МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«**Вятский государственный университет**»

**(«ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №4

по дисциплине «Технологии программирования»

Выполнил студент группы ИВТ-21 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Щесняк Д. С./

Проверил доцент кафедры ЭВМ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Долженкова М. Л./

Киров 2016

1. Задание на лабораторную работу

Используя WinAPI реализовать один родительский и 2 дочерних окна. Родительское окно должно содержать кнопки, при нажатии на которые открывались соотвествующие окна. Одно из дочерних окон должно демонстрировать процесс обучения нейросети, а другое должно позволять протестировать корректность работы нейросети по распознаванию цифр. Реализовать нейросеть в отдельной динамической библиотеке.

1. Словесное описание алгоритма работы

Создается заголовочный файл, в котором объявляются все экспортируемые функции. Далее они описываются в основном файле исходного кода. Скомпилированную динамическую библиотеку можно подключить к клиентскому приложению двумя способами: при помощи явного и неявного связывания.  
В случае неявного связывания требуется к клиентскому проекту подключить заголовочный файл и lib файл библиотеки.

В случае явного связывания необходимо подключить заголовочный файл, явно загрузить библиотеку при помощи **loadLibrary()**, получить адреса функций библиотеки при помощи функции **getProcAddress().** После окончания работы с библиотекой ее необходимо освободить при помощи **freeLibrary().**

1. Экранные формы

Экранные формы представлены на рисунках 1-3.

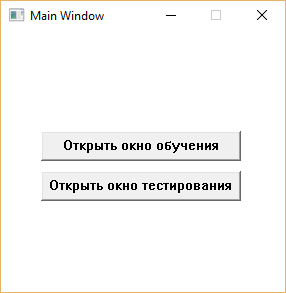


Рисунок 1 – Главное окно программы

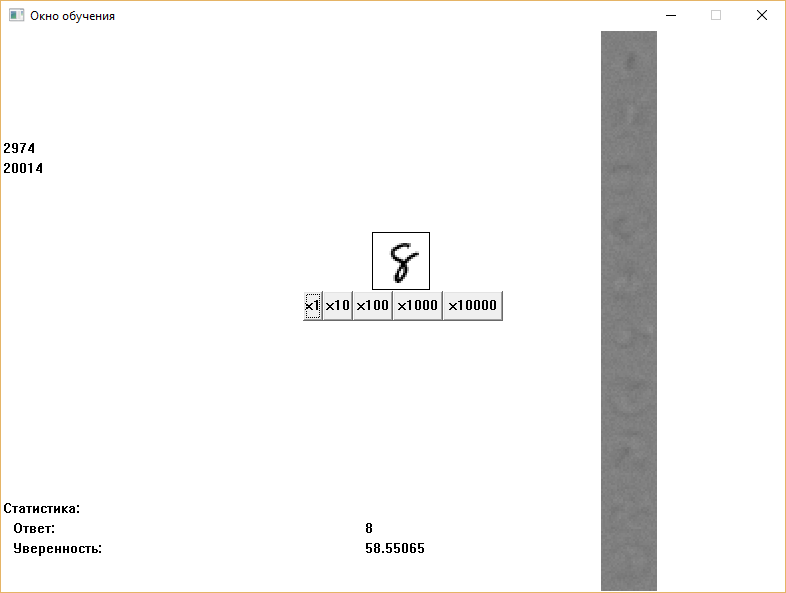


Рисунок 2 – Окно обучения

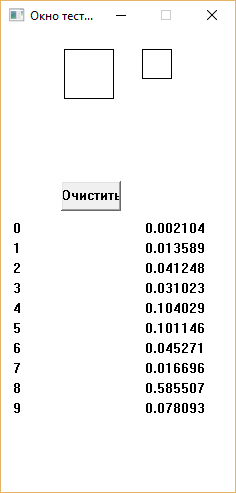


Рисунок 3 – Окно тестирования

1. Исходный код программы

Исходный код библиотеки представлен на рисунке 4. Исходный код явного присваивания представлен на рисунке 5.

|  |
| --- |
| #define COMPILING\_MYDLL  #include "stdafx.h"  #include "NeuralNet.h"  #include <math.h>  #include <cstdlib>  template <typename t>  t\* sub(t\* a, t\* b, int len) {  t\* out = new t[len];  for (int i = 0; i < len; i++)  {  out[i] = a[i] - b[i];  }  return out;  }  template <typename t>  t\* add(t\* a, t\* b, int len) {  t\* out = new t[len];  for (int i = 0; i < len; i++)  {  out[i] = a[i] + b[i];  }  return out;  }  double sigmoid(double x, bool deriv = false) {  if (deriv) {  return sigmoid(x) \* (1 - sigmoid(x));  }  else {  double d = 1.0 + exp(-x);  return 1.0 / d;  }  }  extern "C" \_\_declspec(dllexport) int run(NeuralNet\* NN, unsigned char\* k) {  NN->out = new double[10];  double\* input = new double[28 \* 28];  for (int i = 0; i < (28 \* 28); i++) {  input[i] = (double)k[i] / 1000;  }  NN->out = NN->l[0].dot(input);  NN->ans = 0;  for (int i = 1; i < 10; i++)  {  if (NN->out[i] > NN->out[NN->ans])  NN->ans = i;  }  NN->ac = NN->out[NN->ans];  return NN->ans;  }  extern "C" \_\_declspec(dllexport) void learn(NeuralNet\* NN, int n) {  double\* x = new double[10];  for (int i = 0; i < 10; i++)  {  x[i] = 0;  }  x[n] = 1;  x = sub(x, NN->out, 10);  NN->l[0].learn(x);  }  extern "C" \_\_declspec(dllexport) void draw(NeuralNet\* NN, HDC hdc, int x, int y) {  NN->l[0].draw(hdc, x, y);  NN->l[1].draw(hdc, x + 56, y);  }  NeuralNet NN;  extern "C" \_\_declspec(dllexport) NeuralNet\* getNeuralNet() {  return &NN;  } |

Рисунок 4 – Исходный код программы

|  |
| --- |
| typedef int(cdecl \*Run)(NeuralNet\*, unsigned char\*);  typedef void(cdecl \*Learn)(NeuralNet\*, int);  typedef void(cdecl \*Draw)(NeuralNet\*, HDC, int, int);  Run run;  Learn learn;  Draw draw;  int APIENTRY WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpszCmdLine, int nCmdShow) {  HINSTANCE lib = LoadLibrary("DLL.dll");  NN = (\*(NeuralNet\* (\*) ()) GetProcAddress(lib, "getNeuralNet")) ();  run = (Run) GetProcAddress(lib, "run");  learn = (Learn)GetProcAddress(lib, "learn");  draw = (Draw)GetProcAddress(lib, "draw");  run(NN, img.getImage());  WNDCLASSEX mainClass = CreateWinClass(szClassName, hInstance, MainProc);  WNDCLASSEX learnWindowClass = CreateWinClass("LearnWindowClass", hInstance, WndProc);  WNDCLASSEX testWindowClass = CreateWinClass("testWindowClass", hInstance, TestProc);  mainWin = mainWindow(hInstance, NULL, nCmdShow);  learnWin = learnWindow(hInstance, mainWin, nCmdShow);  testWin = testWindow(hInstance, mainWin, nCmdShow);  MSG msg;  while (GetMessage(&msg, NULL, 0, 0))  {  TranslateMessage(&msg);  DispatchMessage(&msg);  }  return 0;  } |

Рисунок 5 – Явное присваивание

1. Вывод

В ходе выполнения лабораторной работы были полученные необходимые знания базового синтаксиса языка программирования С++. Была изучена работа с динамическими библиотеками, явное и неявное связывание. Данные знания являются фундаментальными и необходимы для дальнейшего продолжения изучения различных технологий программирования и языка С++.