МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Брестский государственный технический университет»

Кафедра «Интеллектуальные информационные технологии»

Лабораторная работа №7

По дисциплине «Интеллектуальные методы обработки данных»

За 7 семестр

Тема: «Векторный квантователь»

Выполнила:  
студентка 4 курса  
группы АС-56  
Карпенко М.В.

Проверил:

Савицкий Ю.В.

Брест 2022

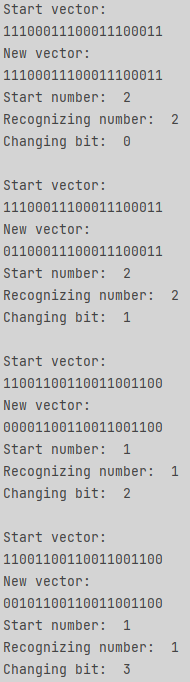
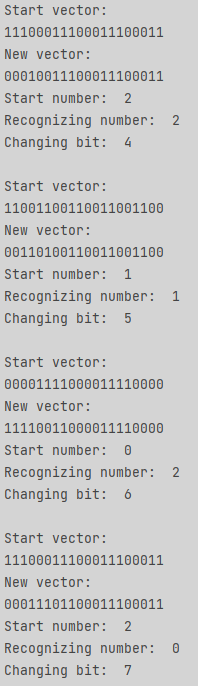
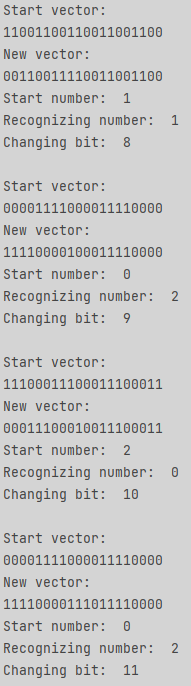
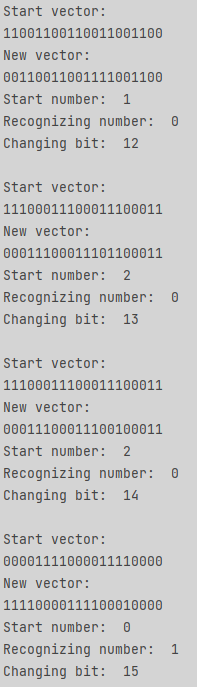
*Цель работы:* Изучить обучение и функционирование искусственной нейронной сети (ИНС) векторного квантователя при решении задач распознавания образов.

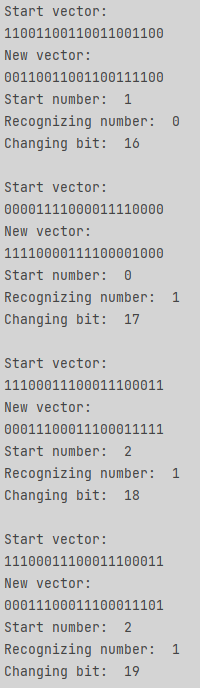
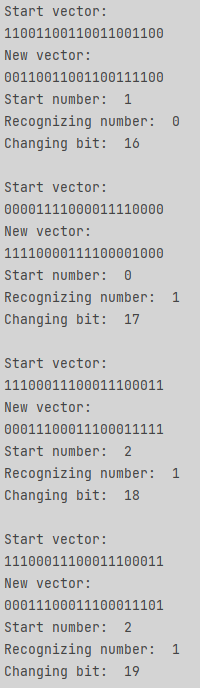
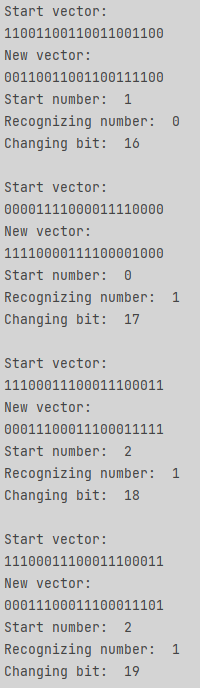
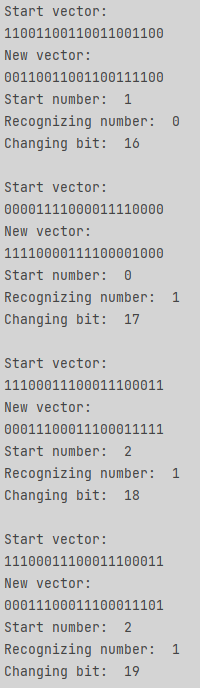
*Задание:*   
Написать на любом языке высокого уровня программу моделирования ИНС векторного квантователя для распознавания образов.

*Текст программы:*

import random  
import math  
  
  
class Helper:  
 input\_vector = []  
 all\_vector = [[], [], []]  
 w = [[], [], []]  
  
 def \_\_int\_\_(self, \_all\_vectors):  
 self.all\_vector = \_all\_vectors  
 for i in range(3):  
 self.w[i] = [0 for n in range(20)]  
 for j in range(20):  
 self.w[i][j] = random.randint(0, 1)  
  
 def set\_vector(self, vector\_number):  
 self.input\_vector = [0 for n in range(20)]  
  
 for i in range(len(self.all\_vector[vector\_number])):  
 self.input\_vector[i] = self.all\_vector[vector\_number][i]  
  
 def calculate\_vector(self):  
 output = [0 for n in range(3)]  
 for i in range(3):  
 for j in range(20):  
 output[i] += (self.input\_vector[j] - self.w[i][j]) \*\* 2  
 output[i] = math.sqrt(output[i])  
 return output  
  
 def find\_winer(self, output):  
 minimal = min(output)  
  
 for i in range(len(output)):  
 if output[i] == minimal:  
 return i  
 return 0  
  
 def recalculate\_winner(self, winner, real, iteration):  
  
 for i in range(20):  
 self.w[winner][i] += (1 if winner == real else -1) \* (float(self.input\_vector[i]) - float(self.w[winner][i])) / (  
 float(iteration + 1))  
  
 def study(self, total\_iterations):  
 for i in range(total\_iterations):  
 for j in range(3):  
 self.set\_vector(j)  
 output = self.calculate\_vector()  
 winner = self.find\_winer(output)  
 self.recalculate\_winner(winner, j, i)  
  
 def recognize(self, recognizing\_vector):  
 for i in range(len(self.input\_vector)):  
 self.input\_vector[i] = recognizing\_vector[i]  
 output = []  
 output = self.calculate\_vector()  
 return self.find\_winer(output)  
  
  
def main():  
 vector\_2 = [0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0]  
 vector\_4 = [1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0]  
 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\_2, vector\_4, vector\_8]  
  
 network = Helper()  
 network.\_\_int\_\_(all\_vectors)  
 network.study(400)  
 i = 2  
 for i in range(20):  
 count = 0  
 vector\_number = random.randint(0, 2)  
 temp\_vector = [0 for n in range(20)]  
  
 for c in range(20):  
 temp\_vector[c] = all\_vectors[vector\_number][c]  
  
 print("Start vector:")  
 for j in range(len(all\_vectors[vector\_number])):  
 print(all\_vectors[vector\_number][j], end="")  
 print(" ")  
 for j in range(i):  
 temp\_vector[j] = (1 if temp\_vector[j] == 0 else 0)  
 j += 2  
 count += 1  
 print("New vector:")  
 for j in range(len(temp\_vector)):  
 print(temp\_vector[j], end="")  
 print("")  
 print("Start number: ", vector\_number)  
 print("Recognizing number: ", network.recognize(temp\_vector))  
 print("Changing bit: ", count)  
 print("")  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

*Результаты работы программы:*

*Выводы по лабораторной работы:* Изучила обучение и функционирование искусственной нейронной сети (ИНС) векторного квантователя при решении задач распознавания образов.