Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №5

По дисциплине: «МиАПР»

Тема: «Изучить обучение и функционирование нелинейной ИНС при решении задач распознавания образов.»

Вариант №1

Выполнил:

Студент 2 курса

Группы ПО-7

Белев С.В.

Проверил:

Крощенко А.А.

Брест 2021

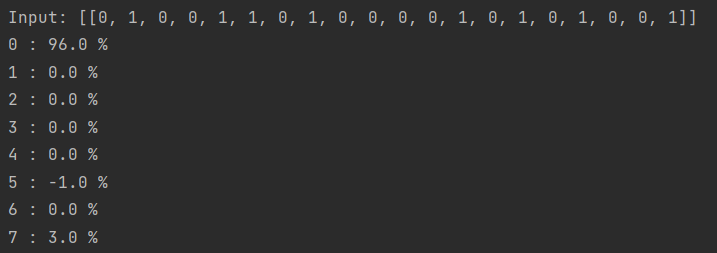
**Цель работы:** изучить обучение и функционирование нелинейной ИНС при решении задач распознавания образов.

**Задание:** Написать на любом ЯВУ программу моделирования нелинейной ИНС для распознавания образов. Рекомендуется использовать сигмоидную функцию, но это не является обязательным. Количество НЭ в скрытом слое взять согласно варианту работы №3. Его можно варьировать, если сеть не обучается или некорректно функционирует.

**Код программы:**

from random import random as r  
from math import exp as exp  
  
vectors = [  
 [0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0],  
 [0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0],  
 [1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1],  
 [1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0],  
 [1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0],  
 [1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],  
 [1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1]  
]  
vectors\_code = [  
 [1, 0, 0, 0, 0, 0, 0, 0],  
 [0, 1, 0, 0, 0, 0, 0, 0],  
 [0, 0, 1, 0, 0, 0, 0, 0],  
 [0, 0, 0, 1, 0, 0, 0, 0],  
 [0, 0, 0, 0, 1, 0, 0, 0],  
 [0, 0, 0, 0, 0, 1, 0, 0],  
 [0, 0, 0, 0, 0, 0, 1, 0],  
 [0, 0, 0, 0, 0, 0, 0, 1]  
]  
x\_list = [vectors[0], vectors[5], vectors[7]]  
e\_list = [vectors\_code[0], vectors\_code[5], vectors\_code[7]]  
Wki = [[r() for i in range(20)], [r() for i in range(20)]]  
Wij = [[r() for i in range(8)], [r() for i in range(8)]]  
Ti = [r() for i in range(2)]  
Tj = [r() for i in range(8)]  
ls = 0.01  
MSE = 0  
  
  
def sigm(x):  
 return 1 / (1 + exp(-x))  
  
  
def der\_sigm(x):  
 return x \* (1 - x)  
  
  
def H\_Calc(x\_list, Wki, Ti, ne, model):  
 S = 0  
 for i in range(20):  
 S += Wki[ne][i] \* x\_list[model][i]  
 return sigm(S - Ti[ne])  
  
  
def O\_Calc(x\_list, Wki, Wij, Ti, Tj, model):  
 S = []  
 for i in range(8):  
 S.append(H\_Calc(x\_list, Wki, Ti, 0, model) \* Wij[0][i] + H\_Calc(x\_list, Wki, Ti, 1, model) \* Wij[1][i] - Tj[i])  
 return S  
  
  
def Errs\_calc(e\_list, res):  
 Errs = [[], []]  
 Errsum = 0  
 for i in range(8):  
 Errs[0].append(res[i] - e\_list[i])  
 for i in range(2):  
 for k in range(8):  
 Errsum += Errs[0][k] \* Wij[i][k]  
 Errs[1].append(Errsum)  
 Errsum = 0  
 return Errs  
  
  
for Age in range(10000):  
 for model in range(3):  
 res = O\_Calc(x\_list, Wki, Wij, Ti, Tj, model)  
 h\_res = [H\_Calc(x\_list, Wki, Ti, 0, model), H\_Calc(x\_list, Wki, Ti, 1, model)]  
 Errs = Errs\_calc(e\_list[model], res)  
  
 for i in range(2):  
 for k in range(20):  
 Wki[i][k] = Wki[i][k] - ls \* Errs[1][i] \* der\_sigm(h\_res[i]) \* x\_list[model][k]  
 for i in range(2):  
 Ti[i] = Ti[i] + ls \* Errs[1][i] \* der\_sigm(h\_res[i])  
  
 for i in range(2):  
 for k in range(8):  
 Wij[i][k] = Wij[i][k] - ls \* Errs[0][k] \* h\_res[i]  
 for i in range(8):  
 Tj[i] = Tj[i] + ls \* Errs[0][i] \* res[i]  
  
 for i in range(8):  
 MSE += Errs[0][i] \*\* 2  
 MSE = 0  
inp\_vector = [[0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1]]  
print("Input:", inp\_vector)  
result = O\_Calc(inp\_vector, Wki, Wij, Ti, Tj, 0)  
for i in range(8):  
 print(i, ":", round(result[i], 2)\*100, "%")

**Результат работы программы:**



**Вывод:** На ввод я подал первый вектор с измененным последним битом, ответ виден в результате.