МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Брестский государственный технический университет»

Кафедра «Интеллектуальные информационные технологии»

Лабораторная работа №2

По дисциплине «Интеллектуальные методы обработки данных»

За 7 семестр

Тема: «Линейная искусственная нейронная сеть.   
Правило обучения Видроу-Хоффа»

Выполнила:  
студентка 4 курса  
группы АС-56  
Карпенко М.В.

Проверил:

Савицкий Ю.В.

Брест 2022

*Цель работы:* Изучить процессы обучения и функционирования линейной искусственной нейронной сети (ИНС) при решении задач прогнозирования.

*Задание:*   
1. Изучить теоретические сведения

2. Написать на любом языке высокого уровня программу моделирования прогнозирующей линейной ИНС. Для тестирования использовать функцию. Обучение и прогнозирование производить на 30 и 15 значениях соответственно, табулируя функцию с шагом 0.1. Скорость обучения выбирается студентом самостоятельно, для чего моделирование проводится несколько раз для разных значений *a*. Результаты оцениваются по двум критериям – 1) скорости обучения и 2) минимальной достигнутой среднеквадратичной ошибке. Необходимо заметить, что эти критерии в общем случае являются взаимоисключающими, и оптимальные значения для каждого критерия достигаются при разных значениях *a*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № варианта | *a* | *b* | *d* | Кол-во входов ИНС |
| 9 | 1 | 8 | 0.3 | 5 |

*Текст программы:*

#include <iostream>

#include <math.h>

#include <time.h>

using namespace std;

int main()

{

clock\_t start = clock();

int num\_input = 5, // количество входов

sample = 30, // размер выборки для обучения

predictive\_sample = 15, // размер прогнозируемой выборки

k = 0; // счетчик итераций

float w[5], // весовые коэффициенты

T, // пороговое значение

E, E\_min, // ошибка обучения и мин.ошибка

t[60], // массив эталонных значений

y[60], // массив выходных значений сети

a, Y; // скорость обучения

for (int i = 0; i < num\_input; i++)

w[i] = (1 + rand() % 9) / (float)10; // инициализация весовых коэффициентов

T = (1 + rand() % 9) / (float)10; // инициализация порогового значения

float tmp = 0.0;

for (int j = 0; j < 60; j++) // заполнение массива выборки, на которой проходит обучение, т.е. эталонными значениями

t[j] = 1 \* sin(8 \* (j + 1)) + 0.3;

cout << "Enter a: ";

cin >> a;

cout << "Enter Emin: ";

cin >> E\_min;

cout.width(16); cout << "t ";

cout.width(16); cout << "Y ";

cout.width(19); cout << "E " << endl;

//обучение

do

{

E = 0;

k++;

cout << "it = " << k << endl;

for (int i = 0; i < (sample - num\_input); i++)

{

Y = 0;

for (int j = 0; j < num\_input; j++)

Y += w[j] \* t[i + j];

Y -= T; // рассчитываем выходную активность

for (int j = 0; j < num\_input; j++)

w[j] -= a \* (Y - t[i + num\_input]) \* t[i + j]; // изменяем весовые коэффициенты

T += a \* (Y - t[i + num\_input]); // меняем пороговое значение

}

for (int i = 0; i < (sample - num\_input); i++)

{

Y = 0;

for (int j = 0; j < num\_input; j++)

Y += w[j] \* t[i + j];

Y -= T; // расчитываем выходную активность

E += 0.5 \* pow((Y - t[i + num\_input]), 2); // находим суммарную среднеквадратичную ошибку

cout.width(15); cout << t[i + num\_input];

cout.width(15); cout << Y;

cout.width(18); cout << E << endl;

}

} while (E >= E\_min);

// прогнозирование

cout << "\n\t\t\tPREDICTION\n" << endl;

cout.width(16); cout << "t ";

cout.width(16); cout << "Y ";

cout.width(19); cout << "E " << endl;

for (int i = (sample - num\_input); i < (sample + predictive\_sample); i++)

{

Y = 0;

for (int j = 0; j < num\_input; j++)

Y += w[j] \* t[i + j];

Y -= T;

E = 0.5 \* pow((Y - t[i + num\_input]), 2);

cout.width(15); cout << t[i + num\_input];

cout.width(15); cout << Y;

cout.width(18); cout << E << endl;

t[i + num\_input] = Y;

}

clock\_t finish = clock();

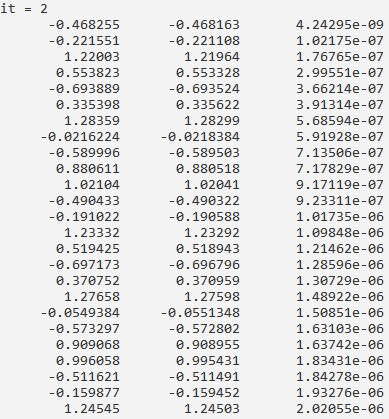
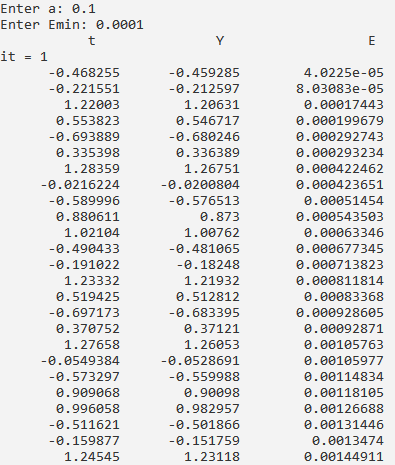
cout << "Time: " << (finish - start) << " ms\n";

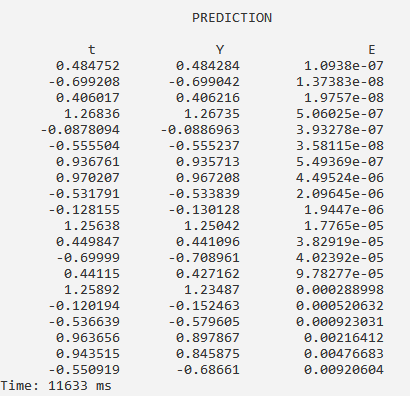
system("pause");

return 0;

}

*Результаты работы программы:*

**

**

При E=0.0001 получили следующие значение с различными альфа.

|  |  |  |  |
| --- | --- | --- | --- |
| alpha | Error | Time, ms | Количество итераций |
| 0,02 | 3.08e-5 | 8836 | 9 |
| 0.1 | 3,06e-6 | 11633 | 2 |
| 0,4 | 6.12e-5 | 9024 | 2 |

Отобразим значения для альфа, равного 0.02, т.к. скорость обучения при данном шаге обучения минимальна, ошибка достаточно мала.

график изменения среднеквадратичной ошибки в зависимости от итерации

график, содержащий кривую прогнозирования и кривую эталонного временного ряда

*Выводы по лабораторной работы:* Изучила прогнозирование и

функционирование линейной ИНС при решении задачи прогнозирования, определила оптимальную альфа, при которой обучение происходит с максимальной скоростью при достаточно малой ошибке, спрогнозировала значения и убедилась, что нейронная сеть работает правильно.