

Лабораторная работа №6

Нейронная сеть Хемминга для распознавания изображений

1. Цель лабораторных занятий

Приобретение и закрепление знаний и получение практических навыков работы с нейронной сетью Хемминга.

2. Краткие сведения из теории

2.1. Сеть Хемминга

Сеть Хемминга – это одна из наиболее многообещающих распознающих и классифицирующих нейронных сетей. В этой сети черно-белые изображения представляются в виде m -мерных биполярных векторов. Свое название она получила от расстояния Хемминга, которое используется в сети в мере сходства R изображений входного и эталонных, хранимых с помощью весов связей сети. Мера сходства определяется соотношением

$$R = m - R_x \quad (1)$$

где m – число компонент входного и эталонных векторов; R_x – расстояние Хемминга между векторами.

Определение 1. Расстоянием Хемминга между двумя двоичными векторами называется число компонент, в которых векторы различны.

В силу определения расстояния Хемминга мера сходства изображений (1) может быть задана и как число a компонент двоичных векторов, в которых они совпадают: $R = a$.

Запишем для биполярных векторов $S = (s_1, \dots, s_m)$ и $Z = (z_1, \dots, z_m)$ их скалярное произведение через число совпадающих и отличающихся компонент:

$$SZ = \sum_{i=1}^m s_i z_i = a - d, \quad (2)$$

где a – число одинаковых компонент векторов; d – число различных компонент векторов S и Z .

Поскольку m – размерность векторов, то $m = a + d$, следовательно, скалярное произведение (2) можно записать в виде

$$SZ = 2a - m.$$

Отсюда несложно получить:

$$a = m/2 + \frac{SZ}{2} = \frac{m}{2} + \frac{1}{2} \sum_{i=1}^m s_i z_i. \quad (3)$$

Правую часть выражения (3) можно рассматривать как входной сигнал нейрона, имеющего m синапсов с весовыми коэффициентами $z_i/2$ ($i = \overline{1, m}$) и смещением $m/2$. Синапсы нейрона воспринимают m компонент входного вектора $S = (s_1, \dots, s_m)$. Такая интерпретация правой части выражения (3) приводит к архитектуре нейронной подсети, изображенной в нижней части рис. 1. Одни авторы сеть, изображенную на рис 1, называют сетью Хемминга,

другие сетью Хемминга называют только ее нижнюю часть, считая, что приведенная сеть состоит из двух подсетей –Хемминга и Maxnet. Мы будем придерживаться первой точки зрения.

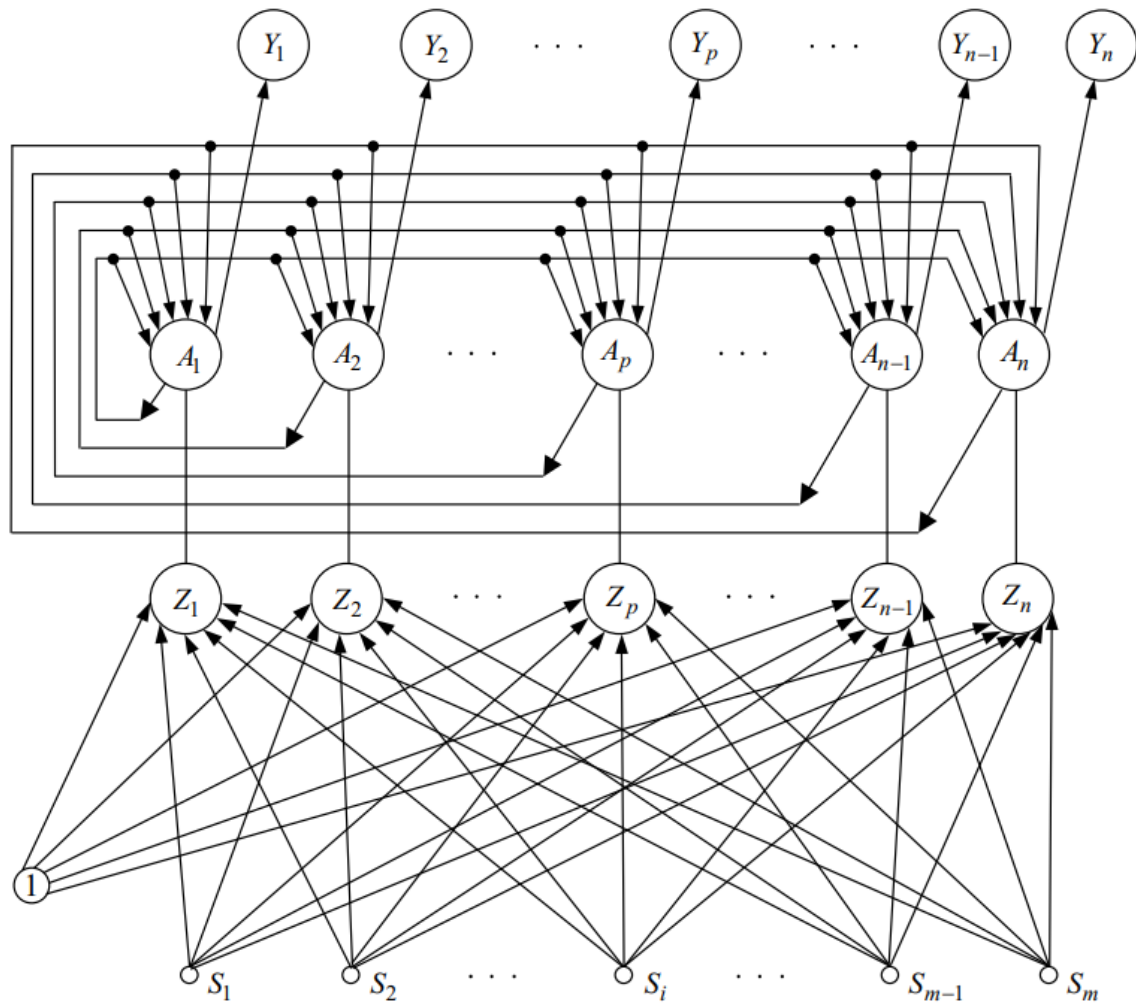


Рис. 1. Сеть Хемминга

Сеть Хемминга имеет m входных нейронов S_1, \dots, S_m , воспринимающих биполярные компоненты s_1^q, \dots, s_m^q входных изображений S^q ($q = \overline{1, L}$). Выходные сигналы S -элементов определяются соотношением

$$U_{\text{вых.}Si} = \begin{cases} +1, & \text{если } s_i^q = 1, \\ -1, & \text{если } s_i^q = -1, \end{cases} \quad (4)$$

т. е. выходной сигнал S -элемента повторяет его входной сигнал:

$$U_{вых.Si} = U_{ex.Si} = s_i^q.$$

Каждый нейрон S_j ($j = \overline{1, m}$) связан со входом каждого элемента Z_k ($k = \overline{1, n}$). Веса этих связей w_{1k}, \dots, w_{mk} содержат информацию о k -м эталонном изображении $V^k = (v_1^k, \dots, v_m^k)$:

$$w_{1k} = v_1^k / 2, \dots, w_{mk} = v_m^k / 2. \quad (5)$$

Функции активации Z -элементов описывается соотношением

$$g_Z(U_{ex.}) = \begin{cases} 0, & \text{если } U_{ex.} \leq 0, \\ k_1 U_{ex.}, & \text{если } 0 \leq U_{ex.} \leq U_n, \\ U_n, & \text{если } U_{ex.} > U_n, \end{cases} \quad (6)$$

где $U_{ex.}$ – входной сигнал нейрона; k_1, U_n – константы.

При предъявлении входного изображения $S^* = (s_1^*, \dots, s_m^*)$ каждый Z -нейрон рассчитывает свой входной сигнал в соответствии с выражением вида (3):

$$U_{ex.Zk} = m/2 + \sum_{i=1}^m w_{ik} S_i^*, \quad (7)$$

и с помощью функций активации, определяет выходной сигнал $U_{ex.Zk}$. Выходные сигналы $U_{вых.Z1}, \dots, U_{вых.Zn}$ Z -элементов являются входными сигналами a_1, \dots, a_n верхней подсети, которой является сеть Maxnet. Функции активации нейронов A_p ($p = \overline{1, n}$) и веса их связей задаются соотношениями

$$g(U_{ex.}) = \begin{cases} U_{ex.}, & \text{если } U_{ex.} > 0, \\ 0, & \text{если } U_{ex.} \leq 0. \end{cases}$$

$$w_{ij} = \begin{cases} 1, & \text{если } i = j, \\ -\varepsilon, & \text{если } i \neq j, \quad i, j = \overline{1, n}, \end{cases}$$

где ε – константа, удовлетворяющая неравенствам $0 < \varepsilon \leq 1/n$.

Сеть функционирует циклически, динамика нейронов описывается итерационным выражением

$$U_i(t+1) = g(U_i(t) - \varepsilon \sum_{j=1, j \neq i}^n U_j(t)), \quad i = \overline{1, n},$$

при начальных условиях

$$U_i(0) = a_i = U_{\text{вх.}Zi}, \quad i = \overline{1, n}. \quad (8)$$

Если среди входных сигналов a_1, \dots, a_n нейронов A_1, \dots, A_n имеется один наибольший сигнал a_p ($p \in \{1, 2, \dots, n\}$), то в результате итерационного процесса в подсети Махпет только один нейрон A_p останется с выходным сигналом, большим нуля, т.е. станет "победителем". Поскольку выходные сигналы $U_1, \dots, U_p, \dots, U_n$ A -элементов поступают на входы Y -нейронов, которые имеют функцию активации вида

$$g_Y(U_{\text{вх.}}) = \begin{cases} 1, & \text{если } U_{\text{вх.}} > 0, \\ 0, & \text{если } U_{\text{вх.}} \leq 0, \end{cases} \quad (9)$$

то в результате на выходе сети Хемминга только один нейрон Y_p окажется с единичным выходным сигналом. Единичный выход этого нейрона и нулевые всех остальных и будут указывать на то, что предъявленное изображение $S^* = (s_1^*, \dots, s_m^*)$ наиболее близко, в смысле заданной меры близости (1), к эталонному изображению $V^P = (v_1^P, \dots, v_m^P)$.

Существенное достоинство сети Хемминга заключается в том, что она не требует трудоемких вычислительных процедур для своего обучения. Заметный недостаток сети: она не выделяет два и более эталонных изображений, имеющих с предъявленным одинаковые максимальные меры близости.

Пример 1. Разработать сеть Хемминга, имеющую в качестве эталонных пять черно-белых изображений V^1, \dots, V^5 , приведенных на рис. 2. Определить реакцию сети на изображения, приведенные на рис. 3.

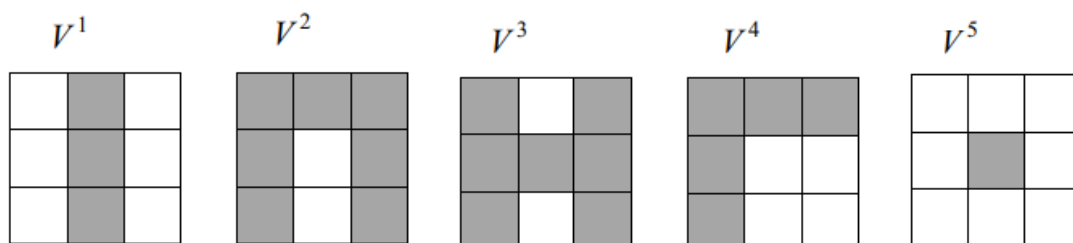


Рис. 2. Эталонные изображения

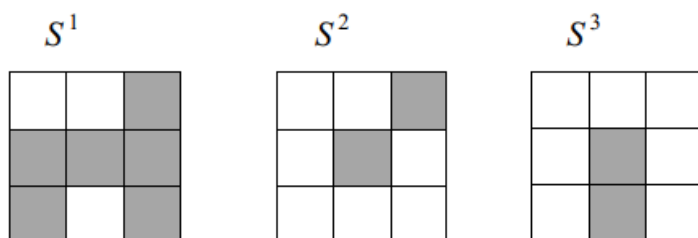


Рис. 3. Предъявляемые изображения

1	2	3
4	5	6
7	8	9

Рис. 4. Нумерация элементов изображений

Поскольку задано всего пять эталонных изображений, то сеть должна иметь по пять Z -, A - и Y -нейронов. Наличие девяти черно-белых элементов в изображениях рис. 2 и рис. 3 определяют девять S -нейронов, воспринимающих элементы входных изображений.

Пронумеруем элементы изображений рис. 2 и 3 в соответствии с рис. 4 и представим изображения V^p ($p = \overline{1,5}$) в векторной форме, используя биполярное представление векторов:

$$V^1 = (-1, 1, -1, -1, 1, -1, -1, 1, -1),$$

$$V^2 = (1, 1, 1, 1, -1, 1, 1, -1, 1),$$

$$V^3 = (1, -1, 1, 1, 1, 1, 1, -1, 1),$$

$$V^4 = (1, 1, 1, 1, -1, -1, 1, -1, -1),$$

$$V^5 = (-1, -1, -1, -1, 1, -1, -1, -1, -1).$$

Зная вектора эталонных изображений и их число по соотношению (5), рассчитаем матрицу $|W_{ik}|$ ($i = \overline{1,9}$, $k = \overline{1,5}$) весов связей нижней подсети сети Хемминга:

$$|W_{ik}| = \begin{matrix} & V^1(Z_1) & V^2(Z_2) & V^2(Z_3) & V^4(Z_4) & V^5(Z_5) \\ \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \end{matrix} & \begin{vmatrix} -0,5 & 0,5 & 0,5 & 0,5 & -0,5 \\ 0,5 & 0,5 & -0,5 & 0,5 & -0,5 \\ -0,5 & 0,5 & 0,5 & 0,5 & -0,5 \\ -0,5 & 0,5 & 0,5 & 0,5 & -0,5 \\ 0,5 & -0,5 & 0,5 & -0,5 & 0,5 \\ -0,5 & 0,5 & 0,5 & -0,5 & -0,5 \\ -0,5 & 0,5 & 0,5 & 0,5 & -0,5 \\ 0,5 & -0,5 & -0,5 & -0,5 & -0,5 \\ -0,5 & 0,5 & 0,5 & -0,5 & -0,5 \end{vmatrix} \end{matrix}, \quad (10)$$

где для наглядности строки и столбцы матрицы пронумерованы соответственно с помощью S -элементов и эталонных изображений V^p или нейронов Z_p , взятых в круглые скобки.

Смещения b_1, \dots, b_5 Z -нейронов рассчитываются с помощью выражения (11):

$$b_1 = b_2 = \dots = b_5 = \frac{m}{2} = \frac{9}{2} = 4,5. \quad (11)$$

Функции активации Z -нейронов зададим соотношением (6) при $k_1 = 0,1$ и $U_n = 1/k_1 = 1/0,1 = 10$. Функции активации Y -нейронов определим как функции (9). Константу ε , определяющую веса отрицательных связей в подсети Махпет, найдем из равенства $\varepsilon = 1/n$, а так как $n = 5$, то $\varepsilon = 0,2$.

Зная все параметры сети Хемминга, рассмотрим её функционирование при предъявлении изображения $S^1 = (-1, -1, 1, 1, 1, 1, 1, -1, 1)$ рис. 3.

После предъявления изображения S^1 на выходах S -нейронов в силу того, что их выходные сигналы повторяют входные (соотношения (4)), появится вектор сигналов $S_{\text{вых}.S} \equiv S^1$. Используя выходные сигналы S -элементов, каждый Z -нейрон рассчитывает свой входной сигнал в соответствии с выражением (7), матрицей весов (10) и смещением b_k , $k = \overline{1,5}$ (11):

$$U_{\text{ex}.Z1} = \frac{9}{2} + \sum_{i=1}^9 w_{i1} S_i^1 = 4,5 + (-0,5) \cdot (-1) + 0,5 \cdot (-1) + (-0,5) \cdot 1 + (-0,5) \cdot 1 + 0,5 \cdot 1 + \\ + (-0,5 \cdot 1) + (-0,5) \cdot 1 + (+0,5) \cdot (-1) + (-0,5) \cdot 1 = 2,$$

$$U_{\text{ex}.Z2} = \frac{9}{2} + \sum_{i=1}^9 w_{i2} S_i^1 = 4,5 + 0,5 \cdot (-1) + 0,5 \cdot (-1) + 0,5 \cdot 1 + 0,5 \cdot 1 + (-0,5) \cdot 1 + \\ + 0,5 \cdot 1 + 0,5 \cdot 1 + (-0,5) \cdot (-1) + 0,5 \cdot 1 = 6,$$

$$U_{\text{ex}.Z3} = \frac{9}{2} + \sum_{i=1}^9 w_{i3} S_i^1 = 4,5 + 0,5 \cdot (-1) + (-0,5) \cdot (-1) + 0,5 \cdot 1 + 0,5 \cdot 1 + 0,5 \cdot 1 + \\ + 0,5 \cdot 1 + (-0,5) \cdot (-1) + 0,5 \cdot 1 = 8,$$

$$U_{\text{ex}.Z4} = \frac{9}{2} + \sum_{i=1}^9 w_{i4} S_i^1 = 4,5 + 0,5 \cdot (-1) + 0,5 \cdot (-1) + 0,5 \cdot 1 + 0,5 \cdot 1 + (-0,5 \cdot 1) + \\ + (-0,5) \cdot 1 + 0,5 \cdot 1 + (-0,5) \cdot (-1) + (-0,5) \cdot 1 = 4,$$

$$U_{\text{ex}.Z5} = \frac{9}{2} + \sum_{i=1}^9 w_{i5} S_i^1 = 4,5 + (-0,5) \cdot (-1) + (-0,5) \cdot (-1) + (-0,5) \cdot 1 + (-0,5) \cdot 1 + \\ + 0,5 \cdot 1 + (-0,5) \cdot 1 + (-0,5) \cdot 1 + (-0,5) \cdot (-1) + (-0,5) \cdot 1 = 4.$$

По входному сигналу $U_{\text{ex}.Zk}$, используя свою функцию активации (6) при $k_1 = 0,1$ и $U_n = 10$, каждый Z -нейрон рассчитывает свой выходной сигнал:

$$U_{\text{вых}.Z1} = k_1 U_{\text{ex}.Z1} = 0,1 \cdot 2 = 0,2,$$

$$U_{\text{вых}.Z2} = k_1 U_{\text{ex}.Z2} = 0,1 \cdot 6 = 0,6,$$

$$U_{\text{вых}.Z3} = k_1 U_{\text{ex}.Z3} = 0,1 \cdot 8 = 0,8,$$

$$U_{\text{вых}.Z4} = k_1 U_{\text{ex}.Z4} = 0,1 \cdot 4 = 0,4,$$

$$U_{\text{вых}.Z5} = k_1 U_{\text{ex}.Z5} = 0,1 \cdot 4 = 0,4,$$

Вектор

$$U_{\text{вых}.Z} = (0,2; 0,6; 0,8; 0,4; 0,4) \quad (12)$$

является входным вектором подсети Maxnet, которая начинает итерационный процесс выделения максимального выходного сигнала при начальных условиях (12). Для $t = 1$ имеем

$$\begin{aligned}
 U_{вых.A1}(1) &= g(U_{вых.A1}(0) - \varepsilon \sum_{k=2}^5 U_{вых.Ak}(0)) = g(0,2 - 0,2(0,6 + 0,8 + 0,4 + 0,4)) = \\
 &= g(-0,24) = 0, \\
 U_{вых.A2}(1) &= g(U_{вых.A2}(0) - \varepsilon \sum_{k=1, k \neq 2}^5 U_{вых.Ak}(0)) = g(0,6 - 0,2(0,2 + 0,8 + 0,4 + 0,4)) = \\
 &= g(0,24) = 0,24, \\
 U_{вых.A3}(1) &= g(U_{вых.A3}(0) - \varepsilon \sum_{k=1, k \neq 3}^5 U_{вых.Ak}(0)) = g(0,8 - 0,2(0,2 + 0,6 + 0,4 + 0,4)) = \\
 &= g(0,48) = 0,48, \\
 U_{вых.A4}(1) &= g(U_{вых.A4}(0) - \varepsilon \sum_{k=1, k \neq 4}^5 U_{вых.Ak}(0)) = g(0,4 - 0,2(0,2 + 0,6 + 0,8 + 0,4)) = \\
 &= g(0) = 0, \\
 U_{вых.A5}(1) &= g(U_{вых.A5}(0) - \varepsilon \sum_{k=1}^4 U_{вых.Ak}(0)) = g(0,4 - 0,2(0,2 + 0,6 + 0,8 + 0,4)) = \\
 &= g(0) = 0.
 \end{aligned}$$

Используя вектор $(U_{вых.A1}(1), \dots, U_{вых.A5}(1))$ выходных сигналов A -элементов при $t = 1$, аналогичным образом рассчитывают выходные сигналы A -нейронов при $t = 2, 3$ и 4 . Результаты расчетов приведены в табл. 1.

Таблица 1. Результаты расчетов итерационного процесса в подсети Maxnet

Время	Величина выходных сигналов нейронов A_k ($k = \overline{1,5}$)				
	$U_{вых.A1}$	$U_{вых.A2}$	$U_{вых.A3}$	$U_{вых.A4}$	$U_{вых.A5}$
0	0,200	0,600	0,800	0,400	0,400
1	0,000	0,240	0,480	0,000	0,000
2	0,000	0,144	0,432	0,000	0,000
3	0,000	0,058	0,403	0,000	0,000
4	0,000	0,000	0,402	0,000	0,000

Итерационный процесс в подсети Maxnet заканчивается при $t = 5$, поскольку на этом шаге функционирование подсети не изменяется ни один выходной сигнал A -элементов. Вектор выходных сигналов A -элементов,

записанный в последней строке табл. 1, поступает на входы Y -элементов. Так как Y -нейроны имеют функцию активации вида (9), то на выходе только одного элемента Y_3 появится единичный сигнал. Появление этого сигнала говорит о том, что предъявленное изображение S^1 наиболее близко к эталонному изображению V^3 . Визуальное сопоставление рис. 2 и 3 подтверждает правильность работы сети.

Определим теперь реакцию сети при предъявлении изображения $S^2 = (-1, -1, 1, -1, 1, -1, -1, -1, -1)$. Поскольку расчеты аналогичны, то приведем только основные промежуточные результаты:

$$U_{ex.Z1}(S^2) = 6, U_{ex.Z2}(S^2) = 2, U_{ex.Z3}(S^2) = 4, U_{ex.Z4}(S^2) = 4, U_{ex.Z5}(S^2) = 8;$$

$$U_{вых.Z1}(S^2) = a_1 = 0,6; U_{вых.Z2}(S^2) = a_2 = 0,2; U_{вых.Z3}(S^2) = U_{вых.Z4}(S^2) = a_3 = a_4 = 0,4;$$

$$U_{вых.Z5}(S^2) = a_5 = 0,8.$$

Поскольку входной вектор $(0,6; 0,2; 0,4; 0,4; 0,8)$ подсети Maxnet содержит единственный максимальный элемент $a_5 = 0,8$, то в результате итерационного процесса на выходе только элемента A_5 окажется положительный выходной сигнал, который вызовет единичный сигнал на выходе нейрона Y_3 . Следовательно, предъявленное изображение наиболее близко к эталонному изображению V^5 , что подтверждает и визуальное сопоставление рис. 2 и 3.

Определим теперь реакцию сети Хемминга на входное изображение $S^3 = (-1, -1, -1, -1, 1, -1, -1, 1, -1)$ (рис. 3). При предъявлении изображения S^3 имеем: $U_{ex.Z1}(S^3) = 8, U_{ex.Z2}(S^3) = 0, U_{ex.Z3}(S^3) = 1, U_{ex.Z4}(S^3) = 2, U_{ex.Z5}(S^3) = 8$.

Поскольку сигналы $U_{ex.Z1}(S^3) = U_{ex.Z5}(S^3)$ являются одинаковыми максимальными входными сигналами, то одинаковыми будут и максимальные сигналы на выходах Z -элементов ($U_{вых.Z1}(S^3) = U_{вых.Z5}(S^3) = 0,8$) и на входах A -нейронов ($a_1(S^3) = a_5(S^3) = 0,8$). Следовательно, подсеть Maxnet не сможет выделить единственного максимального сигнала и в результате ее функционирования на всех выходах A - и Y -нейронов появятся нулевые сигналы.

Таким образом, сеть Хемминга не может определить, к какому из эталонных изображений наиболее близко предъявленное изображение S^3 .

Пример программной реализации сети Хемминга

На рис. 5. представлена программа (Python), которая является простым примером нейронной сети Хемминга для распознавания символов 3 на 3.

Функция `hamming_distance` (Хеммингово расстояние) принимает два массива и считает количество различий между ними.

Функция `recognize_symbol` (Распознавание символа) принимает на вход 3×3 матрицу и сравнивает её с известными символами, используя Хеммингово расстояние. Минимальное расстояние указывает на распознанный символ.

Используется `matplotlib` для отображения распознаваемого символа (в качестве визуализации).

Python ▾



```
1 import numpy as np # Импортируем библиотеку numpy для работы с массивами
2 import matplotlib.pyplot as plt # Импортируем matplotlib для визуализации данных
3
4 # Создание обучающего набора данных с символами
5 symbols = {
6     'L': np.array([[1, 0, 0], # Определяем символ 'L' как 3x3 массив
7                    [1, 0, 0], # Вторая строка
8                    [1, 1, 1]]), # Третья строка
9     'T': np.array([[1, 1, 1], # Определяем символ 'T'
10                   [0, 1, 0], # Вторая строка
11                   [0, 1, 0]]), # Третья строка
12     'X': np.array([[1, 0, 1], # Определяем символ 'X'
13                   [0, 1, 0], # Вторая строка
14                   [1, 0, 1]]) # Третья строка
15 }
16
17 # Функция для вычисления Хеммингового расстояния
18 def hamming_distance(a, b):
19     return np.sum(a != b) # Считаем количество различий между массивами a и b
20
21 # Функция для распознавания символа
22 def recognize_symbol(input_symbol):
23     distances = {} # Создаем словарь для хранения расстояний до известных символов
24     for symbol, template in symbols.items(): # Проходим по всем известным символам
25         distances[symbol] = hamming_distance(input_symbol, template) # Вычисляем расстояние
26
27     # Находим символ с минимальным расстоянием
28     recognized_symbol = min(distances, key=distances.get) # Находим символ с наименьшим расстоянием
29     return recognized_symbol # Возвращаем распознанный символ
30
31 # Пример использования
32 input_symbol = np.array([[0, 0, 0], # Входной символ для распознавания
33                          [1, 0, 0],
34                          [1, 1, 0]])
35
36 recognized = recognize_symbol(input_symbol) # Вызываем функцию распознавания
37 print(f'Распознанный символ: {recognized}') # Выводим распознанный символ
38
39 # Визуализация
40 fig, ax = plt.subplots() # Создаем фигуру и оси для графика
41 ax.imshow(input_symbol, cmap='gray', vmin=0, vmax=1) # Отображаем входной символ с цветовой картой
42 ax.set_title('Входной символ') # Устанавливаем заголовок для графика
43 plt.show() # Показываем график
```

Рис. 5. Пример реализации нейронной сети Хемминга на Python для распознавания изображений

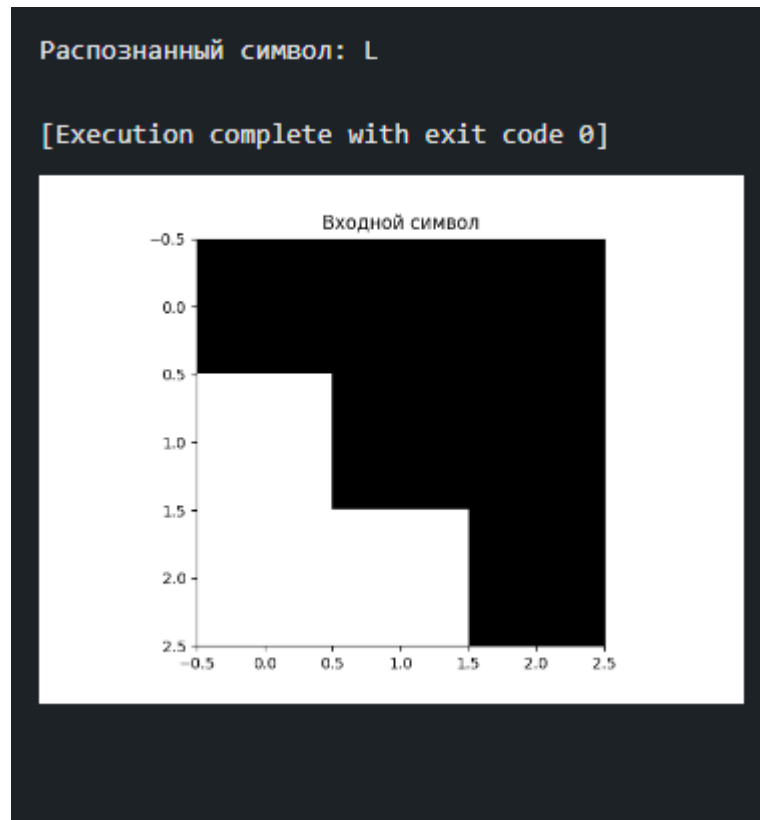


Рис. 7. Результат выполнения программы с распознанным изображением

3. Индивидуальные задания

3.1. Разработайте нейронную сеть Хемминга, которая сможет распознавать не менее 3 различных букв. При этом обоснуйте выбор:

- числа рецепторных нейронов;
- числа нейронов выходного слоя;
- вида функций активации нейронов каждого слоя;
- величин весов связей и смещений в подсети Хемминга.

3.2. Обучите нейронную сеть эталонным изображениям букв.

3.3. Исследуйте возможности сети по распознаванию искаженных изображений.

3.4. Подберите входное изображение, равноудаленное по расстоянию Хемминга от двух эталонных изображений. Какова реакция при предъявлении этого изображения? Можете ли Вы предложить способ более информативного поведения сети при предъявлении подобных изображений?

4. Содержание отчета

4.1. Тема лабораторных занятий.

4.2. Индивидуальное задание.

4.3. Результаты выполнения пунктов 3.1 – 3.4 индивидуального задания.