Министерство науки и высшего образования Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

по дисциплине «Интеллектуальные системы»

на тему « Реализация с помощью нейронной сети конвертер валют из долларов в евро»

Выполнил

студент группы №

931903

Лазаренко Иван

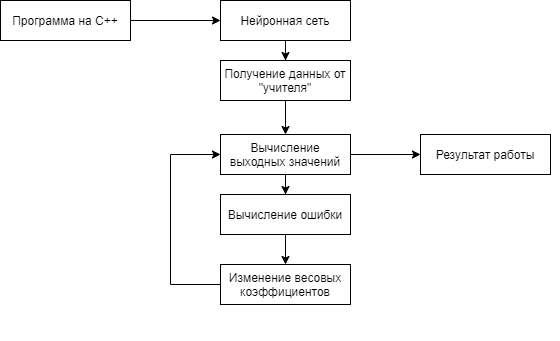
Томск 2021

**Цель работы:** понять принцип работы нейронной сети и научиться использовать его в реальных задачах.

**Постановка задачи:** реализовать с помощью нейронной сети конвертер валют из долларов в евро.

**Метод решения задачи:** разработка алгоритма с помощью объектно-ориентированного подхода и его реализация в программе Visual Studio на языке C++.

**Структурная схема алгоритма:**



**Нейронная сеть** – это множество связанных между собой нейронов; входные сигналы одних нейронов являются выходными сигналами для других.

В данной работе реализован подход обучения с учителем, то есть используется заранее сформированное множество обучающих примеров. А также используется алгоритм обратного распространения ошибки, чтобы минимизировать отклонения реальных значений выходных сигналов нейронной сети от требуемых.

Простое объяснение работы данной нейронной сети: нейронная сеть способна преобразовывать одну величину в другую. Мы вправе сами выбрать, что это будет за величина, например: километры в метры, секунды в минуты, доллары в евро и т.д.

Каким же образом это происходит? Происходит обучение ИНС с помощью метода "с учителем". Соответственно, учителем тут выступают 2 заданные величины: курс доллара и евро, взятых из гугла. Таким образом, как бы говорим нашей нейронной сети: смотри, вот в таком количестве содержится столько, так что давай, учись, потом приду, спрошу, сколько содержится в другом количестве, и не дай Бог не ответишь! Далее нейронная сеть начинает изучать данные и подстраивать вес, с помощью которого она и запоминает коэффициент, на который нужно менять то или иное значение. И так мы получаем результат, четкость которого зависит от "скорости обучаемости", которую мы дали нашей ИНС.

**Получение данных от “учителя”.** Вводим два массива данных USD и EU для доллара и евро соответственно.

**Вычисление выходных значений.** В методе ProcessInputdata считаем нейрон, умножая вход на вес. Вес которого в дальнейшем будет корректироваться в методе Train. Также реализван обратрый метод вывода RestoreInputdata, чтобы получить значения долларов из евро.

**Вычисление ошибки.** Вычисление ошибки происходит методом обратного распространения ошибки, то есть ожидаемый результат минус актуальный.

**Изменение весовых коэффициентов.** Далее нужно изменить вес после того, как вычислили величину ошибку, меняем вес для более точного получения результата.

**Листинг программы:**

**Github : https://github.com/bleskaces/IS3**

#include <iostream>

using namespace std;

class neuron { //класс, обозначающий наш нейрон

double w = rand()%2; //значение веса

public:

double ProccesInputdata(double input) { //метод, отвечающий за вывод значения, просчитанного нейроном == вход\*вес

return (input \* w);

}

double RestoreInputData(double output) { //обратный метод вывода, для получения из выхода значение входа

return (output / w);

}

void Train(double input, double expected){ //метод тренировки нейрона

double actresult = input \* w; //результат, полученный нерйоном на данный момент

LastError = expected - actresult; //величина ошибки на данный момент

double correct = (LastError / actresult)\*speed; //величина треубемой корректировки на данный момент

w += correct; //изменение веса на величину корректировки

}

double LastError;

double speed=0.000001; //скорость обучения нейрона. Чем ниже - тем качественнее, но медленнее.

};

int main()

{

double USD[5] = { 1,7,12,15,28 }; //"учитель", обозначающий

double EU[5] = { 0.82, 5.79, 9.91, 12.38, 23.11 }; //настоящее значение

int r = rand() % 5;

neuron n; //создание объекта "нейрон"

int i = 0;

while (n.LastError > n.speed || n.LastError < -n.speed) { //цикл обучения

i++;

n.Train(USD[r], EU[r]);

if (i % 10000 == 0) {

cout << "error's number: " << i << " Error: " << n.LastError << endl; //вывод информации об обучении: попытка и значение ошибки

}

}

cout << "100 dollars equals " << n.ProccesInputdata(100) << " euro" << endl; //финальные

cout << "10 euro equals " << n.RestoreInputData(10) << " dollars" << endl; //ответы

cout << "40 dollars equals " << n.ProccesInputdata(40) << " euro" << endl;

cout << "50 euro equals " << n.RestoreInputData(50) << " dollars" << endl;

cout << "3 dollars equals " << n.ProccesInputdata(3) << " euro" << endl;

cout << "356 euro equals " << n.RestoreInputData(356) << " dollars" << endl;

}

**Результаты работы алгоритма:**



**Вывод:**

Применение нейронной сети не может выдать точной результат, а выдает максимально приближенный. Но нейронные сети затронули почти каждый аспект жизни человека, так как применяются в разных сферах, и могут подстраиваться под нужные задачи.