Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №2

за 1 семестр

По дисциплине: «МиАПР»

Тема: «Линейная искусственная нейронная сеть. Адаптивный шаг обучения»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Андрейчиков Д.С.

Проверил:

Крощенко А.А.

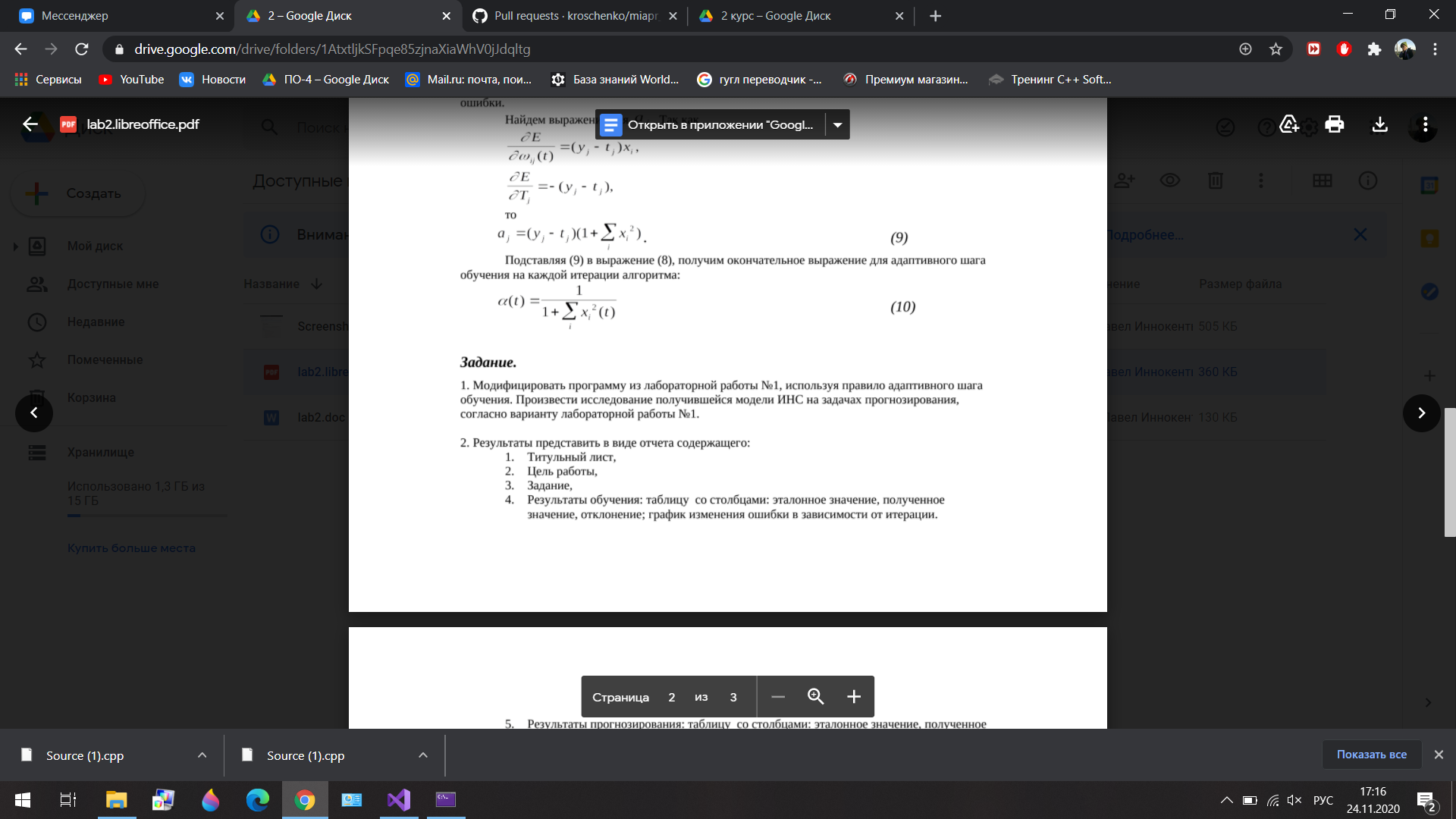
2020

Лабораторная работа №2

Линейная искусственная нейронная сеть. Адаптивный шаг обучения

Цель работы: изучить обучение и функционирование линейной ИНС с применением адаптивного шага.

Вариант 1



Код программы:

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <cmath>

using namespace std;

double function(int a, int b, double x, double d);

void print\_result(int n, double T, double\* Y, int n\_learn, int N, int n\_protected, double\* W);

int main() {

SetConsoleOutputCP(1251);

system("color f0");

srand(time(0));

int a = 1, b = 5, n = 3, n\_learn = 30, n\_predicted = 15;

double d = 0.1, step = 0.1, x1 = 0, E, Em = 0.001, T = 1;

double\* W = new double[n];

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

W[i] = 1.0 / (double)rand();

}

int N = n\_learn + n\_predicted;

double\* Y = new double[N];

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

x1 += step;

Y[i] = function(a, b, x1, d);

}

double y1, //выходное значение нейронной сети

alpha; //шаг обучения

int epoh = 0;

do {

E = 0;

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

y1 = 0;

double x2 = 0.0;

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

x2 += pow(Y[i + j], 2);

}

alpha = 1 / (1 + x2); //адаптивный шаг

for (int j = 0; j < n; j++) { //векторы выходной активности сети

y1 += W[j] \* Y[i + j];

}

y1 -= T;

for (int j = 0; j < n; j++) { //изменение весовых коэффициентов

W[j] -= alpha \* (y1 - Y[i + n]) \* Y[i + j];

}

T += alpha \* (y1 - Y[i + n]); //изменение порога нейронной сети

E += 0.5 \* pow(y1 - Y[i + n], 2); //расчет суммарной среднеквадратичной ошибки

}

epoh++;

} while (E > Em);

cout << epoh;

print\_result(n, T, Y, n\_learn, N, n\_predicted, W);

delete[] Y;

delete[] W;

return 0;

}

double function(int a, int b, double x, double d)

{

return a \* sin(b \* x) + d;

}

void print\_result(int n, double T, double\* Y, int n\_learn, int N, int n\_predicted, double\* W)

{

cout << "РЕЗУЛЬТАТЫ:" << endl;

cout << "1) Обучение:" << endl;

cout << setw(30) << left << "Эталонные значения" << setw(38) << left << "Полученные значения" << "Отклонение" << endl;

double\* prediction = new double[N];

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

prediction[i] = 0;

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

prediction[i] += W[j] \* Y[j + i];

}

prediction[i] -= T;

cout << "y[" << i << "] = " << setw(30) << left << Y[i + n] << setw(30) << left << prediction[i] << pow(Y[i + n] - prediction[i], 2) << endl;

}

cout << "2) Прогнозирование:" << endl;

cout << setw(30) << left << "Эталонные значения" << setw(38) << left << "Полученные значения" << "Отклонение" << endl;

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

prediction[i + n\_learn] = 0;

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

prediction[i + n\_learn] += W[j] \* Y[i + j + n\_learn - n];

}

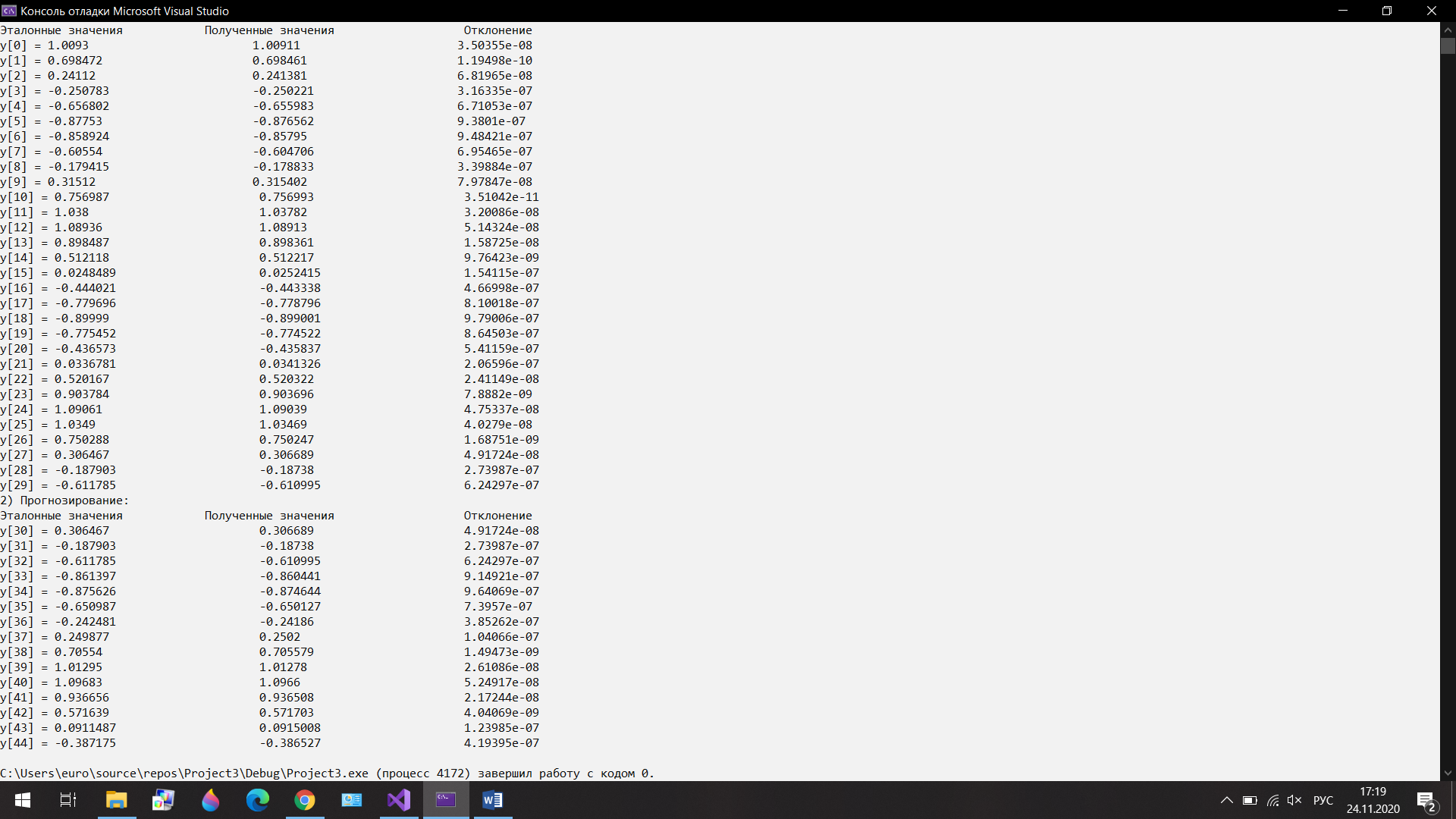
prediction[i + n\_learn] -= T;

cout << "y[" << i + n\_learn << "] = " << setw(30) << left << Y[i + n\_learn] << setw(30) << left << prediction[i + n\_learn] << pow(Y[i + n\_learn] - prediction[i + n\_learn], 2) << endl;

}

delete[] prediction;

}



E

era

Вывод: Изучил обучение и функционирование линейной ИНС с использованием адаптивного шага.