Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №5
По дисциплине: «МиАПР»
Тема: «Нелинейные ИНС в задачах распознавания образов»

Выполнил:
Студент 2 курса
Группы ПО-7(2)
Фурсевич Д.С.
Проверил:
Крощенко А.А.

2021

**Цель работы:** изучить обучение и функционирование нелинейной ИНС при решении задач распознавания образов.

## Ход работы

Вариант 11

**Задание:**

Написать на любом ЯВУ программу моделирования нелинейной ИНС для распознавания образов (ррекомендуется использовать сигмоидную функцию). Количество НЭ в скрытом слое взять согласно варианту работы №3 (можно варьировать, если сеть не обучается или некорректно функционирует).

Провести исследование полученной модели. При этом на вход сети необходимо подавать искаженные образы, в которых инвертированы некоторые биты. Критерий эффективности процесса распознавания - максимальное кодовое расстояние (количество искаженных битов) между исходным и поданным образом.

Таблица 1. Варианты заданий.

| Вариант | Вектор 1 | Вектор 2 | Вектор 3 |
|---|---|---|---|
| 1 | 1 | 6 | 8 |
| 2 | 2 | 1 | 8 |
| 3 | 3 | 2 | 8 |
| 4 | 4 | 3 | 8 |
| 5 | 5 | 4 | 8 |
| 6 | 6 | 5 | 8 |
| 7 | 7 | 6 | 8 |
| 8 | 1 | 3 | 8 |
| 9 | 2 | 4 | 8 |
| 10 | 3 | 5 | 8 |
| 11 | 4 | 6 | 8 |

Таблица 2. Набор векторов.

| № | Данные вектора | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 3 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |

**Код программы:**

```
import random
import math
import matplotlib.pyplot as plt

step = 0.5
mistake_min = 1e-4
num_in = 8  # количество входов инс
num_hid = 3  # количество нэ в скрытом слое
num_out = 1
w_ij = [[random.uniform(-0.1, 0.1) for i in range(num_hid)] for j in range(num_in - 1)]
w_jk = [[random.uniform(-0.1, 0.1) for _ in range(num_hid)] for j in range(num_out)]
thresholdValue_j = [random.uniform(-0.5, 0.5) for _ in range(num_hid)]
thresholdValue_k = [random.uniform(-0.5, 0.5) for _ in range(num_out)]
```

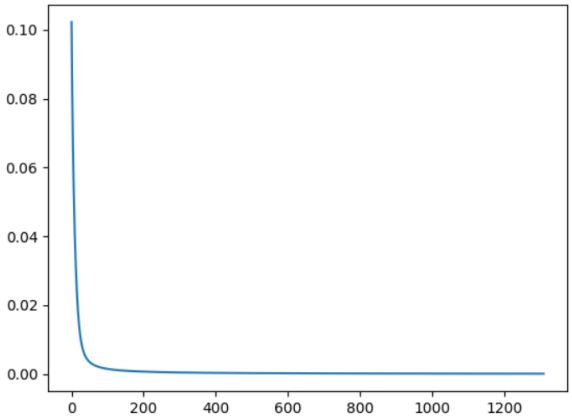
```
vector0 = [1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0]
vector1 = [1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
vector2 = [1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1]
vector3 = [0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0]
vector4 = [0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0]
vector5 = [1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1]
Vector = [vector0, vector1, vector2, vector3, vector4, vector5]


def sigmoidActivationFunction(s):
    return (1 / (1 + math.exp(-1 * s)))


def hiddenLayer_Sj(y):
    S_j = []
    for i in range(num_hid):
        value = 0
        for j in range(num_in - 1):
            value += y[j] * w_ij[j][i]
        value -= thresholdValue_j[i]
        S_j.append(sigmoidActivationFunction(value))
    return S_j


def outputLayer_Sk(y_j):
    S_k = []
    for j in range(num_out):
        value = 0
        for i in range(num_hid):
            value += y_j[i] * w_jk[j][i]
        value -= thresholdValue_k[j]
        S_k.append(sigmoidActivationFunction(value))
    return S_k


def change_w_jk(y_j, y_k, mistake):
    global thresholdValue k
    for j in range(num_out):
        for i in range(num_hid):
            w_jk[j][i] -= step * mistake[j] * y_k[j] * (1 - y_k[j]) * y_j[i]
        thresholdValue_k[j] += mistake[j] * step * y_k[j] * (1 - y_k[j])


def change_w_ij(y_j, mistake_hid, y):
    for j in range(num_hid):
        for i in range(num_in - 1):
            w_ij[i][j] -= step * mistake_hid[j] * y[i] * y_j[j] * (1 - y_j[j])
        thresholdValue_j[j] += step * mistake_hid[j] * y_j[j] * (1 - y_j[j])

if __name__ == '__main__':
    x_points, y_points = [], []
    mistakes = [0] * num_out
    etalons = [0] * num_out
    mistake_hid = [0] * num_hid
    iter = 1
    n = 0
    while True:
        error = 0
        for N in range(num out):
            y, y_j, y_k = [], [], []
            etalons[N] = 1
            for i in range(iter):
                y = Vector[N]
                y_j = hiddenLayer_Sj(y)
                y_k = outputLayer_Sk(y_j)
                for i in range(num_out):
                    mistakes[i] = y_k[i] - etalons[i]
                for j in range(num_hid):
                    for k in range(num_out):
```

```
                mistake_hid[j] += mistakes[k] * y_k[k] * (1 - y_k[k]) * w_jk[k][j]
            change_w_jk(y_j, y_k, mistakes)
            change_w_ij(y_j, mistake_hid, y)
            error += mistakes[N] ** 2
    error /= 2
    y_points.append(error)
    x_points.append(n)

    if abs(error) < mistake_min:
        print("finaly: ", error)
        break
    print(n, ';', error)
    n += 1
plt.plot(x_points, y_points)
plt.show()

for i in range(6):
    input = Vector[i]
    print("Result of vector: ", i + 1)
    for j in range(20):
        print(input[j], end=' ')
    print("\nResult : ")
    hid_pred = hiddenLayer_Sj(input)
    Values = outputLayer_Sk(hid_pred)
    print(Values[0])
```

## Результат выполнения программы:

```
1308 ; 0.00010003141744634856
Result of vector: 1
11001100110011001100
Result : 0.9858666625496909
Result of vector: 2
11111111000000000000
Result : 0.9858650741542371
Result of vector: 3
11100011100011100011
Result : 0.9854529659872567
Result of vector: 4
01001101000010101000
Result : 0.9857997321501704
Result of vector: 5
00001111000011110000
Result : 0.9854509063944695
Result of vector: 6
11110000111100001111
Result : 0.9854666766881919
```

## График изменения ошибки в зависимости от итерации:



**Вывод:** изучила обучение и функционирование нелинейной ИНС при решении задач распознавания образов.