

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Кафедра інтелектуальних та інформаційних систем

Лабораторна робота № 1
з дисципліни
“Нейромережні технології та їх застосування”

Виконав студент
групи КН-31
Пашковський Павло Володимирович

Київ-2021

Короткі теоретичні відомості

Штучні нейрони, які також називаються нейронними клітинами, вузлами, модулями, моделюють структуру й функції біологічних нейронів. Архітектура й особливості штучних нейронних мереж, утворених нейронами, залежать від конкретних завдань, які мають бути вирішені з їхньою допомогою.

Вхідними сигналами штучного нейрона $x_i (i = \overline{1, N})$ є вихідні сигнали інших нейронів, кожний з яких узятий зі своєю вагою $w_i (i = \overline{1, N})$ аналогічною синаптичній силі.

Вхідний оператор $f_{\text{вх}}$ перетворює зважені входи й подає їх на оператор активації f_a . Вихідний сигнал нейрона y являє собою перетворений вихідним оператором $f_{\text{вих}}$ вихідний сигнал оператора активації. Таким чином, нелінійний оператор перетворення вектора вхідних сигналів x у вихідний сигнал y може бути записаний у такий спосіб:

$$y = f_{\text{вих}}(f_a(f_{\text{вх}}(x, w)))$$

Контрольні питання

1. Що таке штучний нейрон?

Штучні нейрони моделюють структуру й функції біологічних нейронів.

2. Дайте означення штучної нейронної мережі.

Штучні нейронні мережі – обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок тварин. Такі системи навчаються задач (поступально покращують свою продуктивність на них), розглядаючи приклади, загалом без спеціального програмування під задачу.

3. Як функціонує нейрон? Структура штучного нейрона.

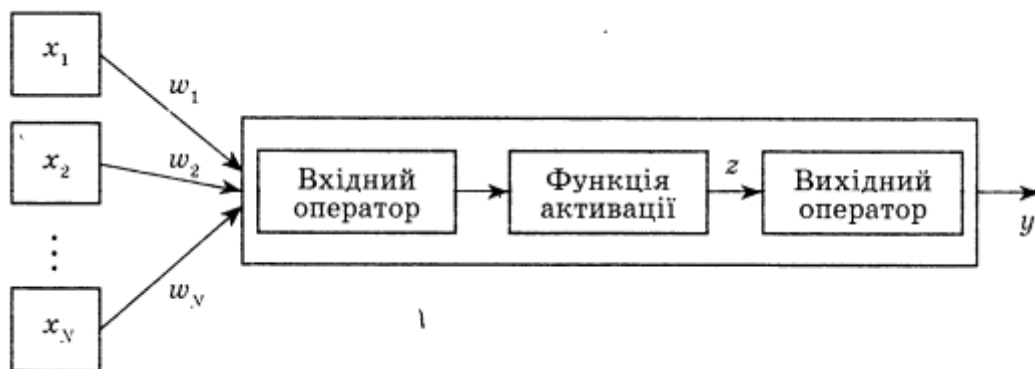


Рис. 1.1. Структура штучного нейрона.

Вхідними сигналами штучного нейрона $x_i (i = \overline{1, N})$ є вихідні сигнали інших нейронів, кожний з яких узятий зі своєю вагою $w_i (i = \overline{1, N})$ аналогічною синаптичній силі.

Вхідний оператор $f_{\text{вх}}$ перетворює зважені входи й подає їх на оператор активації f_a . Вихідний сигнал нейрона y являє собою перетворений вихідним оператором $f_{\text{вих}}$ вихідний сигнал оператора активації. Таким чином, нелінійний оператор перетворення вектора

вхідних сигналів x у вихідний сигнал y може бути записаний у такий спосіб:

$$y = f_{\text{вих}}(f_a(f_{\text{вх}}(x, w)))$$

4. Наведіть приклади основних функцій активації.

— лінійна

$$f(z) = Kz, \quad K = \text{const};$$

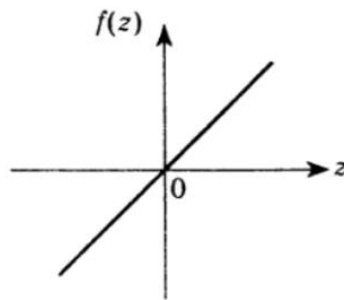


Рис. 1.2. Лінійна функція.

— лінійна біполярна з насиченням

$$f(z) = \begin{cases} 1 & \text{при } z > \alpha_2, \\ Kz & \text{при } -\alpha_1 \leq z \leq \alpha_2, \\ -1 & \text{при } z < -\alpha_1; \end{cases}$$

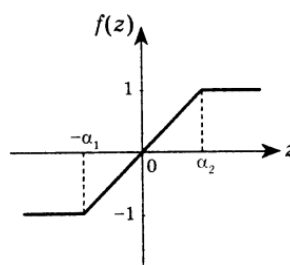


Рис 1.3. лінійна біполярна функція з насиченням.

—лінійна уніполярна з насиченням

$$f(z) = \begin{cases} 1 & \text{при } z \geq \frac{1}{2\alpha}, \\ \alpha z + 0,5 & \text{при } |z| < \frac{1}{2\alpha}, \\ 0 & \text{при } z \leq -\frac{1}{2\alpha}. \end{cases}$$

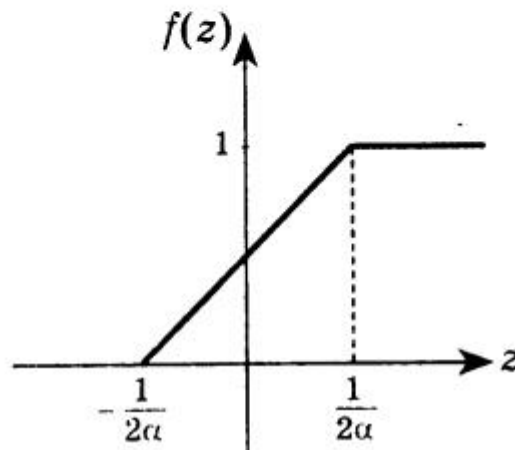


Рис. 1.4. лінійна уніполярна з насиченням

5. Наведіть приклади топологій штучних нейронних мереж.

ШНМ без зворотних зв'язків (прямого поширення, Feed forward):

- першого порядку;
- другого порядку (з «shortcut connections»);

ШНМ зі зворотними зв'язками (зворотного поширення, рекурентні, Feedback):

- з прямими зворотними зв'язками (direct feedback);
- з непрямыми зворотними зв'язками (indirect feedback);
- з латеральними зв'язками (lateral feedback); – повнозв'язні.

6. Що таке задача класифікації?

Задача класифікації представляє собою задачу віднесення зразка до одного з декількох попарно непересічних множин.

7. Рівні складності класифікації.

Перший (найпростіший) - коли класи можна розділити прямими лініями (або гіперплощинами, якщо простір входів має розмірність більше двох) - так звана лінійна роздільність.

В другому випадку класи неможливо розділити лініями (площинами), але їх, можливо, відділити за допомогою більш складного розподілу - нелінійна роздільність.

В третьому випадку класи перетинаються, і можна говорити тільки про роздільність ймовірності.

8. Яку топологію ШНМ частіше застосовують при вирішенні задачі класифікації?

ШНМ без зворотних зв'язків другого порядку

9. Особливості застосування одношарових ШНМ.

Одношарові нейронні мережі здатні розв'язувати тільки лінійно роздільні задачі.

10. Особливості навчання з учителем та без нього.

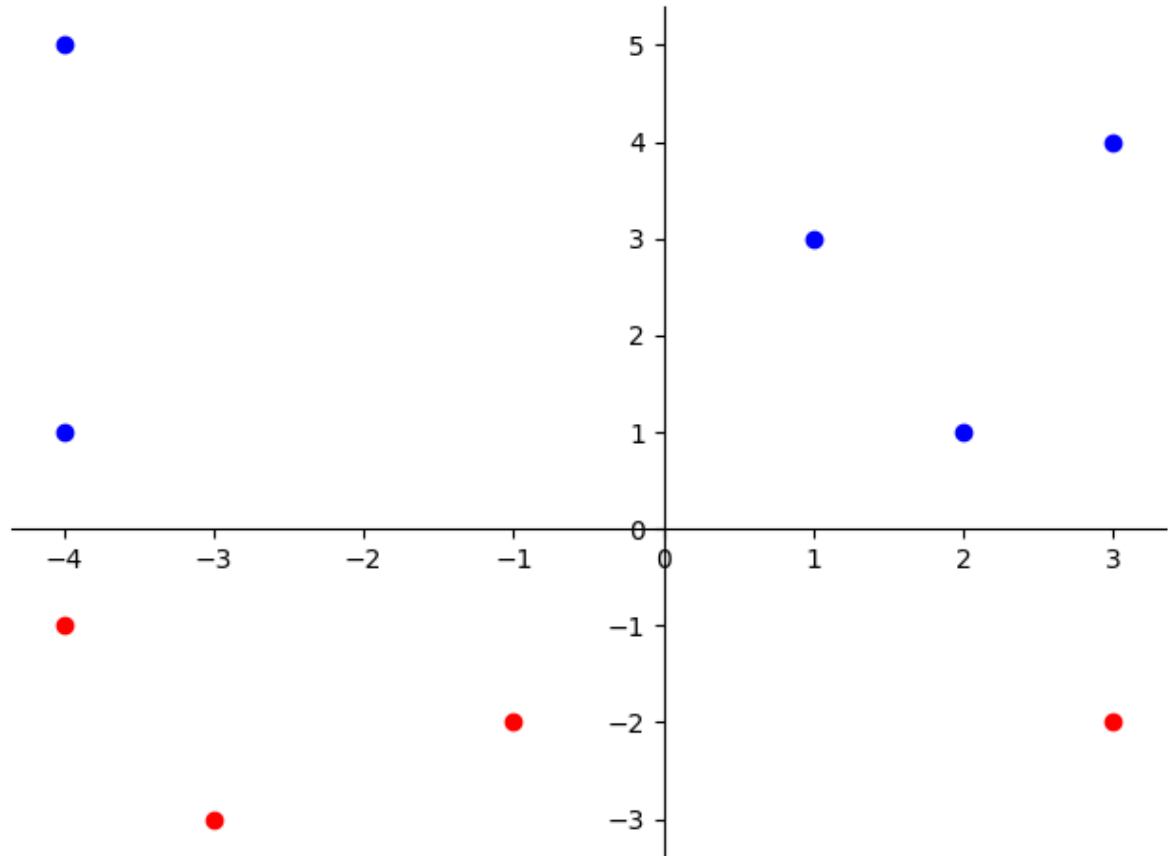
Навчання з учителем припускає, що є «учитель», що задає пари, які навчають — для кожного вхідного вектора, що навчає, необхідний вихід мережі. Для кожного вхідного вектора, що навчає, обчислюється вихід мережі, порівнюється з відповідно необхідним, визначається похибка виходу, на основі якої й коректуються ваги. Пари, що навчають, подаються мережі послідовно й ваги уточнюються доти, поки похибка за такими парами не досягне необхідного рівня.

Без учителя – мережі подаються тільки вектори вхідних сигналів, і мережа сама, використовуючи деякий алгоритм навчання, підстроювала б ваги так, щоб при поданні їй досить близьких вхідних векторів вихідні сигнали були б однаковими. У цьому випадку в процесі навчання виділяються статистичні властивості множини вхідних векторів, що навчають, і відбувається об'єднання близьких (подібних) векторів у класи

Індивідуальне завдання:

Варіант 3

Значення $(x_1; x_2)$ у 1 і 2 чверті (наприклад, $(2; 1)$ $(3; -2)$) та у 3 і 4 чверті координатної площини (наприклад, $(-4; 5)$ $(-4; -1)$).



Математична модель штучного нейрона:

$$y = \sum_{i=1}^2 w_i x_i$$

$$y = 0 \cdot x_1 + x_2$$

$$z = \frac{1}{1 + e^{-(0 \cdot x_1 + x_2)}}$$

Обмеження:

$z > 0.5 \Rightarrow$ I та II чверті;

$z < 0.5 \Rightarrow$ III та IV чверті;

$x_1, x_2 = 0 \Rightarrow$ нейрон не класифікує.

Висновок: була проведена робота з вивчення основ функціонування штучного нейрона і його застосування до лінійно роздільної найпростішої класифікації.

Через свою здатність відтворювати та моделювати нелінійні процеси, ШНМ знайшли застосування в широкому діапазоні дисциплін.

Код програми:

```
import matplotlib.pyplot as plt
import numpy as np
x = np.array([[2, 1], [3, -2], [-4, 5], [-4, -1], [-1, -2], [3, 4], [-4, 1], [-3, -3], [1, 3], [0, 0]])
for x1, x2 in x:
    z = (1 / (1 + np.exp(x2)))
    if z > 0.5:
        print("1 and 2 = (", x1, ":", x2, ")")
        plt.scatter(x1, x2, c='r')
    elif z < 0.5:
        print("3 and 4 = (", x1, ":", x2, ")")
        plt.scatter(x1, x2, c='b')

ax = plt.gca()
ax.spines['left'].set_position('zero')
ax.spines['bottom'].set_position('zero')
ax.spines['top'].set_visible(False)
ax.spines['right'].set_visible(False)
plt.show()
```