

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Кафедра інтелектуальних та інформаційних систем

Лабораторна робота № 3
з дисципліни
“Нейромережні технології та їх застосування”

Виконав студент
групи КН-31
Пашковський Павло Володимирович

Київ-2021

Контрольні питання

1. У чому полягає відмінність елементарного перцептрона від одношарового перцептрона?

Одношаровий - перцептрон, кожен S-елемент якого однозначно відповідає одному A-елементу, S-A зв'язки завжди рівні 1, а поріг будь-якого A-елемента дорівнює 1.

Елементарним перцептроном називається спрощений перцептрон, у якого є лише прості R і A елементи.

2. Який алгоритм використовується для навчання одношарового перцептрона?

Навчання за дельта правилом.

Крок 1. Ініціалізація ваг перцептрона в діапазоні від $[-0.1; +0.1]$

Крок 2. На вхід перцептрона подається вхідний вектор X. Кожен нейрон виконує зважене підсумування вхідних сигналів і формує вихідний сигнал Y.

Крок 3. Якщо реакція нейронної мережі збігається з еталонним значенням, то ваговий коефіцієнт не змінюється.

Крок 4. Якщо вхідна реакція збігається з еталонною, то проводиться модифікація вагових коефіцієнтів.

Крок 5. Алгоритм триває до тих пір, поки не стане $y = t$ для всіх вхідних образів, або не перестануть змінюватися вагові коефіцієнти.

3. Як оцінюється якість розпізнавання одношарового перцептрона?

Відношення кількості правильних відповідей, які дала мережа, до всіх відповідей мережі.

Індивідуальне завдання:

Виконати програмну реалізацію для моделювати роботи одношарової мережі, з 9 входами і 4 нейронами, яка здатна розпізнавати 8 об'єктів, представлених на рис.1

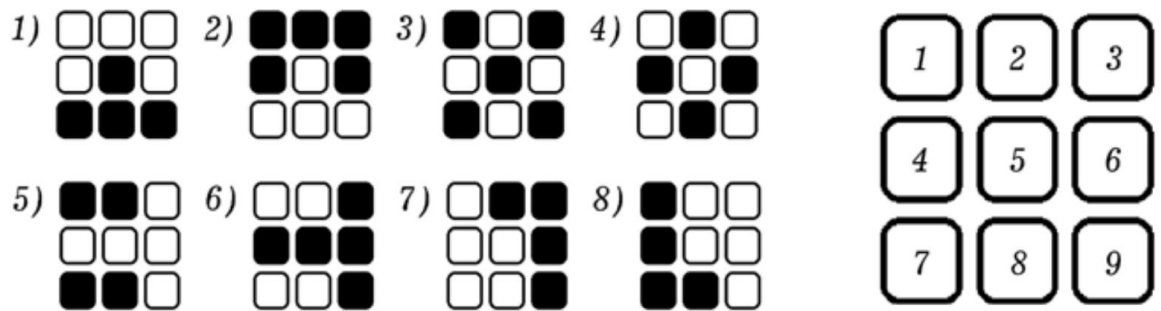


Рис. 1. Об'єкти для розпізнавання.

Вхідні об'єкти закодируємо так:

o [1] = {000010111},

o [2] = {111101000},

....

o [8] = {100100110}.

Нехай їм відповідають такі сигнали на виходах нейронів:

y[1,j]={0000},

y[2,j]={1111},

y[3,j]={1001},

y[4,j]={0110},

y[5,j]={1010},

y[6,j]={0101},

y[7,j]={1100},

y[8,j]={0011}

```

epoch: 6
number 1      [0. 0. 0. 0.]    [0. 0. 0. 0.]
number 2      [1. 1. 1. 1.]    [1. 1. 1. 1.]
number 3      [1. 0. 0. 1.]    [1. 0. 0. 1.]
number 4      [0. 1. 1. 0.]    [0. 1. 1. 0.]
number 5      [1. 0. 1. 0.]    [1. 0. 1. 0.]
number 6      [0. 1. 0. 1.]    [0. 1. 0. 1.]
number 7      [1. 1. 0. 0.]    [1. 1. 0. 0.]
number 8      [0. 0. 1. 1.]    [0. 0. 1. 1.]

```

Рис. 2. Результати роботи програми.

Висновок:

Було створено програму для розпізнавання певного образу, що подається на вході, за допомогою одношарового перцептрона і навчання за допомогою дельта правила.

Дану програму можна використовувати для розпізнавання чисел, образів.

В майбутньому програму можна вдосконалити та використовувати для розпізнавання будь-яких лінійно-роздільних об'єктів.

Код програми:

```

import numpy as np

def training(X,T,Weights,LR):
    sigma = np.zeros(len(T[0]))
    Y = np.zeros(len(T[0]))
    Beta = np.zeros(len(T[0]))
    fl = 1
    epoch = 0
    while(fl):
        epoch+=1
        for k, xk in enumerate(X):
            for i, wi in enumerate(Weights):
                d = 0
                for j in range(len(wi)):
                    d+= wi[j]*xk[j]
                sigma[i]= d
                if sigma[i] >=0:
                    Y[i]=1
                else:
                    Y[i]=0
                Beta[i] = T[k][i]-Y[i]
            fl1 = 1
        for i in range(len(Y)):

```

```

        if T[k][i] != Y[i]:
            fl1 = 0
    if(fl1):
        fl+=1
    else:
        for i,wi in enumerate(Weights):
            for j in range(len(Weights[i])):
                Weights[i][j] += LR*xk[j]*Beta[i]
    if fl == 9 :
        fl= 0
    else:
        fl=1
    print("epoch:",epoch)

```

```

def test(X,T,Weights):
    sigma = np.zeros(len(T[0]))
    Y = np.zeros(len(T[0]))
    for k, xk in enumerate(X):
        for i, wi in enumerate(Weights):
            d = 0
            for j in range(len(wi)):
                d += wi[j] * xk[j]
            sigma[i] = d
            if sigma[i] >= 0:
                Y[i] = 1
            else:
                Y[i] = 0
    print("number ", k+1, "\t", Y, "\t", T[k])

```

```

def inputx(Weights):
    print("\ninput x:")
    xtest=input()
    xi=xtest.split()
    xi=list(xi)
    sigma = np.zeros(len(T[0]))
    Y = np.zeros(len(T[0]))
    for i, wi in enumerate(Weights):
        d = 0
        for j in range(len(wi)):
            d += wi[j] * int(xi[j])
        sigma[i] = d
        if sigma[i] >= 0:
            Y[i] = 1
        else:
            Y[i] = 0
    print(Y)

```

```

X = np.genfromtxt('X.txt',delimiter=' ',dtype=np.float)
T = np.genfromtxt('T.txt', delimiter=' ', dtype=np.float)
Weights = np.random.uniform(-0.1, 0.1, (4,9))
LR = 0.01
training(X,T,Weights,LR)
test(X,T,Weights)

```

```

while(1):
    inputx(Weights)

```