

Министерство образования Российской Федерации

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ
им. Н.Э. БАУМАНА**

Факультет: Информатика и системы управления
Кафедра: Информационная безопасность (ИУ8)

**ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ
ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ**

Лабораторная работа №1 на тему:
**«Исследование однослойных нейронных сетей на примере
моделирования булевых выражений»**

Вариант 23

Преподаватель:
Коннова Н.С.

Студент:
Соловьева Е.А.

Группа:
ИУ8-61

Москва 2016

Цель работы

Исследовать функционирование простейшей нейронной сети (НС) на базе нейрона с нелинейной функцией активации и ее обучение 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 $y \in \{0, 1\}$ и заданной нелинейной функцией активации (ФА) $f: R \rightarrow (0, 1)$.

Для заданной БФ реализовать обучение НС для двух случаев:

1. с использованием всех комбинаций переменных x_1, x_2, x_3, x_4 ;
2. с использованием части возможных комбинаций переменных x_1, x_2, x_3, x_4 ; остальные комбинации используются в качестве тестовых.

Ход работы

Получим таблицу истинности для моделируемой БФ:

$$\overline{(x_2 + x_4)}x_1 + x_1x_3$$

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
F	1	1	1	1	1	1	1	1	0	0	0	0	1	0	0	0

На начальном шаге $l = 0$ (эпоха $k = 0$) весовые коэффициенты берутся в виде:

$$w_0^{(0)} = w_1^{(0)} = w_2^{(0)} = w_3^{(0)} = w_4^{(0)} = 0$$

Норма обучения для всех случаев выбирается $\eta = 0.3$

1. Обучение НС с использованием всех комбинаций переменных x_1, x_2, x_3, x_4 .

1.1. Используя пороговую ФА:

$$f(\text{net}) = \begin{cases} 1, & \text{net} > 0, \\ 0, & \text{net} \leq 0 \end{cases}$$

Таблица 1 Параметры НС на последовательных эпохах (пороговая ΦA)

Номер эпохи, k	Вектор весов W , выходной вектор Y , суммарная ошибка E
0	$Y = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.0, 0.0, 0.0, 0.0, 0.0), E = 9$
1	$Y = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.0, -0.3, 0.0, 0.0, -0.3), E = 9$
2	$Y = (1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.3, -0.6, 0.0, 0.0, -0.3), E = 5$
3	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.6, -0.6, 0.0, 0.0, -0.3), E = 1$
4	$Y = (1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.6, -0.6, 0.0, 0.0, -0.6), E = 5$
5	$Y = (1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.6, -0.9, 0.0, 0.0, -0.6), E = 5$
6	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.9, -0.9, 0.0, 0.0, -0.6), E = 1$
7	$Y = (1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.9, -0.9, 0.0, 0.0, -0.9), E = 5$
8	$Y = (1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.9, -1.2, 0.0, 0.0, -0.9), E = 5$
9	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.2, -1.2, 0.0, 0.0, -0.9), E = 1$
10	$Y = (1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.2, -1.2, 0.0, 0.0, -1.2), E = 5$
11	$Y = (1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.2, -1.5, 0.0, -0.3, -0.9), E = 3$
12	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.5, -1.5, 0.0, 0.0, -0.9), E = 1$
13	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.5, -1.5, 0.0, 0.0, -1.2), E = 1$

14	$Y = (1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.5, -1.5, 0.0, -0.3, -1.2), E = 3$
15	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.5, -1.8, 0.0, -0.3, -0.9), E = 1$
16	$Y = (1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.5, -1.8, 0.0, -0.6, -0.9), E = 3$
17	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.8, -1.8, 0.0, -0.3, -0.9), E = 1$
18	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.8, -1.8, 0.0, -0.3, -1.2), E = 1$
19	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.8, -1.8, 0.0, -0.6, -1.2), E = 1$
20	$Y = (1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.8, -1.8, 0.0, -0.9, -1.2), E = 3$
21	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (1.8, -2.1, 0.0, -0.9, -0.9), E = 1$
22	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0),$ $W = (2.1, -1.8, 0.3, -0.9, -0.9), E = 1$
23	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0),$ $W = (2.1, -1.8, 0.6, -0.9, -0.9), E = 1$
24	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0),$ $W = (1.8, -2.1, 0.6, -0.9, -0.9), E = 0$

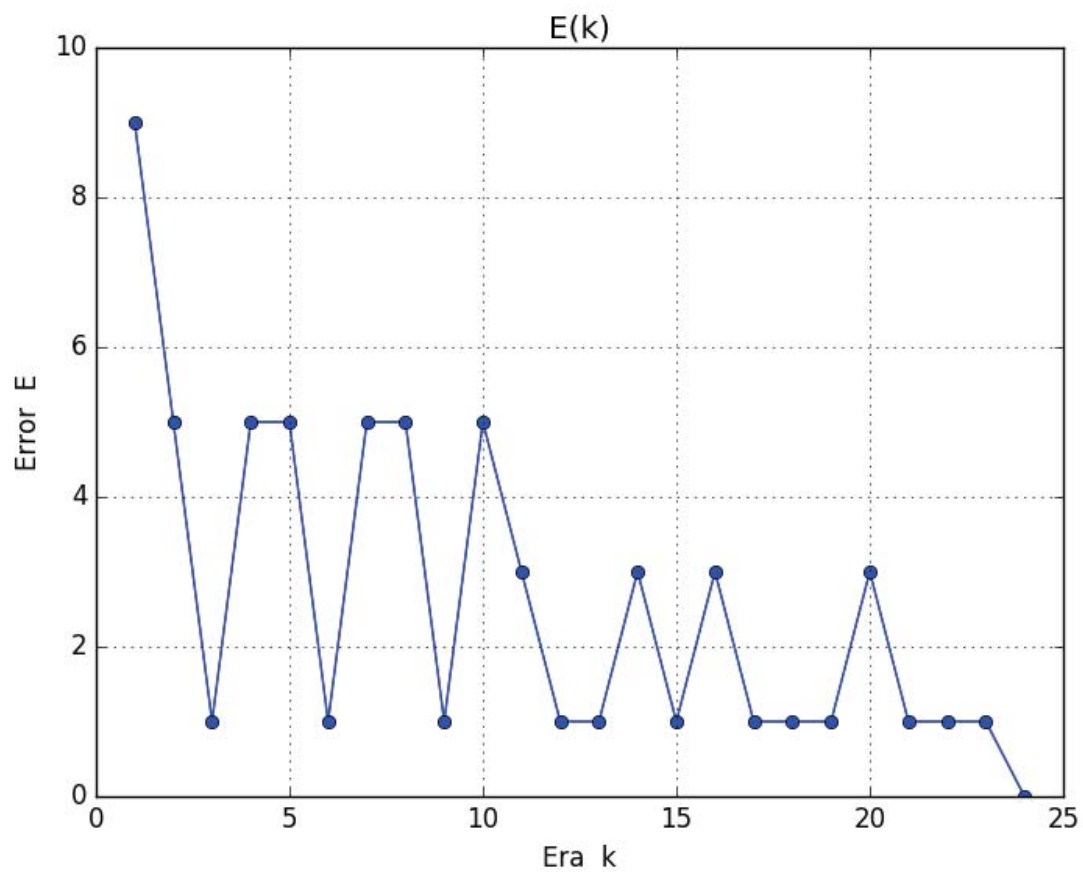


Рисунок 1 График суммарной ошибки НС по эпохам обучения (пороговая ΦA)

1.2. Используя сигмоидальную (логистическую) ФА:

$$f(net) = \frac{1}{1 + \exp(-net)},$$

производная которой

$$\frac{df(net)}{dnet} = f(net)(1 - f(net)).$$

Таблица 2 Параметры НС на последовательных эпохах (логистическая ФА)

Номер эпохи, k	Вектор весов W , выходной вектор Y , суммарная ошибка E
0	$Y = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.0, 0.0, 0.0, 0.0, 0.0), E = 9$
1	$Y = (1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.0, -0.075, 0.0, 0.0, -0.075), E = 5$
2	$Y = (1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.0, -0.15, 0.0, 0.0, -0.075), E = 5$
3	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.075, -0.15, 0.0, 0.0, -0.074), E = 1$
4	$Y = (1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.075, -0.15, 0.0, 0.0, -0.149), E = 5$
5	$Y = (1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.075, -0.225, -0.0, 0.0, -0.149), E = 5$
6	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.15, -0.225, -0.0, 0.0, -0.149), E = 1$
7	$Y = (1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.15, -0.225, 0.0, -0.075, -0.149), E = 3$
8	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0),$ $W = (0.149, -0.3, -0.0, -0.075, -0.074), E = 1$
9	$Y = (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0),$ $W = (0.224, -0.225, 0.074, -0.075, -0.074), E = 0$

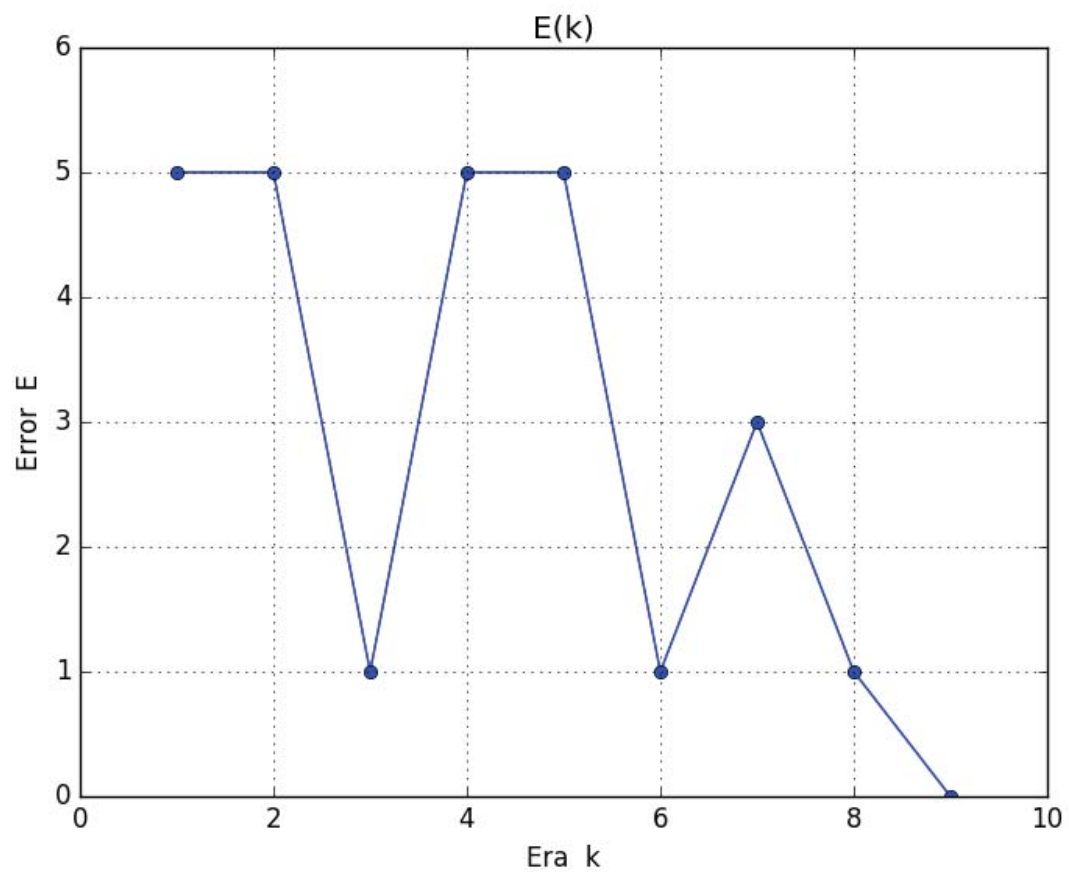


Рисунок 2 График суммарной ошибки НС по эпохам обучения (логистическая ФА)

2. Обучение НС с использованием части комбинаций переменных x_1, x_2, x_3, x_4 .

Последовательно уменьшая выборку количества векторов, найдём наименьшее количество необходимых для обучения векторов.

2.1. Используя пороговую ФА.

Минимальный набор из четырёх векторов:

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

Даёт следующие синаптические коэффициенты:

$$W = (0.9, -0.9, 0.3, -0.3, -0.3)$$

Для полного обучения потребовалось 4 эпохи.

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

Номер эпохи, k	Вектор весов W , выходной вектор Y , суммарная ошибка E
0	$Y = (0, 0, 0, 0),$ $W = (0.0, 0.0, 0.0, 0.0, 0.0), E = 3$
1	$Y = (0, 0, 0, 0),$ $W = (0.0, -0.3, -0.3, -0.3, 0.0), E = 3$
2	$Y = (1, 0, 0, 0),$ $W = (0.3, -0.6, -0.3, -0.6, 0.0), E = 2$
3	$Y = (1, 1, 0, 0),$ $W = (0.6, -0.9, 0.0, -0.6, -0.3), E = 1$
4	$Y = (1, 1, 1, 0),$ $W = (0.9, -0.9, 0.3, -0.3, -0.3), E = 0$

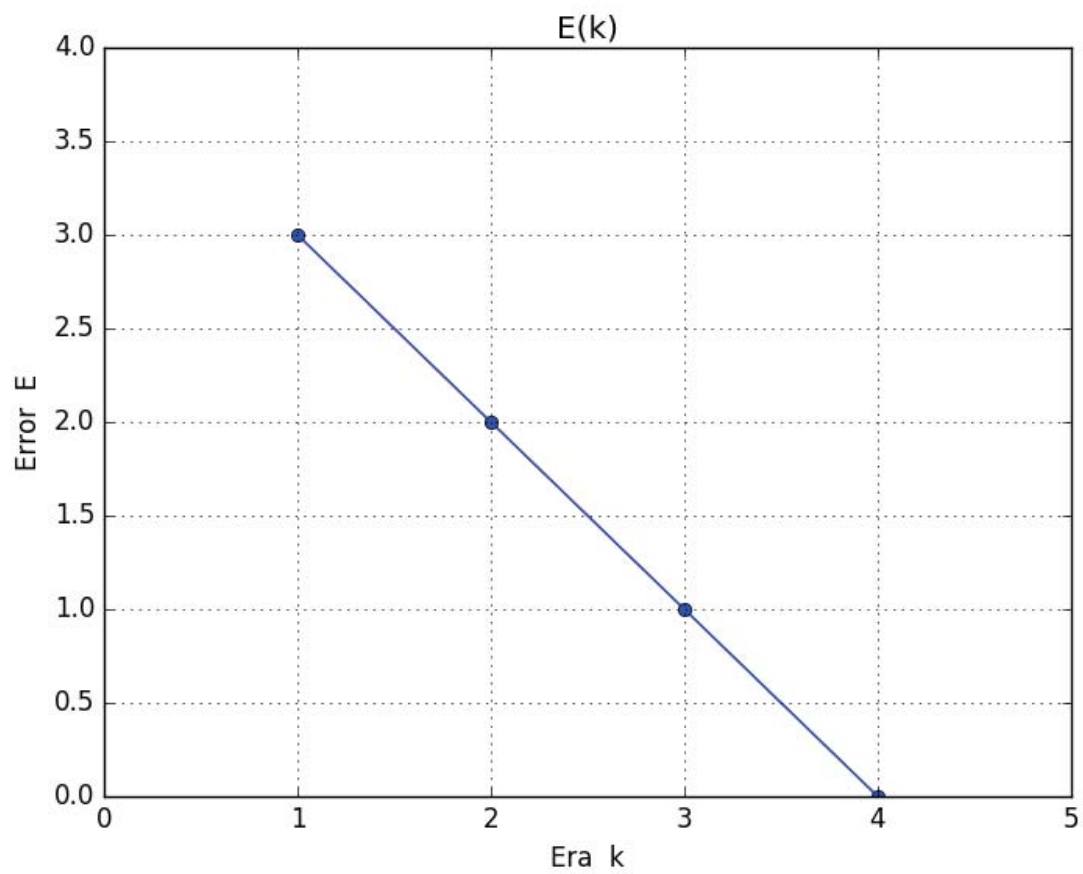


Рисунок 3 График суммарной ошибки НС по эпохам обучения (пороговая ФА) при наборе из 4-х векторов

2.2. Используя сигмоидальную (логистическую) ФА.

Минимальный набор из четырёх векторов:

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

Даёт следующие синаптические коэффициенты:

$$W = (0.225, -0.298, 0.077, -0.073, -0.073)$$

Для полного обучения потребовалось 10 эпох.

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

Номер эпохи, k	Вектор весов W , выходной вектор Y , суммарная ошибка E
0	$Y = (0, 0, 0, 0),$ $W = (0.0, 0.0, 0.0, 0.0, 0.0), E = 2$
1	$Y = (1, 0, 0, 0),$ $W = (0.002, -0.073, 0.002, 0.002, 0.002), E = 1$
2	$Y = (0, 0, 0, 0),$ $W = (0.002, -0.073, 0.002, -0.073, -0.073), E = 2$
3	$Y = (0, 0, 0, 0),$ $W = (0.001, -0.148, 0.002, -0.073, -0.073), E = 2$
4	$Y = (0, 0, 0, 0),$ $W = (0.076, -0.148, 0.002, -0.073, -0.073), E = 2$
5	$Y = (0, 0, 0, 0),$ $W = (0.076, -0.223, 0.002, -0.073, -0.073), E = 2$
6	$Y = (1, 0, 0, 0),$ $W = (0.151, -0.223, 0.002, -0.073, -0.073), E = 1$
7	$Y = (0, 0, 0, 0),$ $W = (0.151, -0.223, 0.002, -0.148, -0.148), E = 2$
8	$Y = (1, 1, 1, 0),$ $W = (0.225, -0.223, 0.077, -0.073, -0.073), E = 1$
9	$Y = (0, 0, 0, 0),$ $W = (0.15, -0.298, 0.077, -0.148, -0.148), E = 2$
10	$Y = (1, 0, 1, 0),$ $W = (0.225, -0.298, 0.077, -0.073, -0.073), E = 0$

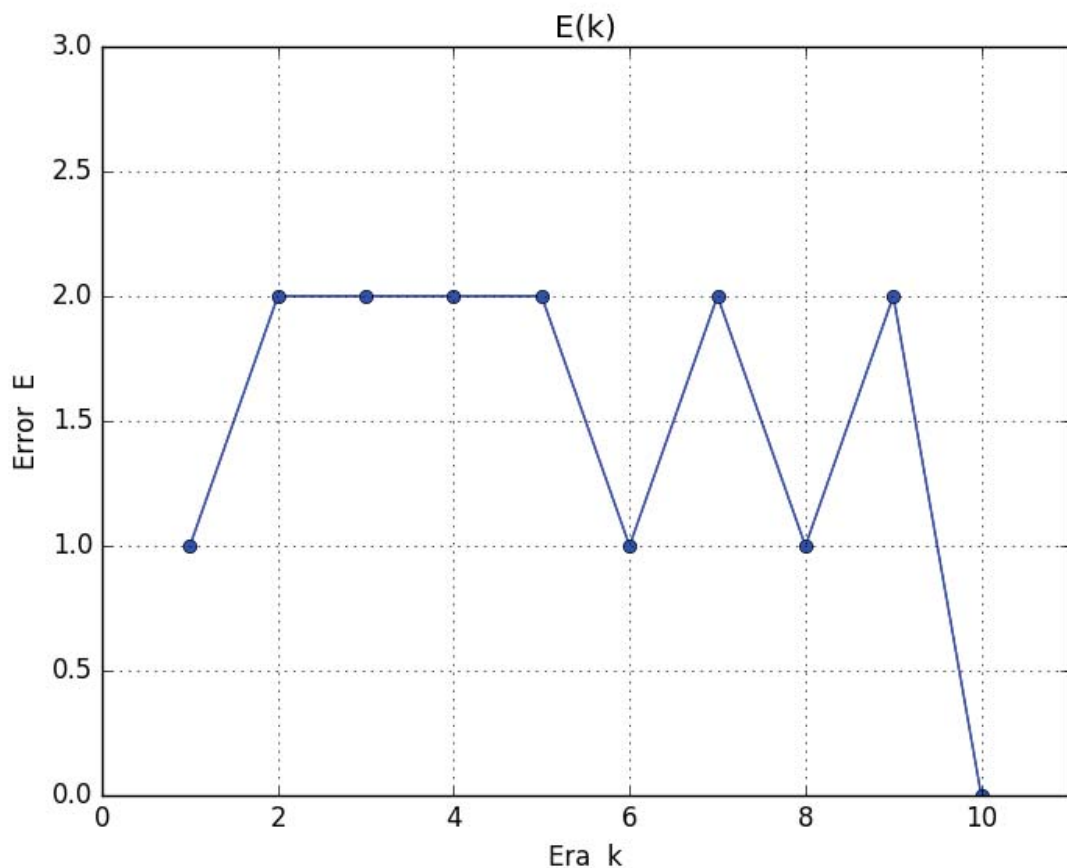


Рисунок 4 График суммарной ошибки НС по эпохам обучения (логистическая ФА) при наборе из 4-х векторов

Выводы

В ходе проделанной работы была изучена работа простейшего нейрона НС на базе нейрона с нелинейной функцией активации и проведено обучение по правилу Видроу-Хоффа. В качестве функций активации брались две различные функции – пороговая и логистическая. В ходе обучения на полных наборах было выявлено, что с использованием логистической функции активации понадобилось меньше эпох, чем для обучения с использованием пороговой функции активации.

Кроме того, для случаев пороговой и логистической функций активации были найдены минимально возможные наборы векторов, на которых можно обучить НС. В обоих случаях удалось найти наборы, состоящие из четырёх векторов. В случае обучения с использованием пороговой функции активации понадобилось меньшее количество эпох, чем с использованием логистической.

Приложение А.

Файл 'lab-1.py'

...

Лабораторная работа No 1

Исследование однослойных нейронных сетей на примере моделирования булевых выражений.

Цель: Исследовать функционирование простейшей нейронной сети (НС) на базе нейрона с нелинейной функцией активации и ее обучение по правилу Видроу-Хоффа.

Вариант 23.

...

```
import sys
```

```
from AF import *
```

```
from Education import *
```

```
from Tools import boolean_function, bin_generation, IntToByte
```

```
def initialize_components():
```

```
    ...
```

```
    Функция инициализирует необходимые для расчётов компоненты
```

```
    :param return: F - значения БФ, W - начальные весовые коэффициенты
```

```
    ...
```

```
    W = [0, 0, 0, 0, 0]
```

```
    n = 4 # число переменных
```

```
    X = bin_generation(n)
```

```
    F = get_F(X)
```

```
    return F, W
```

```
def get_F(X):
```

```
    ...
```

```
    Функция возвращает значения БФ на заданных ей наборах переменных
```

```
    :param X: наборы переменных значения БФ
```

```
    :param return: значения БФ
```

```
    ...
```

```
    F = list()
```

```
    for x in X:
```

```
        # x0 в расчёт не берётся. Оно необходимо лишь для правила Видроу-Хоффа
```

```
        F.append(boolean_function(x[1], x[2], x[3], x[4]))
```

```
    return F
```

```
def nnm_BF(W, F, outputFile):
```

```
    ...
```

```
    Функция производит расчёт и построения нейросетевой модели БФ, используя пороговую и логистическую ФА
```

```
    :param W: начальные весовые коэффициенты
```

```
    :param F: значения БФ
```

```
    :param outputFile: имя файла вывода
```

```
    :param return: none
```

```
    ...
```

```
    # Получим нейросетевую модель БФ, используя пороговую ФА
```