

Министерство образования Республики Беларусь Учреждение
образования
«Белорусский государственный университет информатики и
радиоэлектроники»
Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе

№2 по курсу:

«Модели решения задач в интеллектуальных системах»

Вариант №11

Выполнил студент группы 021702:

Семченков Н.А.

Проверил:

Жук А.А

МИНСК 2022

1. ЦЕЛЬ

Ознакомиться, проанализировать и получить навыки реализации модели релаксационной нейронной сети для задачи распознавания образов.

2. ПОСТАНОВКА ЗАДАЧИ

Реализовать модель двунаправленной ассоциативной памяти.

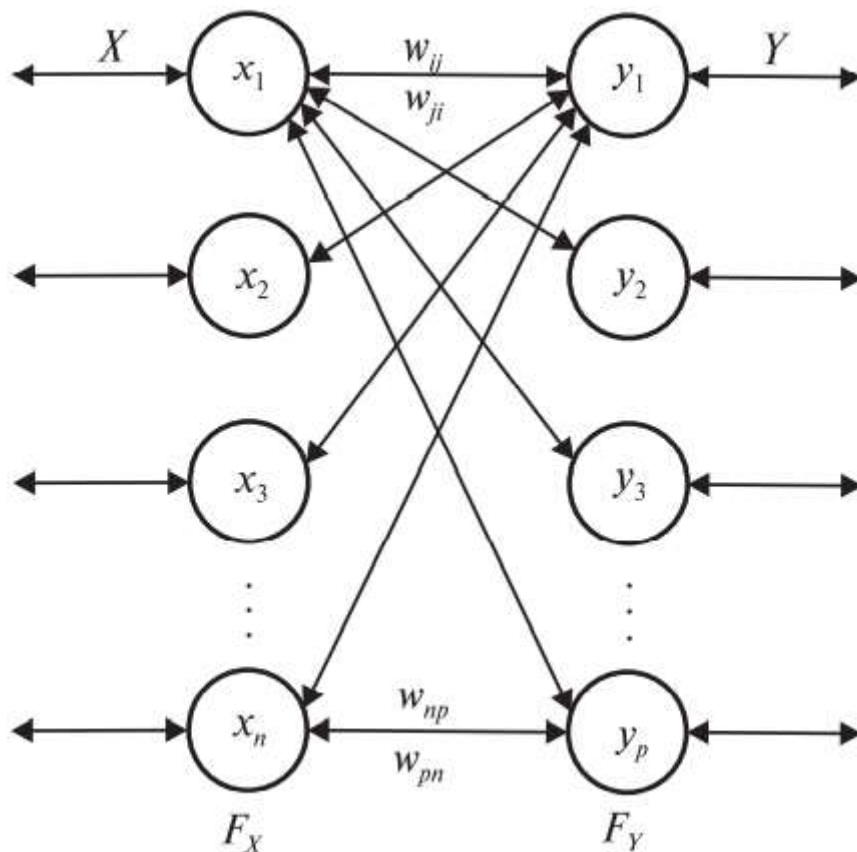
3. ОПИСАНИЕ МОДЕЛИ

Данные:

set input size – размер векторов входных/выходных данных ;

ser out size – размер векторов выходных/входных данных;

Двунаправленная ассоциативная память представляет собой нейронную сеть, состоящую из двух слоев нейронных элементов. Нейронные элементы каждого из слоев могут быть как входными, так и выходными.



Для обучения двунаправленной памяти используется правило Хебба. Пусть задано L пар векторов X и Y :

$$X_k = [x_1^k, x_2^k, \dots, x_n^k],$$

$$Y_k = [y_1^k, y_2^k, \dots, y_p^k],$$

где $k = \overline{1, L}$; L — количество образов одного типа, хранимых в нейронной сети.

Тогда в соответствии с правилом Хебба весовые коэффициенты такой сети определяются следующим образом:

$$w_{ij} = \sum_{k=1}^L x_i^k y_j^k \quad (7.81)$$

или в матричной форме:

$$W = X^T Y. \quad (7.82)$$

Найдем весовые коэффициенты W' обратного слоя:

$$W' = W^T. \quad (7.83)$$

Взвешенную сумму S_j j -го нейронного элемента слоя F_y вычислим по формуле

$$S_j = \sum_{i=1}^n x_i w_{ij}. \quad (7.84)$$

Получим выходные значения нейронов слоя F_y :

$$y_j(t+1) = F(S_j) = \begin{cases} 1, & \text{если } S_j > 0, \\ y_j(t), & \text{если } S_j = 0, \\ -1, & \text{если } S_j < 0, \end{cases}$$

где $j = \overline{1, p}$; F — оператор нелинейного преобразования. Взвешенную сумму i -го нейронного элемента слоя F_x можно представить как

$$S_i = \sum_{j=1}^p y_j w'_{ji}. \quad (7.85)$$

Алгоритм функционирования двунаправленной ассоциативной памяти можно представить в виде следующих шагов.

1. На входные слои F_X и F_Y подаются входные образы X и Y или только один из них.
2. Активация нейронных элементов в слое F_X передается через весовую матрицу W на слой F_Y .
3. Вычисляются выходные значения слоя F_Y .
4. Выходные значения слоя F_Y через весовую матрицу W' поступают на слой F_X .
5. Рассчитываются выходные значения нейронных элементов слоя F_X .
6. Процедура повторяется, начиная с п. 2, пока все выходные значения нейронных элементов слоев F_X и F_Y не перестанут изменяться. Это положение равновесия называется резонансом.

4. РЕЗУЛЬТАТЫ

В лабораторной работе есть стандартные входные данные , а так же можно ввести свои. Покажем пример на стандартных.

Входные данные :

$$\text{Set } \mathbf{A} : X_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad X_2 = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \quad X_3 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \quad X_4 = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

Выходные данные :

$$\text{Set } \mathbf{B} : Y_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad Y_2 = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \quad Y_3 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \quad Y_4 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}$$

Матрица весов:

$$W = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} -1 & 1 & -1 \end{bmatrix} \\ = \begin{bmatrix} 4 & 0 & 4 \\ 4 & 0 & 4 \\ 0 & 4 & 0 \\ 0 & 4 & 0 \\ 4 & 0 & 4 \\ 4 & 0 & 4 \end{bmatrix}$$

Протестируем входные данные:

```
Output of input pattern1
[[1]
 [1]
 [1]]

Output of input pattern2
[[-1]
 [-1]
 [-1]]

Output of input pattern3
[[ 1]
 [-1]
 [ 1]]

Output of input pattern4
[[-1]
 [ 1]
 [-1]]
```

Теперь на вход подадим выходные данные и протестируем их:

```
Testing for target patterns: Set B
```

```
Output of input pattern1
```

```
[[1]
 [1]
 [1]
 [1]
 [1]
 [1]]
```

```
Output of input pattern2
```

```
[[ -1]
 [ -1]
 [ -1]
 [ -1]
 [ -1]
 [ -1]]
```

```
Output of input pattern3
```

```
[[ 1]
 [ 1]
 [ -1]
 [ -1]
 [ 1]
 [ 1]]
```

```
Output of input pattern4
```

```
[[ -1]
 [ -1]
 [ 1]
 [ 1]
 [ -1]
 [ -1]]
```

Теперь попробуем подать на вход зашумленный образ:

```

Проверить работоспособность сети при распознавании эталонных образов? (yes), or any other letter for(no).
yes
Введите список чисел размером 6 или 3 , разделённых пробелом:
-1 -1 1

Testing for target patterns: Set B

Output of input pattern1
[[-1]
 [-1]
 [-1]
 [-1]
 [-1]
 [-1]]

```

Таким образом, в результате релаксации сеть установилась в состояние равновесия и распознала искаженный образ.

Так же есть возможность работать с зашифрованной информацией

Пример:

Таким данным(Танк)

1	- - - - # - - - -
2	- # - # # # - # -
3	- - - # # # - - -
4	- # - - - - - # -

Соответствуют такие данные

1	##### - - - - - # - - - - - # - - # - - - # - - # -
2	- - - # - - - - - # - # - - - - ##### - - - # # - - -
3	- - - # - - - - - ##### - - - - ##### - - - # # - - -
4	- - - - - - - - # - - - - # - - - # - - # - - - # - - # -

Посмотрим выводы в коде:

Входные и выходные данные -

```

"C:\PyCharm\5 сем\NeuralNetwork Two\venv\Scripts\python.exe" "C:/PyCharm/5
Взять стандартные set(1) / свои(2) или взять зашифрованные символы(3):
3
Выберите входной set:
1) Танк
1
inputset
[[-1 -1 -1 -1]
 [-1 1 -1 1]
 [-1 -1 -1 -1]
 [-1 1 1 -1]
 [ 1 1 1 -1]
 [-1 1 1 -1]
 [-1 -1 -1 -1]
 [-1 1 -1 1]
 [-1 -1 -1 -1]]
targetset
[[ 1 1 1 1 1 1 1 -1 -1 -1 -1 -1 1 -1 -1 -1 -1 -1 1 -1 -1 -1
 -1 -1 1 -1 -1 1 -1]
 [-1 -1 -1 1 -1 -1 -1 -1 -1 -1 1 -1 1 -1 -1 -1 -1 1 1 1 -1
 -1 -1 1 1 -1 -1 -1]
 [-1 -1 -1 1 -1 -1 -1 -1 -1 1 1 1 1 1 -1 -1 -1 1 1 1 -1
 -1 -1 1 1 -1 -1 -1]
 [-1 -1 -1 -1 -1 -1 -1 -1 1 -1 -1 -1 -1 -1 1 -1 -1 1 -1 -1
 -1 -1 1 -1 -1 1 -1]]

```

Матрица весов –

```

Weight matrix:
[[ 2  2  2 -2  2  2  2  4  4  2  2  0  0  0  2  2  4  4  4 -4  0  0 -4  4
  4  4 -4  0  4  0  4]
 [-2 -2 -2 -2 -2 -2 -2  0  0  2 -2  0 -4  0 -2  2  0  0  0  0  0  0  0
  0  0  0  0  0  0]
 [ 2  2  2 -2  2  2  2  4  4  2  2  0  0  0  2  2  4  4  4 -4  0  0 -4  4
  4  4 -4  0  4  0  4]
 [-2 -2 -2  2 -2 -2 -2  0  0 -2  2  4  0  4  2 -2  0  0  0  0  4  4  0  0
  0  0  0  4  0 -4  0]
 [ 0  0  0  4  0  0  0 -2 -2 -4  0  2  2  2  0 -4 -2 -2 -2  2  2  2  2 -2
 -2 -2  2  2 -2 -2 -2]
 [-2 -2 -2  2 -2 -2 -2  0  0 -2  2  4  0  4  2 -2  0  0  0  0  4  4  0  0
  0  0  0  4  0 -4  0]
 [ 2  2  2 -2  2  2  2  4  4  2  2  0  0  0  2  2  4  4  4 -4  0  0 -4  4
  4  4 -4  0  4  0  4]
 [-2 -2 -2 -2 -2 -2 -2  0  0  2 -2  0 -4  0 -2  2  0  0  0  0  0  0  0  0
  0  0  0  0  0  0]
 [ 2  2  2 -2  2  2  2  4  4  2  2  0  0  0  2  2  4  4  4 -4  0  0 -4  4
  4  4 -4  0  4  0  4]]

```

Тест входных данных:

Output of input pattern1	Output of input pattern2	Output of input pattern3	Output of input pattern4
[[1]	[[-1]	[[-1]	[[-1]
[1]	[-1]	[-1]	[-1]
[1]	[-1]	[-1]	[-1]
[1]	[1]	[1]	[-1]
[1]	[-1]	[-1]	[-1]
[1]	[-1]	[-1]	[-1]
[1]	[-1]	[-1]	[-1]
[-1]	[-1]	[-1]	[-1]
[-1]	[-1]	[-1]	[-1]
[-1]	[-1]	[-1]	[1]
[-1]	[-1]	[1]	[-1]
[-1]	[1]	[1]	[-1]
[1]	[-1]	[1]	[-1]
[-1]	[1]	[1]	[-1]
[-1]	[1]	[1]	[-1]
[-1]	[-1]	[1]	[-1]
[-1]	[-1]	[-1]	[1]
[-1]	[-1]	[-1]	[-1]
[-1]	[-1]	[-1]	[-1]
[1]	[1]	[1]	[1]
[-1]	[1]	[1]	[-1]
[-1]	[1]	[1]	[-1]
[-1]	[1]	[1]	[-1]
[-1]	[-1]	[-1]	[-1]
[1]	[1]	[1]	[1]
[-1]	[1]	[1]	[-1]
[-1]	[-1]	[-1]	[-1]
[1]	[-1]	[-1]	[1]
[-1]	[-1]	[-1]	[-1]

Подадим на вход выходные данные:

Output of input pattern1	Output of input pattern2	Output of input pattern3	Output of input pattern4
[[-1]	[[-1]	[[-1]	[[-1]
[-1]	[1]	[-1]	[1]
[-1]	[-1]	[-1]	[-1]
[-1]	[1]	[1]	[-1]
[1]	[1]	[1]	[-1]
[-1]	[1]	[1]	[-1]
[-1]	[-1]	[-1]	[-1]
[-1]	[1]	[-1]	[1]
[-1]	[-1]	[-1]	[-1]

Также подадим на вход зашумленные образы:

1	----#----
2	----##-#-
3	---#-#----
4	-#-----#-

1	-#####--#-#-----#-#-----#-
2	----#-----#-----###-----##-
3	--#-----#-#-#-----###-----##-
4	--#-----#-----#-#-----#-

Тест входных данных:

Output of input pattern1	Output of input pattern2	Output of input pattern3	Output of input pattern4
[1]	[-1]	[-1]	[-1]
[1]	[-1]	[-1]	[-1]
[1]	[-1]	[1]	[-1]
[1]	[-1]	[-1]	[-1]
[1]	[-1]	[-1]	[-1]
[1]	[-1]	[-1]	[-1]
[-1]	[-1]	[-1]	[-1]
[-1]	[-1]	[-1]	[-1]
[-1]	[-1]	[-1]	[1]
[-1]	[-1]	[1]	[-1]
[-1]	[1]	[1]	[-1]
[1]	[-1]	[1]	[-1]
[-1]	[1]	[1]	[-1]
[-1]	[-1]	[1]	[-1]
[-1]	[-1]	[-1]	[1]
[-1]	[-1]	[-1]	[-1]
[-1]	[-1]	[-1]	[-1]
[-1]	[-1]	[-1]	[-1]
[1]	[-1]	[1]	[1]
[-1]	[-1]	[1]	[-1]
[-1]	[1]	[1]	[-1]
[-1]	[-1]	[-1]	[-1]
[-1]	[-1]	[-1]	[-1]
[1]	[1]	[1]	[1]
[-1]	[1]	[1]	[-1]
[-1]	[-1]	[-1]	[-1]
[1]	[1]	[1]	[1]
[-1]	[-1]	[-1]	[-1]
[1]	[-1]	[-1]	[1]
[-1]	[-1]	[-1]	[-1]

Подадим на вход выходные данные:

Output of input pattern1	Output of input pattern2	Output of input pattern3	Output of input pattern4
[-1]	[-1]	[-1]	[-1]
[-1]	[1]	[-1]	[1]
[-1]	[-1]	[-1]	[-1]
[-1]	[1]	[1]	[-1]
[1]	[1]	[1]	[1]
[-1]	[1]	[1]	[-1]
[-1]	[-1]	[-1]	[-1]
[-1]	[1]	[-1]	[1]
[-1]	[-1]	[-1]	[-1]

Таким образом, в результате релаксации сеть установилась в состояние равновесия и распознала искаженный образ.

Вывод :

В ходе лабораторной работы была реализована модель двунаправленной ассоциативной памяти. В качестве функции активации использовалась модифицированная функция знака. На практике были получены результаты распознавания образов с помощью модели релаксационной нейронной сети.

