

Лабораторна робота № 1

Нейронна реалізація логічних функцій AND, OR, XOR

Мета роботи: дослідити математичну модель нейрона.

Хід роботи

Завдання 1.1. 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, [trinket](https://trinket.io), і.т.д.).

Лістинг LR_1_task1

```
def or_func(x1, x2):  
    w1, w2, w0 = 1, 1, -0.5  
    return 1 if (w1*x1 + w2*x2 + w0) >= 0 else 0  
  
def and_func(x1, x2):  
    w1, w2, w0 = 1, 1, -1.5  
    return 1 if (w1*x1 + w2*x2 + w0) >= 0 else 0  
  
def xor_func(x1, x2):  
    y1 = or_func(x1, x2)  
    y2 = and_func(x1, x2)  
    return 1 if (y1 - y2) >= 0.5 else 0  
  
print("x1 | x2 | OR | AND | XOR")  
print("-" * 30)  
for x1 in [0, 1]:  
    for x2 in [0, 1]:  
        or_result = or_func(x1, x2)  
        and_result = and_func(x1, x2)  
        xor_result = xor_func(x1, x2)  
        print(f" {x1} | {x2} | {or_result} | {and_result} | {xor_result}")
```

Рис.1.1.Результат виконання завдання

					ДУ «Житомирська політехніка».25.121.22.000–Пр1		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Свиcтанюк Н.О.					
Перевір.		Масєвський О.В.					
Керівник							
Н. контр.							
Зав. каф.							
					Звіт з лабораторної роботи		
					Лім.	Арк.	Аркушів
						1	3
					ФІКТ Гр. ІПЗ-22-3		

x1	x2	or	and	xor
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

У результаті виконання завдання було успішно реалізовано обчислювальний алгоритм для функції XOR через комбінацію базових логічних функцій OR і AND у середовищі Python. Реалізація підтвердила, що функція XOR не може бути представлена одним нейроном через свою нелінійну природу, але успішно реалізується через двошарову архітектуру. Результати обчислень повністю відповідають таблиці істинності XOR: функція повертає 1 тільки тоді, коли входи різні (01 або 10), і 0 коли входи однакові (00 або 11). Програмна реалізація з вагами $W_1=1$, $W_2=1$, $W_0=-0.5$ для OR та $W_0=-1.5$ для AND забезпечила коректну роботу

Завдання 1.2. Зобразити двошарний персептрон для функції $\text{xor}(x_1, x_2)$ та скласти відповідне рівняння розділюючої прямої, використовуючи теоретичний матеріал даної лабораторної роботи

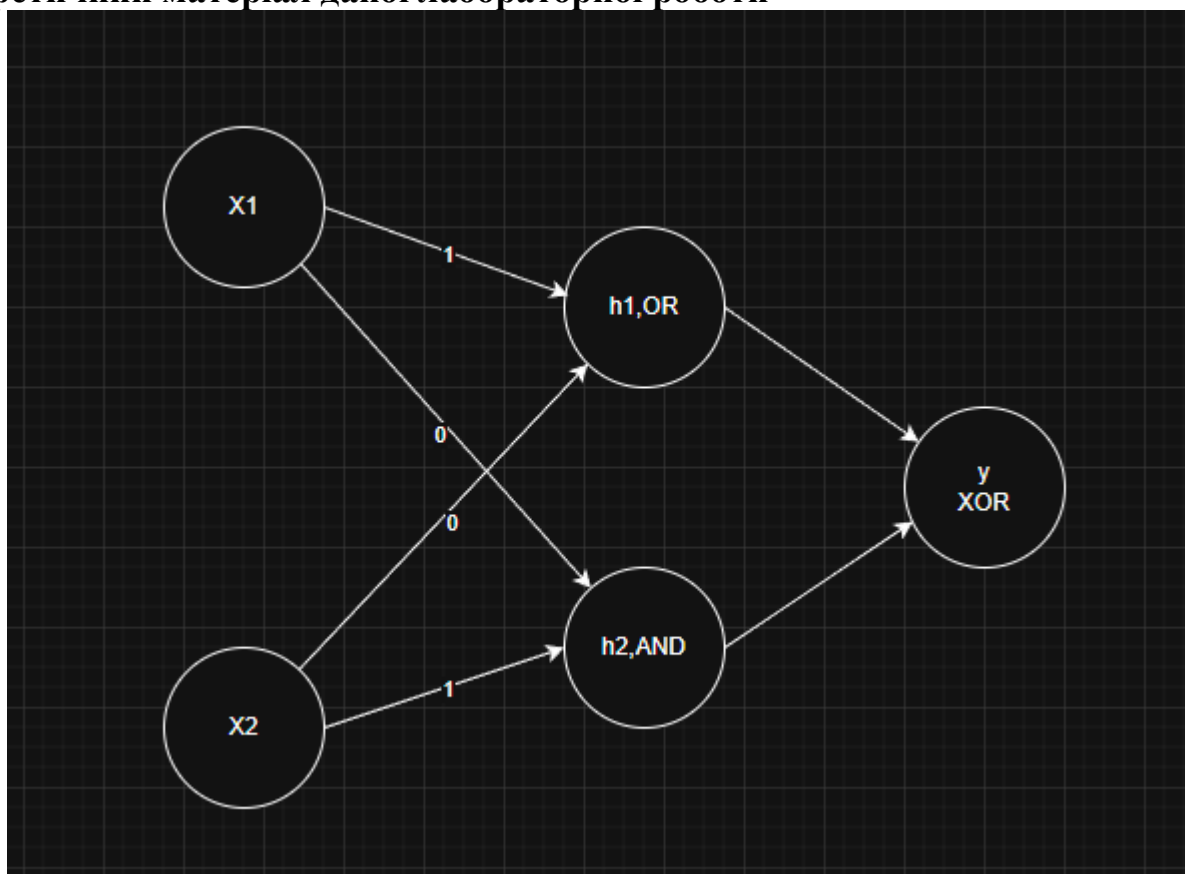


Рис.1.2.Результат виконання завдання

Перший шар (приховані нейрони) складається з двох елементів:

OR: $y_1 = f(x_1 + x_2 - 0.5)$

AND: $y_2 = f(x_1 + x_2 - 1.5)$

Другий шар (вихідний нейрон) обчислює результат:

$y = f(y_1 - y_2 - 0.5)$

		Свистанюк Н.О.			ДУ «Житомирська політехніка».25.121.22.000 – Пр1	Арк.
		Масевський О.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

де $f(z)$ - функція активації з методички (ступінчаста).

Рівняння розділяючої прямої

Для вихідного нейрона маємо рівність:

$$y_1 - y_2 - 0.5 = 0 \Rightarrow y_1 - y_2 = 0.5$$

Інтерпретація:

Якщо $y_1 - y_2 > 0.5$, то $XOR = 1$.

Якщо $y_1 - y_2 < 0.5$, то $XOR = 0$.

Таким чином, ця лінія виконує роль межі, що поділяє класи у просторі координат (y_1, y_2) .

У другому завданні було побудовано двошаровий персептрон для функції XOR та виведено рівняння розділяючої прямої $y_1 - y_2 = 0.5$. Архітектура складається з першого шару (два нейрони OR і AND) та другого шару (один нейрон, що комбінує результати). Математичний аналіз показав, що розділяюча пряма у просторі координат (y_1, y_2) ефективно класифікує вхідні дані: область $y_1 - y_2 > 0.5$ відповідає $XOR = 1$, а $y_1 - y_2 < 0.5$ відповідає $XOR = 0$. Це наочно демонструє принцип роботи багатошарових нейронних мереж для вирішення нелінійно-роздільних задач.

Висновок: в ході виконання лабораторної роботи було досліджено математичну модель нейрона та вирішено класичну проблему XOR у нейронних мережах. Практично підтверджено, що проста функція XOR, яка є нелінійно роздільною, не може бути реалізована одношаровим персептроном, але успішно моделюється двошаровою архітектурою. Робота продемонструвала важливість багатошарових структур для вирішення складних задач класифікації. Отримані результати підтверджують теоретичні положення про обмеження лінійних класифікаторів та необхідність використання прихованих шарів для розширення обчислювальних можливостей нейронних мереж. Практичні навички програмування нейронних алгоритмів на Python дозволяють застосовувати отримані знання для вирішення більш складних задач машинного навчання.

Репозиторій: <https://github.com/Svistaniuk/AIS>

		Сви́станюк Н.О.			ДУ «Житомирська політехніка».25.121.22.000 – Лр1	Арк.
		Масе́вський О.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		