

Description of Code for Subtask 1 Assignment 1 COP290

Avval Amil 2020CS10330

31st January 2022

1 Introduction

In my code (file named yourcode.cpp), I have defined a Matrix class and have used it to do all of the computations regarding matrices. The class has 3 attributes - the row size, the column size, and a vector of vectors containing floating point entries, this 2d vector will have all values initialised to zero when the constructor function for this class is called, and the constructor will take in two values- a row size and a column size.

After this, there are a bunch of functions whose inner workings can be seen from the code (I have included many comments so that all the loops conditionals etc. are understandable). These functions will consist of the bulk of my code, and the main function is simply going to call them according to the parameters it will receive as input. These functions are:

- Constructor function for Matrix class
- writeMatrix - to write a matrix object to a specified file
- readMatrix - to read entries from a file and create a matrix
- AddMat - a function to add two objects of the Matrix class and return a third object of the same class which is their sum
- MultMat - a function to multiply two matrices and return their product (standard matrix multiplication algorithm has been used)
- relu - to calculate the effect of relu on a float
- tanhyperbolic - to calculate the hyperbolic tangent of a float
- relu (overloaded for matrices) - to perform relu on every element of a matrix
- tanh - to perform tanhyperbolic on every element of a matrix

- maxPool - performs max pooling on a matrix and returns another matrix
- avgPool - similar to maxPool, but performs average pooling
- readVector - to read the entries of a vector from a file
- writeVector - to write a vector to a specified file
- softmax - to perform the softmax transformation on a vector
- sigmoid - to perform the sigmoid transformation on a vector

The rest of the functions are just implementations of the tasks asked in the assignment, the ones above I have created for my own convenience and so that the code can be neater.

Finally, in the main function, I have taken input from the user, and according to that performed the required steps. Errors have been raised wherever I felt that an incorrect input could be entered, and a line is written to standard error in these cases. I have also included a makefile file which when invoked creates a yourcode.out file.

2 How to Run

Open the appropriate directory in the terminal. Enter 'make' so that the yourcode.out file can be created. Then use the following commands according to what is required:

- Fully Connected: './yourcode.out fullyconnected inputmatrix.txt weightmatrix.txt biasmatrix.txt outputmatrix.txt'
- ReLU activation : './yourcode.out activation relu inputmatrix.txt outputmatrix.txt'
- Tanh activation: './yourcode.out activation tanh inputmatrix.txt outputmatrix.txt'
- Max Pooling: './yourcode.out pooling max inputmatrix.txt stride outputmatrix.txt'
- Average Pooling: './yourcode.out pooling average inputmatrix.txt stride outputmatrix.txt'
- Softmax Transformation: './yourcode.out probability softmax inputvector.txt outputvector.txt'
- Sigmoid Transformation: './yourcode.out probability softmax inputvector.txt outputvector.txt'

All the words with a .txt at the last denote text files that should be in the same directory containing the input data (except the output.txt, which may or may not be present). The names of these files can be anything, the ones specified above are just placeholders. The 'stride' should not be written as it is above but instead should be replaced by a positive integer denoting the stride length to be used. All other words should be written exactly as they are above, or errors will be raised.