Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №2**

По дисциплине: «Аппаратное обеспечение интеллектуальных систем»

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

**Выполнил:**

Студент 2 курса

Группы ИИ-23

Макаревич Н. Р.

**Проверил:**

Михно Е.В.

Брест 2024

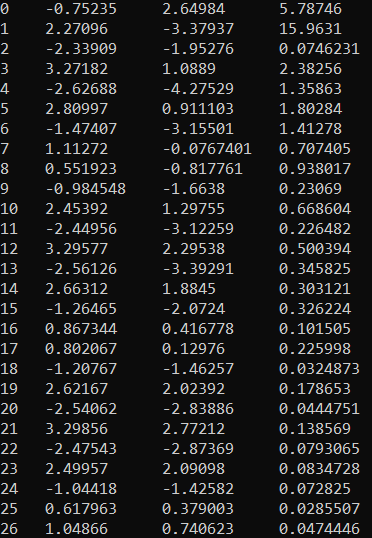
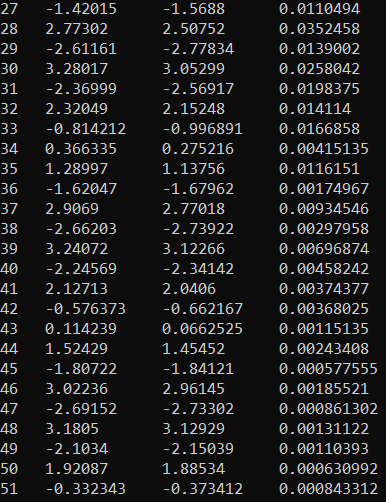
**Цель работы:** изучить обучение и функционирование линейной ИНС при решении задач прогнозирования.

**Задание:** написать на любом ЯВУ программу моделирования прогнозирующей линейной ИНС. Для тестирования использовать функцию ** .

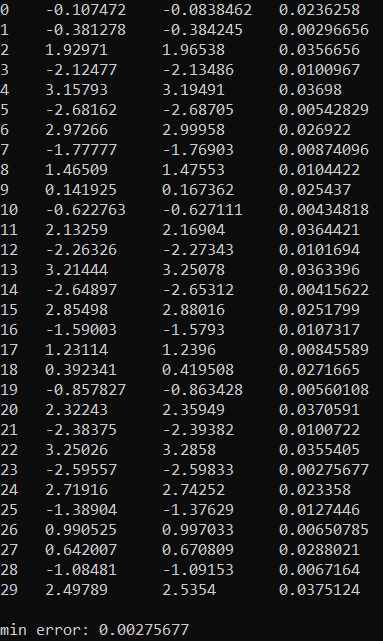
Варианты заданий приведены в следующей таблице:

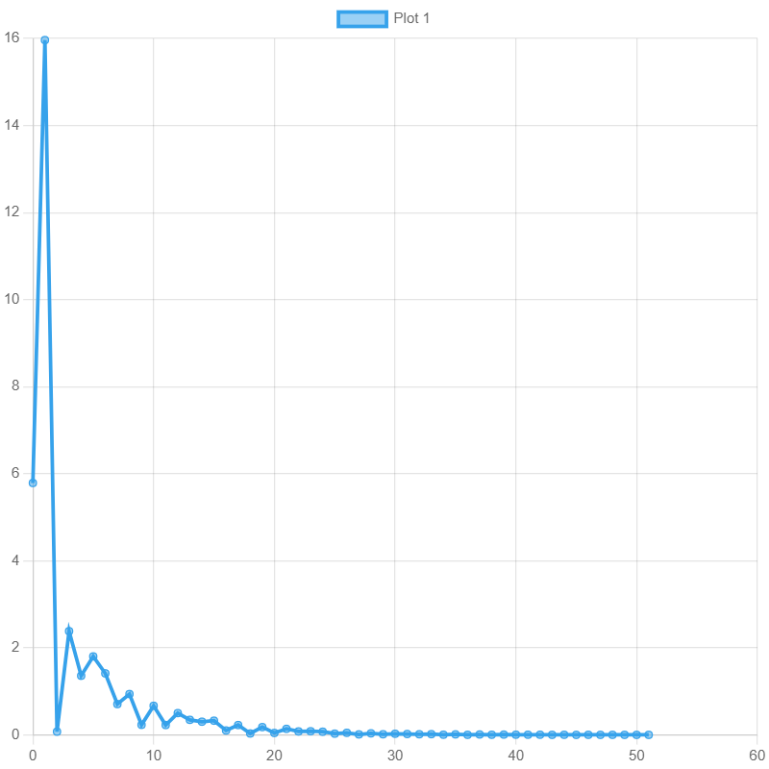
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № варианта | a | b | d | Кол-во входов ИНС |
| 3 | 3 | 7 | 0.3 | 5 |

**Обучение**:

**Прогнозирование:**





double ADALINE::function(double x) {

return 3 \* std::sin(x \* 7) + 0.3;

}

void ADALINE::save() {

std::ofstream fout("weight.txt");

for (int i = 0; i < W.size(); i++) {

fout << W[i] << " ";

}

fout << T;

fout.close();

}

void ADALINE::load() {

std::ifstream fin("weight.txt");

for (int i = 0; i < W.size(); i++) {

fin >> W[i];

}

fin >> T;

fin.close();

}

double ADALINE::getRandomNumber() {

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_real\_distribution<double> dis(0.1, 1.0);

return dis(gen);

}

ADALINE::ADALINE(int n) {

this->n = n;

W = std::vector <double>(n);

T = getRandomNumber();

alphaSpd = 0.04;

minMistake = 0.00033546;

for (int i = 0; i < n; i++) {

W[i] = getRandomNumber();

}

}

double ADALINE::calculateY() {

double res = 0;

for (int i = 0; i < W.size(); i++)

res += W[i] \* X[i].second;

return res - T;

}

void ADALINE::adjustWeight(int ind, double Y, double referense){

W[ind] -= alphaSpd \* (Y - referense) \* X[ind].second;

}

void ADALINE::adjustThreshhold(double Y, double referense) {

T += alphaSpd \* (Y - referense);

}

double ADALINE::calculateError(double Y, double referense) {

return 0.5 \* std::pow(Y - referense, 2);

}

void ADALINE::train(std::vector <std::vector<std::pair<double, double>>> trainSample) {

std::ofstream fout("train.txt");

int ind = 0;

X = trainSample[ind];

double reference = function(X[n - 1].first + 0.1);

double Y = calculateY();

while (calculateError(Y, reference) > minMistake && ind < trainSample.size() - 1) {

fout << ind << " " << calculateError(Y, reference) << "\n";

std::cout << std::setw(5) << std::left << ind

<< std::setw(13) << std::left << reference

<< std::setw(13) << std::left << Y

<< std::setw(13) << std::left << calculateError(Y, reference) << "\n";

ind++;

X = trainSample[ind];

reference = function(X[n - 1].first + 0.1);

Y = calculateY();

for (int i = 0; i < W.size(); i++)

adjustWeight(i, Y, reference);

adjustThreshhold(Y, reference);

}

fout.close();

}

double ADALINE::predict(std::vector<std::pair<double, double>> sample) {

X = sample;

return calculateY();

}

**Вывод:** изучил обучение и функционирование линейной ИНС при решении задач прогнозирования.