Readme -Subtask-2 For Creating a small audio processing library

Ujjwal Mehta 2020CS10401
Avval Amil 2020CS10330

February 21, 2022

Introduction to task :-

Explanation of Code Structure :- This subtask is an extension of the previous one. We have used three ways of multiplying matrices here, using openblas, mkl and using multi-threading. Any one of these three can be used to multiply matrices when you call the functions made in part 1 of this task. We have also compared the performance of these three ways of multiplying matrices, and found that openblas generally performs the best, though it is only very slightly better than mkl. Our own implementation using mkl is not able to keep up with these other two implementations, it is way slower. We have generated data for multiplying matrices of sizes 1 through 300 using each of these three ways, and plotted the appropriate boxplots using gnuplot. The data was generated by a c++ code in the file gnu.cpp (this file only serves to generate the data that we have used, it is not compiled using makefile).
One more thing to note is that, in the boxplots, the times in the y-axis are given in microseconds. The data used to generate the plots is also included in the submission.

The main file is called 'main.cpp'. It contains the class Matrix, and all its operations made in part 1. The changes made from the last part are as follows- There can be a new command line argument (we have assumed that this argument is optional). If the new argument (which will be the last command line argument is either one of 'mkl', 'openblas' or 'pthread' (as these three are only used for multiplication, they are only used with 'fullyconnected', then the respective

method of multiplying will be used. If this argument is not present, then the multiplication will be carried out just as it was in part one of the subtask. If some garbage input is given in the last argument, there will be an error.

Preprocessor Directives Used in Code:-

The directives used in our code are **vector, iostream, cmath, fstream** where vector is used to use vectors in our code, cmath for exponential function and iostream and fstream for reading and writing files. We have also included three header files (one each for the new ways of multiplying matrices). The three header files are named 'mkl_matrix.h' 'open_blas.h' and 'p_thread.h'. The appropriate multiplication processes are present in these header files.
In addition to this, we have also used include algorithm **only** in the data generation code, so that data can be sorted accordingly and plotted. This is not present in the main.cpp file.

Executing the program:-

Open command line in the directory where the files are present. Then type 'make'. Then use the format specified in the last subtask, with the extra addition that a new argument may be present as the last one. Examples of valid inputs can be-

./yourcode.out fullyconnected inputmatrix.txt weightmatrix.txt biasmatrix.txt outputmatrix.txt openblas

./yourcode.out fullyconnected inputmatrix.txt weightmatrix.txt biasmatrix.txt outputmatrix.txt mkl

./yourcode.out fullyconnected inputmatrix.txt weightmatrix.txt biasmatrix.txt outputmatrix.txt pthread

./yourcode.out fullyconnected inputmatrix.txt weightmatrix.txt biasmatrix.txt outputmatrix.txt

./yourcode.out activation relu inputmatrix.txt outputmatrix.txt

./yourcode.out activation tanh inputmatrix.txt outputmatrix.txt

./yourcode.out pooling max inputmatrix.txt stride outputma-

trix.txt

./yourcode.out pooling average inputmatrix.txt stride outputmatrix.txt

./yourcode.out probability softmax inputvector.txt outputvector.txt

./yourcode.out probability sigmoid inputvector.txt outputvector.txt