovskyi/AI-systems)

Чайковський А.В	Маєвський О	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Пр.1	Арк.
Змн.	Арк.	№ докум.	Підпис
		Дата	3