**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ(МИИТ)**)

**Институт управления и цифровых технологий**

Кафедра «Вычислительные системы, сети и информационная безопасность»

**Отчет по Лабораторной работе №1**

**«Обучение искусственного нейрона»**

**По дисциплине «Нейроинформатика»**

*Направление:* 10.03.01*Информационная безопасность*

*Профиль:**Безопасность компьютерных систем*

Выполнил:   
студент группы УИБ-311

Москаленко Виталий Александрович

Проверил:

Доцент Малинский С.В.

(должность, ФИО)

**МОСКВА 2024**

**СОДЕРЖАНИЕ**

[1 ЗАДАНИЕ 3](#_Toc177515883)

[2 ОСНОВНЫЕ ПОНЯТИЯ ОБ ИСКУССТВЕННОМ НЕЙРОНЕ 6](#_Toc177515884)

[3 ФОРМИРОВАНИЕ ОБУЧАЮЩЕЙ ВЫБОРКИ 8](#_Toc177515885)

[4 ОБУЧЕНИЕ ИСКУССТВЕННОГО НЕЙРОНА ПРИ ЗАДАННОМ КОЛИЧЕСТВЕ ПОМЕХ 9](#_Toc177515886)

[5 РЕЗУЛЬТАТ ОБУЧЕНИЯ ИСКУССТВЕННОГО НЕЙРОНА 10](#_Toc177515888)

[ЗАКЛЮЧЕНИЕ 12](#_Toc177515889)

[ПРИЛОЖЕНИЕ 13](#_Toc177515890)

1 ЗАДАНИЕ

Необходимо разработать программу, которая реализует простой искусственный нейрон для распознавания цифры "4" на основе её графического представления, состоящего из элементов сетки вида 3 на 5.

На рисунке 1 представлена сетка 3 на 5, по которой каждая цифра изображается набором сегментов этой сетки,

Например:

- Признак "1" обозначает наличие сегмента в соответствующем ячейке.

- Признак "0" обозначает её отсутствие.

Изображение выглядит как снимок экрана, число, Шрифт, линия

Автоматически созданное описание

Рисунок 1 – Шаблон графического изображения цифр

В таблице 1 представлен полный набор цифр, каждая из которых соответствует своему набору признаков. Эти данные используются для формирования обучающей выборки, на которой нейрон будет обучаться распознавать цифры.

Таблица 1 – Цифры и их представления

|  |  |  |  |
| --- | --- | --- | --- |
| Цифра | Графическое представление | Двоичное представление | |
| 0 | Изображение выглядит как снимок экрана, прямоугольный, Прямоугольник, дизайн  Автоматически созданное описание | 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1 | |
| 1 | Изображение выглядит как снимок экрана, прямоугольный, линия, дизайн  Автоматически созданное описание | 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1 | |
| 2 | Изображение выглядит как снимок экрана, линия, дизайн, типография  Автоматически созданное описание | | 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1 |
| 3 | Изображение выглядит как снимок экрана, линия, дизайн  Автоматически созданное описание | | 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1 |
| 4 | Изображение выглядит как снимок экрана, линия, прямоугольный, дизайн  Автоматически созданное описание | | 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1 |
| 5 | Изображение выглядит как снимок экрана, линия, дизайн  Автоматически созданное описание | | 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1 |
| 6 | Изображение выглядит как снимок экрана, линия, дизайн  Автоматически созданное описание | | 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1 |

Продолжение таблицы 1:

|  |  |  |
| --- | --- | --- |
| 7 | Изображение выглядит как снимок экрана, линия, дизайн  Автоматически созданное описание | 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0 |
| 8 | Изображение выглядит как снимок экрана, прямоугольный, линия, Красочность  Автоматически созданное описание | 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1 |
| 9 | Изображение выглядит как снимок экрана, линия, прямоугольный, дизайн  Автоматически созданное описание | 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1 |

На основе данного шаблона для каждой цифры были сформированы обучающие данные, которые содержат бинарный вектор признаков (массив из 15 значений). Этот вектор однозначно описывает графическое представление каждой цифры,

Согласно заданию, искусственный нейрон обучался распознавать цифру "4", ее бинарный вектор равен: 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1.

Пороговое значение Q в данной работе – 35.

2 ОСНОВНЫЕ ПОНЯТИЯ ОБ ИСКУССТВЕННОМ НЕЙРОНЕ

Искусственный нейрон — это базовый элемент нейронных сетей, основанный на принципах работы биологических нейронов мозга. В искусственных сетях нейроны обрабатывают информацию и принимают решения, основываясь на поступающих сигналах. В задаче распознавания цифр нейрон обрабатывает входные данные и классифицирует их в соответствии с установленными правилами.

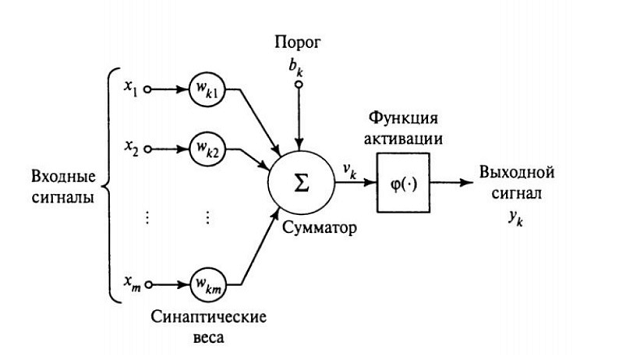


Рисунок 2 – Искусственный нейрон

Основные элементы искусственного нейрона:

1. Входные данные (признаки):

Нейрон получает набор числовых значений, которые представляют собой числовые значения (это сегменты сетки 3 на 5). В данной задаче каждая цифра от 0 до 9 представлена 15 признаками. Эти признаки поступают на вход нейрона.

2. Веса (коэффициенты):

Каждый входной признак умножается на свой вес. Вес определяет важность признака для принятия решения. Например, для распознавания цифры "4" более важным признакам присваиваются более высокие веса. Веса помогают нейрону различать ключевые и второстепенные признаки.

3. Функция активации:

После того как входные данные умножены на веса, нейрон суммирует результаты. Полученная сумма проходит через функцию активации, которая решает, активируется нейрон или нет. Простейший пример такой функции —функция с пороговым значением: если сумма превышает определённое значение, заданное ранее, нейрон активируется и выдаёт положительный результат (распознана цифра "4"), в противном случае — отрицательный результат.

4. Выход (результат):

На основе работы функции активации нейрон генерирует выходной сигнал. Этот сигнал выдается пользователю.

3 ФОРМИРОВАНИЕ ОБУЧАЮЩЕЙ ВЫБОРКИ

Для успешного обучения нейрона, способного распознавать цифры по их бинарному представлению, необходимо сформировать обучающую выборку. В нашем случае цифры представлены в виде набора признаков. Эти признаки в виде бинарных значений описывают наличие или отсутствие сегментов сетки 3 на 5 в изображении цифры.

Используя данный набор данных, нейрон сможет по входным признакам корректно определять, какая цифра перед ним. Обучающая выборка создаётся путём генерации множества примеров этих векторов и передачи их в алгоритм обучения нейрона.

Этот процесс позволяет обучить систему эффективно распознавать цифру "4", основываясь на уникальных признаках каждой цифры.

4 ОБУЧЕНИЕ ИСКУССТВЕННОГО НЕЙРОНА ПРИ ЗАДАННОМ КОЛИЧЕСТВЕ ПОМЕХ

После того, как обучающая выборка была сформирована и в нее было добавлено определенное количество ошибок, запускается алгоритм обучения на распознавание цифры.

Эпохой называется полное прохождение алгоритма по обучающей выборке. Для корректного распознавания программе может потребоваться одна или более эпоха.

На каждой эпохе вычисляется сумма весов для каждого элемента обучающей выборки. В зависимости от того, верно ли классифицируется элемент, то есть является ли объект цифрой "4", происходит корректировка весов:

1. Если объект является цифрой "4", но сумма меньше или равна пороговому значению, веса увеличиваются.
2. Если объект не является цифрой "4", но взвешенная сумма больше порога, веса уменьшаются.

В конце каждой эпохи выводится точность модели, то есть процент правильно классифицированных объектов.

5 РЕЗУЛЬТАТ ОБУЧЕНИЯ ИСКУССТВЕННОГО НЕЙРОНА

В результате выполнения программы были зафиксированы итерации по обучению искусственного нейрона для распознавания цифры "4" на основе её признаков. Основной целью обучения было увеличение весов для признаков, связанных с цифрой "4", если их сумма была меньше или равна порогу Q, и уменьшение весов для других цифр, если их сумма превышала этот порог.

Эпохой называется полное прохождение алгоритма по обучающей выборке. Для корректного распознавания программе может потребоваться одна или более эпоха.

Программа реализует обучение распознавания цифры "4" на трех эпохах. Результаты весов после прохождения каждой из трех эпох, а также начальное и конечное значение представлены на рисунке 3:

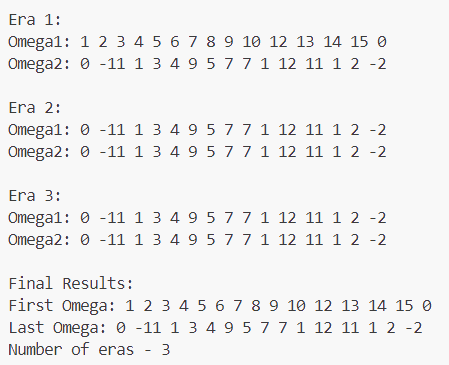


Рисунок 3 – Результат работы программы

Далее была выполнена проверка для каждой цифры, чтобы убедиться в правильности обучения искусственного нейрона на распознавание цифры "4".

Ниже представлен суммы весов для всех цифр, а также их сравнение с пороговым значением Q:

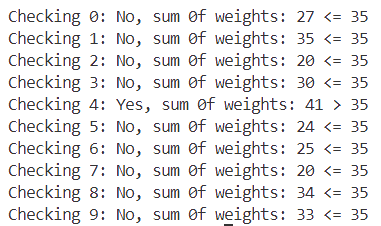


Рисунок 4 – Результаты проверки всех цифр

Как видно из рисунка, только у цифры "4" значение суммы весов превышает заданное пороговое значение Q, следовательно, искусственный нейрон успешно обучился распознаванию заданной цифры.

ЗАКЛЮЧЕНИЕ

По завершении обучения искусственный нейрон успешно обучился распознавать цифру "4" с пороговым значением 35. Распознавание происходило путём корректировки весов в процессе обучения, что продемонстрировало успешную работу алгоритма.

Этот результат подтверждает, что искусственный нейрон, обучаемый на принципах простых механизмов корректировки весов, может быть обучен распознавать необходимые пользователю символы, такие как цифра "4", по заданным признакам.

ПРИЛОЖЕНИЕ

#include <iostream>

#include <algorithm>

#include <ctime>

#include <cstring>

using namespace std;

int main()

{

    int Q=35;

    int Omega[15]={1,2,3,4,5,6,7,8,9,10,12,13,14,15,0};

    int OmegaFirst[15]={1,2,3,4,5,6,7,8,9,10,12,13,14,15,0};

    int ves[300][15];

    int mycifra[15]={1,0,1,1,0,1,1,1,1,0,0,1,0,0,1}; //Моя цифра 4 - по заданию

    int check[15];

    int key;

    int maxeras=3;

    int sumCheck=0;

    int cifri[10][15]=

    {

        {1,1,1,1,0,1,1,0,1,1,0,1,1,1,1}, //0

        {0,0,1,0,1,1,1,0,1,0,0,1,0,0,1}, //1

        {1,1,1,0,0,1,1,1,1,1,0,0,1,1,1}, //2

        {1,1,1,0,0,1,1,1,1,0,0,1,1,1,1}, //3

        {1,0,1,1,0,1,1,1,1,0,0,1,0,0,1}, //4

        {1,1,1,1,0,0,1,1,1,0,0,1,1,1,1}, //5

        {1,1,1,1,0,0,1,1,1,1,0,1,1,1,1}, //6

        {1,1,1,0,0,1,0,1,0,0,1,0,0,1,0}, //7

        {1,1,1,1,0,1,1,1,1,1,0,1,1,1,1}, //8

        {1,1,1,1,0,1,1,1,1,0,0,1,1,1,1}  //9

    };

    int randommassiv[300]=

    {

        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

        1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,

        2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,

        3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,

        4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,

        5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,

        6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,

        7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,

        8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,

        9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9

    };

    int viborka[300][15];

    int kolvo=0;

    random\_shuffle(randommassiv, randommassiv + 300);

    for (int i=0;i<300;i++)

    {

        for (int j=0;j<15;j++)

        {

            viborka[i][j]=cifri[randommassiv[i]][j];

        }

    }

    cout << "Author - Moskalenko Vitaly Alexandrovich UIB-311";

    cout << "Training sample: " << endl;

    for (int i=0;i<300;i++)

    {

        cout << i+1 << " | ";

        for (int j=0;j<15;j++)

        {

            cout << viborka[i][j] << ' ';

        }

        cout << "|";

        cout << endl;

    }

    while (kolvo<maxeras)

    {

        kolvo++;

        cout << endl;

        cout << "Era " << kolvo << ":";

        cout << endl;

        cout << "Omega1: ";

        for (int i=0;i<15;i++)

        {

            cout << Omega[i] << ' ';

        }

        cout << endl;

        for (int i=0;i<300;i++)

        {

            int sum=0;

            for (int j=0;j<15;j++)

            {

                ves[i][j]=Omega[j]\*viborka[i][j];

                sum+=ves[i][j];

            }

            key=0;

            for (int j=0;j<15;j++)

            {

                if (viborka[i][j]==mycifra[j])

                {

                    key++;

                }

            }

            if ((key==15)&&(sum<=Q))

            {

                for (int x=0;x<15;x++)

                {

                    Omega[x]+=viborka[i][x];

                }

            }

            if ((key!=15)&&(sum>Q))

            {

                for (int x=0;x<15;x++)

                {

                    Omega[x]-=viborka[i][x];

                }

            }

        }

        cout << "Omega2: ";

        for (int i=0;i<15;i++)

        {

            cout << Omega[i] << ' ';

        }

        cout << endl;

    }

    cout << endl;

    cout << "Final Results: " << endl;

    cout << "First Omega: ";

    for (int i=0;i<15;i++)

    {

        cout << OmegaFirst[i] << ' ';

    }

    cout << endl;

    cout << "Last Omega: ";

    for (int i=0;i<15;i++)

    {

        cout << Omega[i] << ' ';

    }

    cout << endl;

    cout << "Number of eras - " << kolvo;

    cout << endl;

    cout << endl;

    memset(check,0,sizeof(check));

    sumCheck=0;

    int cifraCheck0[15]={1,1,1,1,0,1,1,0,1,1,0,1,1,1,1}; //0

    int cifraCheck1[15]={0,0,1,0,1,1,1,0,1,0,0,1,0,0,1}; //1

    int cifraCheck2[15]={1,1,1,0,0,1,1,1,1,1,0,0,1,1,1}; //2

    int cifraCheck3[15]={1,1,1,0,0,1,1,1,1,0,0,1,1,1,1}; //3

    int cifraCheck5[15]={1,1,1,1,0,0,1,1,1,0,0,1,1,1,1}; //5

    int cifraCheck6[15]={1,1,1,1,0,0,1,1,1,1,0,1,1,1,1}; //6

    int cifraCheck7[15]={1,1,1,0,0,1,0,1,0,0,1,0,0,1,0}; //7

    int cifraCheck8[15]={1,1,1,1,0,1,1,1,1,1,0,1,1,1,1}; //8

    int cifraCheck9[15]={1,1,1,1,0,1,1,1,1,0,0,1,1,1,1}; //9

    for (int i=0;i<15;i++)

    {

        check[i]=cifraCheck0[i]\*Omega[i];

        sumCheck+=check[i];

    }

    cout << "Checking 0: ";

    if (sumCheck>Q)

    {

        cout << "Yes" << ", " << "sum 0f weights: " << sumCheck << " > " << Q << endl;

    }

    else

    {

        cout << "No" << ", " << "sum 0f weights: " << sumCheck << " <= " << Q << endl;

    }

    memset(check,0,sizeof(check));

    sumCheck = 0;

    for (int i=0;i<15;i++)

    {

        check[i]=cifraCheck1[i]\*Omega[i];

        sumCheck+=check[i];

    }

    cout << "Checking 1: ";

    if (sumCheck > Q)

    {

        cout << "Yes" << ", " << "sum 0f weights: " << sumCheck << " > " << Q << endl;

    }

    else

    {

        cout << "No" << ", " << "sum 0f weights: " << sumCheck << " <= " << Q << endl;

    }

    memset(check,0,sizeof(check));

    sumCheck=0;

    for (int i=0;i<15;i++)

    {

        check[i]=cifraCheck2[i]\*Omega[i];

        sumCheck+=check[i];

    }

    cout << "Checking 2: ";

    if (sumCheck>Q)

    {

        cout << "Yes" << ", " << "sum 0f weights: " << sumCheck << " > " << Q << endl;

    }

    else

    {

        cout << "No" << ", " << "sum 0f weights: " << sumCheck << " <= " << Q << endl;

    }

    memset(check,0,sizeof(check));

    sumCheck=0;

    for (int i=0;i<15;i++)

    {

        check[i]=cifraCheck3[i]\*Omega[i];

        sumCheck+=check[i];

    }

    cout << "Checking 3: ";

    if (sumCheck>Q)

    {

        cout << "Yes" << ", " << "sum 0f weights: " << sumCheck << " > " << Q << endl;

    }

    else

    {

        cout << "No" << ", " << "sum 0f weights: " << sumCheck << " <= " << Q << endl;

    }

    memset(check,0,sizeof(check));

    sumCheck=0;

    for (int i=0;i<15;i++)

    {

        check[i]=mycifra[i]\*Omega[i];

        sumCheck+=check[i];

    }

    cout << "Checking 4: ";

    if (sumCheck>Q)

    {

        cout << "Yes" << ", " << "sum 0f weights: " << sumCheck << " > " << Q << endl;

    }

    else

    {

        cout << "No" << ", " << "sum 0f weights: " << sumCheck << " <= " << Q << endl;

    }

    memset(check,0,sizeof(check));

    sumCheck=0;

    for (int i=0;i<15;i++)

    {

        check[i]=cifraCheck5[i]\*Omega[i];

        sumCheck+=check[i];

    }

    cout << "Checking 5: ";

    if (sumCheck>Q)

    {

        cout << "Yes" << ", " << "sum 0f weights: " << sumCheck << " > " << Q << endl;

    }

    else

    {

        cout << "No" << ", " << "sum 0f weights: " << sumCheck << " <= " << Q << endl;

    }

    memset(check,0,sizeof(check));

    sumCheck=0;

    for (int i=0;i<15;i++)

    {

        check[i]=cifraCheck6[i]\*Omega[i];

        sumCheck+=check[i];

    }

    cout << "Checking 6: ";

    if (sumCheck>Q)

    {

        cout << "Yes" << ", " << "sum 0f weights: " << sumCheck << " > " << Q << endl;

    }

    else

    {

        cout << "No" << ", " << "sum 0f weights: " << sumCheck << " <= " << Q << endl;

    }

    memset(check,0,sizeof(check));

    sumCheck=0;

    for (int i=0;i<15;i++)

    {

        check[i]=cifraCheck7[i]\*Omega[i];

        sumCheck+=check[i];

    }

    cout << "Checking 7: ";

    if (sumCheck>Q)

    {

        cout << "Yes" << ", " << "sum 0f weights: " << sumCheck << " > " << Q << endl;

    }

    else

    {

        cout << "No" << ", " << "sum 0f weights: " << sumCheck << " <= " << Q << endl;

    }

    memset(check,0,sizeof(check));

    sumCheck=0;

    for (int i=0;i<15;i++)

    {

        check[i]=cifraCheck8[i]\*Omega[i];

        sumCheck+=check[i];

    }

    cout << "Checking 8: ";

    if (sumCheck>Q)

    {

        cout << "Yes" << ", " << "sum 0f weights: " << sumCheck << " > " << Q << endl;

    }

    else

    {

        cout << "No" << ", " << "sum 0f weights: " << sumCheck << " <= " << Q << endl;

    }

    memset(check,0,sizeof(check));

    sumCheck=0;

    for (int i=0;i<15;i++)

    {

        check[i]=cifraCheck9[i]\*Omega[i];

        sumCheck+=check[i];

    }

    cout << "Checking 9: ";

    if (sumCheck>Q)

    {

        cout << "Yes" << ", " << "sum 0f weights: " << sumCheck << " > " << Q << endl;

    }

    else

    {

        cout << "No" << ", " << "sum 0f weights: " << sumCheck << " <= " << Q << endl;

    }

    return 0;

}