

Udacity C++ Nanodegree

Project: Capstone Neural Network Deployment

Overview

This C++ Project is the final of Udacity C++ Nanodegree. It's a deployment of a feed forward NN on C++ without the use of third party.

The code is divided to the following component:

- Network: where the topology of the NN is defined and the Main Structure is applied.
- Layer: defines the structure of any Layer "Input, Hidden, Output" each with the specified number of neurons.
- Neuron: In Neuron , the basic functionality is written for a single Neuron. Activation Function have been defined as Sigmoid Function.
- Matrix: The basic implementation of Matrix also performs transpose of Matrix
- Multiply_Matrix: performs the multiplication of any two matrices

The project main scope was OOP that's why Classes was used intensively in multiple files.

Upon launching the user is asked to insert number of Neurons in each Layer, then the user have to insert the values of input layers.

Weights are initialized randomly, and the program performs Feed-Forward Propagation and prints the Result to Console.

Basic Build Instructions

1. Compile: `cmake .. && make`
2. Run it: `./Neural`

Rubric Items Matched

1 Object Oriented Programming - The project uses Object Oriented Programming techniques.

The project code is organized into classes with class attributes to hold the data, and class methods to perform tasks.

The program have classes deployed in multiple files (Neuron.h , Network.h, Layer.h , Matrix.h , Multiply_Matrix.h)

2 Object Oriented Programming - Classes use appropriate access specifiers for class members.

All class data members are explicitly specified as public, protected, or private.

In all Classes , the members were defined as Public and Private in all the files.

3 Object Oriented Programming - Classes abstract implementation details from their interfaces.

All class member functions document their effects, either through function names, comments, or formal documentation. Member functions do not change program state in undocumented ways.

Abstraction was done in all classes . Especially, Multiply_Matrix.h where the user will find only the declaration only while the definition can be found in Multiply_Matrix.cpp

4 Loops, Functions, I/O - The project accepts user input and processes the input.

The project accepts input from a user as part of the necessary operation of the program.

The project accepts user input in `main.cpp` line 13 , 16 , also a for loop in line 23 gets the user input in line 24. after that it process the data and print the results to console.

5 The project demonstrates an understanding of C++ functions and control structures.

A variety of control structures are used in the project and The project code is clearly organized into functions.

The project is built on multiple files , where each file is a class where many methods and function were used as in `void Network::feedForward()` in `Network.cpp` line 51, which is the function that performs the FFNN.