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)

Мозоль И. П.

Проверил:

Крощенко А.А.

2021

Вариант 1

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

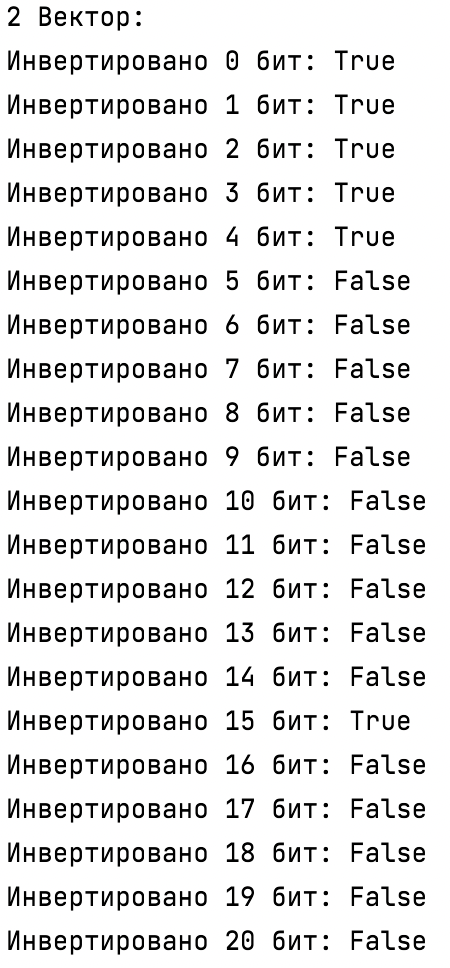
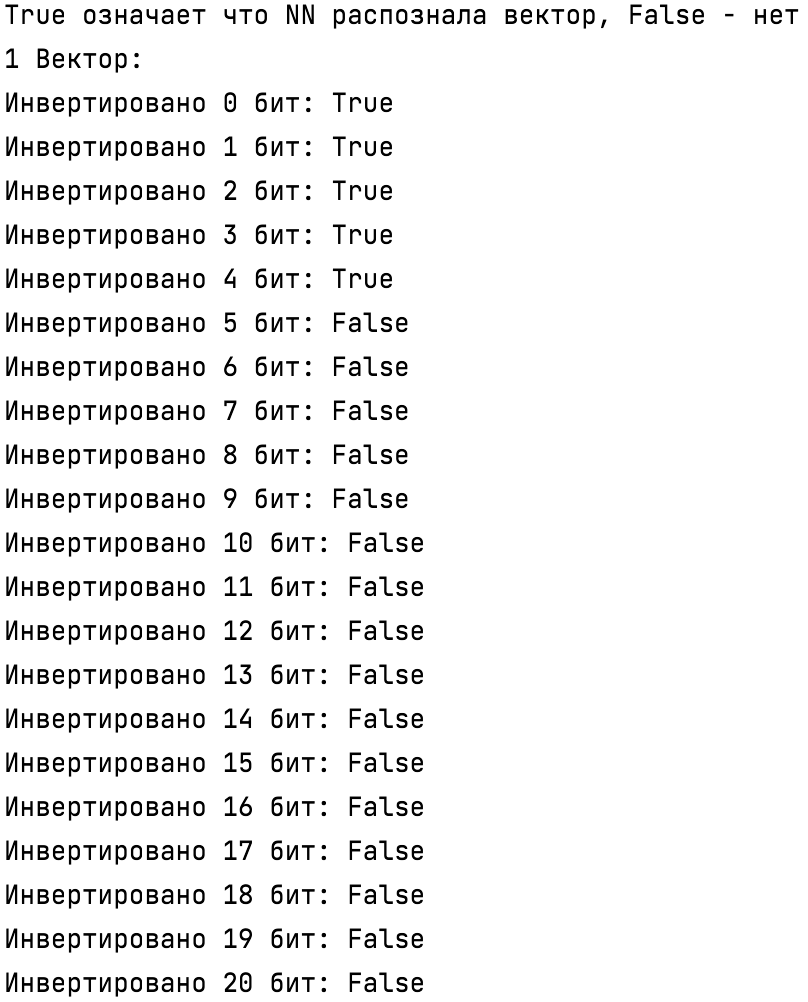
***Порядок выполнения работы.***

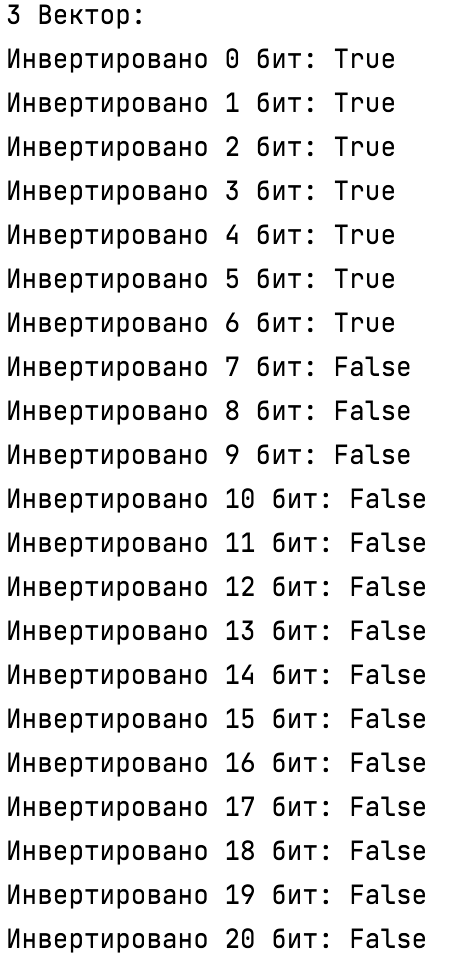
1. Изучить теоретические сведения, приведенные в данных методических указания и методических указаниях для работ №3 и №4.
2. Написать на любом ЯВУ программу моделирования нелинейной ИНС для распознавания образов. Рекомендуется использовать сигмоидную функцию, но это не является обязательным. Количество НЭ в скрытом слое взять согласно варианту работы №3. Его можно варьировать, если сеть не обучается или некорректно функционирует.
3. 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.
4. Оформить отчет, содержащий:
5. Титульный лист,
6. Цель работы,
7. Задание,
8. График изменения ошибки в зависимости от итерации.
9. Описание результатов распознавания.
10. Выводы по лабораторной работе.

**Код программы:**

import numpy as np  
  
def calc\_S\_hid(S\_hid):  
 return 1 / (1 + np.exp(-S\_hid))  
  
def training(inputs, ideal\_outputs, w\_out, w\_hid, T\_out, T\_hid, epochs = 1000):  
 for epoch in range(epochs + 1):  
 for r\_i in np.random.choice(3, 3, replace=False):  
 change\_parametres(inputs[r\_i], ideal\_outputs[r\_i], w\_out, w\_hid, T\_out, T\_hid)  
 return w\_out, w\_hid, T\_out, T\_hid  
  
def change\_parametres(inputs, outputs, w\_out, w\_hid, T\_out, T\_hid):  
 alpha = 0.1  
 y\_hid = calc\_S\_hid(np.dot(w\_hid, inputs) - T\_hid)  
 y = calc\_S\_hid(np.dot(w\_out, y\_hid) - T\_out)  
 gamma\_out = (y - outputs).reshape(1, -1)  
 gamma\_hid = (gamma\_out \* w\_out.T \* y \* (1 - y)).sum(axis=1).reshape(1, -1)  
 w\_out -= alpha \* gamma\_out.reshape(-1, 1) \* y\_hid \* (y \* (1 - y)).reshape(-1, 1)  
 T\_out += (alpha \* gamma\_out \* y \* (1 - y)).reshape(-1)  
 w\_hid -= alpha \* gamma\_hid.reshape(-1, 1) \* inputs.reshape(1, -1) \* (y\_hid \* (1 - y\_hid)).reshape(-1, 1)  
 T\_hid += alpha \* gamma\_hid.reshape(-1) \* y\_hid \* (1 - y\_hid)  
  
def predict(inputs, w\_hid, w\_out, T\_hid, T\_out):  
 hidden\_outputs = calc\_S\_hid(np.dot(w\_hid, inputs) - T\_hid)  
 return calc\_S\_hid(np.dot(w\_out, hidden\_outputs) - T\_out)  
  
def reverse\_bit(arr, index):  
 arr[index] ^= 1  
 return arr  
  
v\_predict = np.vectorize(predict, excluded=[1, 2, 3, 4])  
  
def main():  
 input = 20  
 hidden = 2  
 out = 3  
 input\_bits = np.array([  
 [0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0],  
 [1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],  
 [1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1]  
 ])  
 ref = np.array([[1, 0, 0], [0, 1, 0], [0, 0, 1]])  
 epoch = 30000  
 w\_hid = np.random.normal(0, 1, (hidden, input))  
 w\_out = np.random.normal(0, 1, (out, hidden))  
 T\_hid = np.random.normal(0, 1, hidden)  
 T\_out = np.random.normal(0, 1, out)  
 w\_out, w\_hid, T\_out, T\_hid = training(  
input\_bits, ref, w\_out.copy(), w\_hid.copy(), T\_out.copy(), T\_hid.copy(), epoch)  
 print('True означает что NN распознала вектор, False - нет')  
 for i, ideal\_out in enumerate(ref):  
 print(f"{i + 1} Вектор:")  
 cur\_vector = input\_bits[i]  
 output = predict(cur\_vector, w\_hid, w\_out, T\_hid, T\_out)  
 is\_recognize = ideal\_out[output.argmax()] == 1  
 print(f"Инвертировано 0 бит: {is\_recognize}")  
 inverted\_arr = cur\_vector.copy()  
 for j\_i, j in enumerate(np.random.choice(20, 20, replace=False)):  
 reverse\_bit(inverted\_arr, j)  
 output = predict(inverted\_arr, w\_hid, w\_out, T\_hid, T\_out)  
 is\_recognize = ideal\_out[output.argmax()] == 1  
 print(f"Инвертировано {j\_i + 1} бит: {is\_recognize}")  
main()

**Результат за 30000 эпох:**



****

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