Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный университет”

Кафедра ИИТ

Лабораторная работа №5

По дисциплине: “МиАПР”

Тема: “ Нелинейные ИНС в задачах распознавания образов”

Вариант №4

Выполнил:

Студент 2-го курса

Группы ПО-7

Дмитрук М.А.

Проверил:

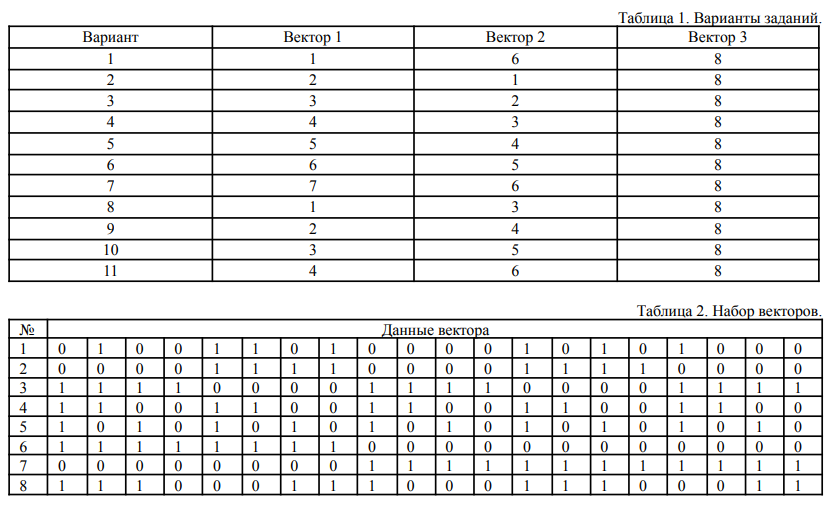
Крощенко А.А.

Брест, 2021

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

**Задание:** Написать на любом ЯВУ программу моделирования нелинейной ИНС для распознавания образов (рекомендуется 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.



**Код программы:**

import random

import math

import matplotlib.pyplot as Graph

def sigm\_Function(S):

return 1 / (1 + math.exp(-S))

def Sj\_Hidden(y):

Sj = []

for j in range(hidden\_neuron\_number):

value = 0

for i in range(input\_neuron\_number - 1):

value += y[i] \* Wij[i][j]

value -= Tj[j]

Sj.append(sigm\_Function(value))

return Sj

def Sk\_Output(Yj):

Sk = []

for j in range(output\_neuron\_number):

value = 0

for i in range(hidden\_neuron\_number):

value += Yj[i] \* Wjk[j][i]

value -= Tk[j]

Sk.append(sigm\_Function(value))

return Sk

def Wjk\_Change(Yj, Yk, error):

global Tk

for j in range(output\_neuron\_number):

for i in range(hidden\_neuron\_number):

Wjk[j][i] -= step \* error[j] \* Yk[j] \* (1 - Yk[j]) \* Yj[i]

Tk[j] += error[j] \* step \* Yk[j] \* (1 - Yk[j])

def Wij\_Change(Yj, Hidden\_error, y):

for j in range(hidden\_neuron\_number):

for i in range(input\_neuron\_number - 1):

Wij[i][j] -= step \* Hidden\_error[j] \* y[i] \* Yj[j] \* (1 - Yj[j])

Tj[j] += step \* Hidden\_error[j] \* Yj[j] \* (1 - Yj[j])

vector\_4 = [1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0]

vector\_3 = [1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1]

vector\_8 = [1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1]

all\_vectors = [vector\_4, vector\_3, vector\_8]

step = 0.2

min\_error = 1e-5

input\_neuron\_number = 10

hidden\_neuron\_number = 4

output\_neuron\_number = 1

Wij = [[random.uniform(-0.1, 0.1) for \_ in range(hidden\_neuron\_number)] for \_ in range(input\_neuron\_number - 1)]

Wjk = [[random.uniform(-0.1, 0.1) for \_ in range(hidden\_neuron\_number)] for \_ in range(output\_neuron\_number)]

Tj = [random.uniform(-0.5, 0.5) for \_ in range(hidden\_neuron\_number)]

Tk = [random.uniform(-0.5, 0.5) for \_ in range(output\_neuron\_number)]

def main():

arr\_Graph = ([], [])

errors = [0] \* output\_neuron\_number

reference = [0] \* output\_neuron\_number

Hidden\_error = [0] \* hidden\_neuron\_number

iteration = 1

generation = 0

error = 1

while error > min\_error:

error = 0

for N in range(output\_neuron\_number):

reference[N] = 1

for i in range(iteration):

y = all\_vectors[N]

Yj = Sj\_Hidden(y)

Yk = Sk\_Output(Yj)

for index in range(output\_neuron\_number):

errors[index] = Yk[index] - reference[index]

for j in range(hidden\_neuron\_number):

for k in range(output\_neuron\_number):

Hidden\_error[j] += errors[k] \* Yk[k] \* (1 - Yk[k]) \* Wjk[k][j]

Wjk\_Change(Yj, Yk, errors)

Wij\_Change(Yj, Hidden\_error, y)

error += errors[N] \*\* 2

error /= 2

arr\_Graph[0].append(generation)

arr\_Graph[1].append(error)

generation += 1

Graph.plot(\*arr\_Graph)

Graph.xlabel("generation")

Graph.ylabel("Error")

Graph.show()

for i in range(len(all\_vectors)):

input = all\_vectors[i]

print("Result vector :", i + 1, end=" : ")

for j in range(len(vector\_4)):

print(input[j], end='')

print("\nResult : ", end='')

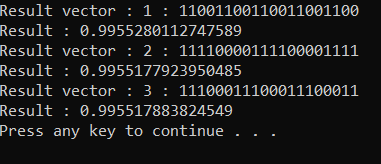
Hidden\_prev = Sj\_Hidden(input)

Values = Sk\_Output(Hidden\_prev)

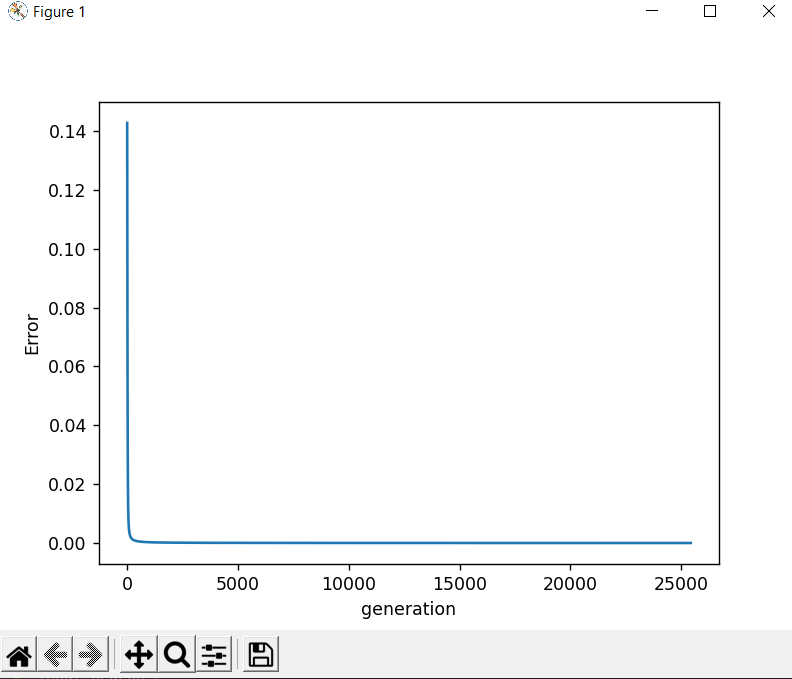
print(Values[0])

main()

**Результат выполнения программы:**



**График изменения ошибки от номера эпохи:**



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