

**Министерство образования Российской Федерации
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Н. Э. БАУМАНА**

Факультет: Информатика и системы управления

Кафедра: Информационная безопасность

**«Интеллектуальные технологии информационной
безопасности»**

ЛАБОРАТОРНАЯ РАБОТА № 1

**«Исследование однослойных нейронных сетей на
примере моделирования булевых выражений»**

Вариант № 6

Преподаватель: Коннова Н.С.

Студент: Кошман А.А.

Группа: ИУ8-61

Москва, 2019

Оглавление

Цель работы	3
Постановка задачи	3
Задание № 1.....	4
Задание № 2.....	6
Задание № 3.....	8
Задание № 4.....	10
Выводы.....	12
Приложения	13

Цель работы

Исследовать функционирование простейшей нейронной сети (НС) на базе нейрона с нелинейной функцией активации и обучить ее по правилу Видроу – Хоффа

Постановка задачи

Получить модель булевой функции (БФ) на основе однослойной НС (единичный нейрон) с двоичными входами $x_1, x_2, x_3, x_4 \in \{0, 1\}$, единичным входом смещения $x_0 = 1$, синаптическими весами w_0, w_1, w_2, w_3, w_4 двоичным выходом $y_e \in \{0, 1\}$ и заданной нелинейной функцией активации (ФА) $f: R \rightarrow (0, 1)$

Для заданной БФ реализовать обучение НС с использованием:

- 1) всех комбинаций переменных x_1, x_2, x_3, x_4 ;
- 2) части возможных комбинаций переменных x_1, x_2, x_3, x_4 ;

Остальные комбинации являются тестовыми.

Таблица истинности

Получим таблицу истинности для моделируемой БФ:

$$F(x_1, x_2, x_3, x_4) = x_3 x_4 + \overline{x_1} + \overline{x_2}$$

X ₁	X ₂	X ₃	X ₄	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1

Задание № 1

Обучение НС с использованием всех комбинаций переменных x_1, x_2, x_3, x_4 , используя пороговую ФА :

$$f(net) = \begin{cases} 1, net \geq 0, \\ 0, net < 0; \end{cases}$$

Таблица 1. Параметры НС на последовательных эпохах (пороговая ФА)

k	Вектор весов w	Выходной вектор y	Суммарная ошибка E
0	[0, 0, 0, 0, 0]	[1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1]	5
1	[-0.3, 0.0, -0.3, 0.0, 0.6]	[0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1]	4
2	[-0.3, 0.0, -0.3, 0.3, 0.9]	[0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1]	5
3	[-0.6, -0.3, 0.0, 0.0, 1.2]	[0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1]	2
4	[-0.59, 0.0, 0.0, 0.0, 1.2]	[0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	1
5	[-0.89, 0.0, 0.0, 0.0, 0.89]	[0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	2
6	[-0.89, 0.0, 0.0, 0.3, 0.89]	[0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	2
7	[-0.89, 0.0, 0.0, 0.6, 0.89]	[0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1]	7

8	$[-1.2, -0.3, 0.0, 0.29, 1.5]$	$[0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$	2
9	$[-1.2, -0.3, 0.0, 0.59, 1.5]$	$[0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$	2
10	$[-1.2, -0.3, 0.3, 0.59, 1.5]$	$[0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1]$	2
11	$[-1.2, 0.0, 0.3, 0.59, 1.5]$	$[0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1]$	2
12	$[-1.2, 0.3, 0.3, 0.59, 1.5]$	$[0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$	1
13	$[-1.5, 0.3, 0.3, 0.59, 1.2]$	$[0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]$	0

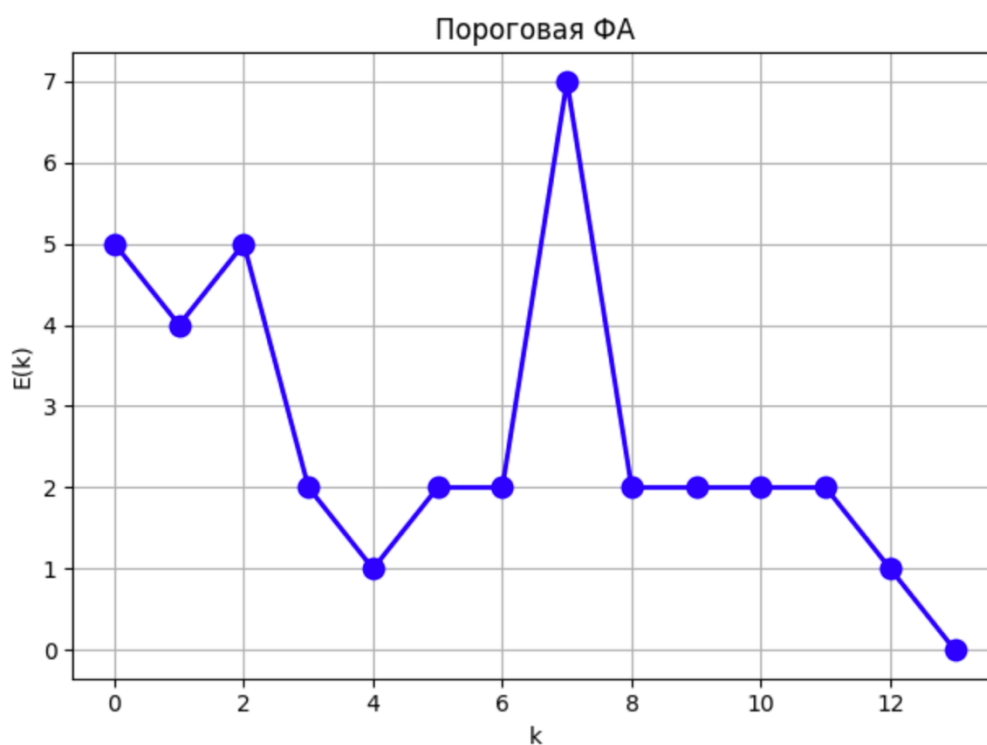


Рисунок 1. График суммарной ошибки НС по эпохам обучения (пороговая ФА)

Задание № 2

Обучения НС с использованием всех комбинаций переменных x_1, x_2, x_3, x_4 , используя логистическую ФА :

$$f(\text{net}) = \frac{1}{2} (\tanh(\text{net}) + 1)$$

Производная которой равна:

$$-\frac{1}{2} \tanh^2(x) + \frac{1}{2}$$

Таблица 2. Параметры НС на последовательных эпохах (логистическая ФА)

k	Вектор весов w	Выходной вектор y	Суммарная ошибка E
0	[0, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1]	9
1	[- 0.147, - 0.143, - 0.146, 0.002, 0.593]	[0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1]	2
2	[-0.122, 0.005, 0.001, 0.002, 0.618]	[0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	1
3	[-0.240, 0.005, 0.001, 0.002, 0.499]	[0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	2
4	[-0.231, 0.005, 0.001, 0.152, 0.509]	[0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	1
5	[-0.369, 0.005, 0.001, 0.152, 0.370]	[0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1]	4
6	[-0.365, 0.005, 0.009, 0.157, 0.517]	[0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	2
7	[-0.364, 0.005, 0.156, 0.157, 0.518]	[0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1]	3
8	[-0.512, 0.004, 0.008, 0.008, 0.519]	[0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	3
9	[-0.373, 0.004, 0.158, 0.147, 0.657]	[0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	1

10	[-0.512, 0.004, 0.158, 0.147, 0.518]	[0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	2
11	[-0.515, 0.004, 0.158, 0.293, 0.515]	[0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1]	4
12	[-0.517, 0.15, 0.009, 0.295, 0.662]	[0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1]	3
13	[-0.666, 0.001, 0.007, 0.146, 0.662]	[0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1]	1
14	[-0.516, 0.151, 0.007, 0.146, 0.812]	[0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	1
15	[-0.653, 0.151, 0.007, 0.146, 0.675]	[0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	2
16	[-0.656, 0.151, 0.007, 0.293, 0.672]	[0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	2
17	[-0.667, 0.151, 0.146, 0.293, 0.662]	[0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1]	0



Рисунок 2. График суммарной ошибки НС по эпохам обучения (логистическая ФА)

Задание № 3

Обучение НС с использованием части комбинаций переменных x_1, x_2, x_3, x_4 , используя пороговую ФА.

Последовательно увеличивая выборку количества векторов, найдем наименьшее количество необходимых для обучения векторов.

Минимальный набор обучающих векторов :

$$x^{(1)} = (0, 0, 0, 1); x^{(2)} = (0, 0, 1, 1); x^{(3)} = (1, 1, 0, 1); x^{(4)} = (1, 1, 1, 0);$$

Вектор синаптических коэффициентов :

$$w = [-1.5, 0.3, 0.3, 0.5(9), 1.2]$$

Для полного обучения потребовалось 20 эпох :

Таблица 3. Параметры НС на последовательных эпохах (пороговая ФА) при наборе из 4 векторов

k	Вектор весов w	Выходной вектор y	Суммарная ошибка E
0	[0, 0, 0, 0, 0]	[1, 0, 1, 1]	1
1	[-0.3, -0.3, -0.3, 0.0, 0.0]	[0, 0, 0, 1]	3
2	[0.0, -0.3, -0.3, 0.0, 0.6]	[1, 1, 0, 1]	3
3	[-0.3, -0.3, -0.3, -0.3, 0.6]	[1, 0, 0, 1]	3
4	[-0.3, -0.3, -0.3, -0.3, 0.8(9)]	[1, 0, 0, 1]	4
5	[-0.3, -0.3, -0.3, -0.3, 1.2]	[1, 0, 1, 0]	4
6	[-0.3, -0.3, -0.3, 0.0, 1.2]	[1, 1, 0, 0]	2

7	$[-0.3, 0.0, 0.0, 0.0, 1.2]$	$[1, 1, 1, 0]$	2
8	$[-0.6, 0.0, 0.0, 0.0, 0.8(9)]$	$[1, 0, 1, 0]$	1
9	$[-0.5(9), 0.0, 0.0, 0.3, 0.8(9)]$	$[1, 1, 0, 1]$	2
10	$[-0.8(9), 0.0, 0.0, 0.0, 0.8(9)]$	$[1, 0, 1, 0]$	3
11	$[-0.8(9), 0.0, 0.0, 0.3, 0.8(9)]$	$[1, 0, 1, 0]$	2
12	$[-0.8(9), 0.0, 0.0, 0.6, 0.8(9)]$	$[1, 0, 1, 1]$	2
13	$[-1.2, -0.3, -0.3, 0.5(9), 0.8(9)]$	$[0, 1, 0, 0]$	3
14	$[-0.8(9), 0.0, 0.0, 0.5(9), 1.2]$	$[1, 1, 0, 1]$	1
15	$[-1.2, 0.0, 0.0, 0.2(9), 1.2]$	$[1, 0, 1, 0]$	3
16	$[-1.2, 0.0, 0.0, 0.5(9), 1.2]$	$[1, 0, 1, 0]$	2
17	$[-1.2, 0.0, 0.0, 0.8(9), 1.2]$	$[1, 1, 0, 1]$	2
18	$[-1.5, 0.0, 0.0, 0.5(9), 1.2]$	$[0, 1, 0, 0]$	3
19	$[-1.2, 0.3, 0.3, 0.5(9), 1.5]$	$[1, 1, 1, 0]$	1
20	$[-1.5, 0.3, 0.3, 0.5(9), 1.2]$	$[0, 1, 1, 0]$	1



Рисунок 3. График суммарной ошибки НС по эпохам обучения с минимальным количеством наборов (пороговая ФА)

Задание № 4

Обучение НС с использованием части комбинаций переменных x_1, x_2, x_3, x_4 , используя логистическую ФА.

Последовательно увеличивая выборку количества векторов, найдем наименьшее количество необходимых для обучения векторов.

Минимальный набор обучающих векторов :

$$x^{(1)} = (0, 0, 0, 1); x^{(2)} = (0, 0, 1, 1); x^{(3)} = (1, 1, 0, 1); x^{(4)} = (1, 1, 1, 0);$$

Вектор синаптических коэффициентов :

$$w = [-0.49, 0.086, 0.086, 0.056, 0.461]$$

Для полного обучения потребовалось 11 эпох :

Таблица 4. Параметры НС на последовательных эпохах (логистическая ФА) при наборе из 4 векторов

к	Вектор весов w	Выходной вектор y	Суммарная ошибка E
0	[0, 0, 0, 0, 0]	[1, 0, 0, 1]	4
1	[0.027, 0.0402, 0.0402, 0.027, 0.137]	[1, 0, 1, 1]	3
2	[-0.108, -0.098, -0.098, 0.037, 0.139]	[1, 0, 0, 1]	4
3	[-0.107, -0.089, -0.089, 0.043, 0.277]	[1, 0, 0, 1]	4
4	[-0.086, -0.072, -0.072, 0.059, 0.43]	[1, 1, 0, 1]	3
5	[-0.217, -0.069, -0.069, -0.087, 0.446]	[1, 0, 1, 0]	2
6	[-0.213, -0.069, -0.069, 0.059, 0.451]	[1, 1, 0, 1]	3
7	[-0.36, -0.074, -0.074, -0.09, 0.454]	[1, 0, 0, 0]	3
8	[-0.221, 0.074, 0.074, 0.047, 0.592]	[1, 1, 1, 0]	1
9	[-0.352, 0.074, 0.074, 0.047, 0.461]	[1, 0, 1, 0]	2
10	[-0.354, 0.074, 0.074, 0.194, 0.46]	[1, 1, 0, 1]	3
11	[-0.49, 0.086, 0.086, 0.056, 0.461]	[0, 1, 1, 0]	0



Рисунок 4. График суммарной ошибки НС по эпохам обучения с минимальным количеством наборов (логистическая ФА)

Выводы

В процессе лабораторной работы было исследовано функционирование простейшей нейронной сети на базе нейрона с нелинейной функцией активации и обучение ее по правилу Видроу – Хоффа. Были обучены НС с использованием пороговой и логистической ФА. А также произведено обучение с использованием части комбинаций переменных.

В результате, НС с использованием пороговой ФА была быстрее обучена на минимальном количестве наборов, чем на полных наборах. А НС с использованием логистической ФА была, наоборот, обучена за меньшее количество эпох на полных наборах.

Приложения

Файл 'script.py' :

```
import numpy as np
import matplotlib.pyplot as plt
from itertools import combinations

n = 0.3

def Hemming(f1, f2): # возвращает расстояние Хэмминга между наборами f1 и f2
    e = 0
    for x1, x2 in zip(f1, f2):
        if(x1 != x2) : e += 1
    return e

def F(x): # возвращает результат моделируемой булевой функции
    return int(x[0] or x[1] or x[2]) and x[3]

def Y1(net): # возвращает результат пороговой ФА
    return 1 if net >= 0 else 0

def Y2(net): # возвращает результат логистической ФА
    out = 0.5 * (np.tanh(net) + 1)
    return 1 if out >= 0.5 else 0

def DeltaW1(x, q, net): # находит величину, на которую изменятся Wi, для пороговой ФА
    return n * q * x

def DeltaW2(x, q, net): # находит величину, на которую изменятся Wi, для логистической ФА
    return n * q * x * ((-0.5) * (np.tanh(net) ** 2) + 0.5)

def Net(x, w): # находит значение сетевого входа НС
    return sum([w_i * x_i for w_i, x_i in zip(w[1:], x)]) + w[0]

def LearningNN(X, Y, DeltaW): #производит обучение НС и возвращает вектор ошибок E(k)и вектор синаптических коэффициентов, на которых обучилась НС
    RightF = [F(x_i) for x_i in X]
    print(RightF, "\n")
    w = [0 for i in range(0, 5)]
    TryF = [0 for i in range(0, len(X))]
    E = [1]
    k = 0

    while E[k] != 0:
        print("\n\нк =", k, "\n")
        print("\nw =", w)

        for i in range(0, len(X)):
            net = Net(X[i], w)
            TryF[i] = Y(net)
            q = RightF[i] - TryF[i]
            for j in range(0, len(X[i])):
                w[j + 1] += DeltaW(X[i][j], q, net)
            w[0] += DeltaW(1, q, net)
        E.append(Hemming(RightF, TryF))
        k += 1
```

```

    print("TryF :", TryF)
    print("E[k] :", E[k])

    return E, w

def Graph(E, name): # строит график зависимости вектора ошибок от эпохи
    plt.plot(E[1:], 'bo-', linewidth = 2, markersize = 9 )
    plt.grid(True)
    plt.title(name)
    plt.xlabel('k')
    plt.ylabel('E(k)')
    plt.show()

def MinimizeSet(X, Y, DeltaW, name): # находит минимальные наборы из общей
выборки, на которых возможно обучение НС
    RightF = [F(x_i) for x_i in X]
    TryF = [0 for i in range(0, len(X))]
    for min_num in range(0, len(X) + 1):
        for min_x in list(combinations(X, min_num)):
            # print("\n", min_x)
            E, w = LearningNN(min_x, Y, DeltaW)
            for i in range(0, len(X)):
                net = Net(X[i], w)
                TryF[i] = Y2(net)
            if (Hemming(RightF, TryF) == 0) :
                Graph(E, name + "(с минимальным кол-вом наборов)")
            return min_x, w

if __name__=="__main__":

    X = np.unpackbits(np.array([[j] for j in range(0, pow(2, 4))],
dtype=np.uint8), axis=1)[: , 4:]

    E, w = LearningNN(X, Y1, DeltaW1)
    Graph(E, "Пороговая ФА")

    E, w = LearningNN(X, Y2, DeltaW2)
    Graph(E, "Логистическая ФА")

    min_x, w = MinimizeSet(X, Y1, DeltaW1, "Пороговая ФА")
    print("\n\nmin_x :", min_x, "\n\nw =", w)

    min_x, w = MinimizeSet(X, Y2, DeltaW2, "Логистическая ФА")
    print("\n\nmin_x :", min_x, "\n\nw =", w)

```