CMSC5743 Lab 01

General Matrix Multiply

1 Sample Code:

- Go to the `./Lab1-GEMM-1/code`
- Run `sh run_matmul.sh` in your terminal
- You can turn annotations **on** or **off** in the code to get different runtime performance
  - `matmul()` (line 104)
  - `matmul_ikj()`
  - `matmul_AT()` (line 105)
  - `matmul_BT()` (line 106)

2 Assignments:

Q1 The shape of matrix A is $I \times K$ and the shape of matrix B is $K \times J$. Please explore the different shapes of matrix A, B mentioned in `matmul.cpp` to get the different runtime performance under `matmul()`, `matmul_ikj()`, `matmul_AT()`, `matmul_BT()`. The value of I, K, J will be fixed at 256,512 or 1024. You can refer to the report-format to finish your submission.

Q2 Learn the `im2col` from the **Useful Materials** Section to implement it from scratch using C++ to optimize standard convolution operation.

- batch: 1
- height_feature: 56
- width_feature: 56
- in_channels: 3
- out_channels: 64
- kernel_size: 3
- stride: 1
- padding: 0

Q3 Based on the findings obtained in **Q1**, explore different techniques for cache optimization. For instance, the writing operation of matrix C is not consistent, and how to improve the write caching? Try at least two different optimization techniques to improve the cache hit ratio and reduce the matrix multiply time consumption. Here are some examples you may use:

- Loop unrolling
- Writing caching
- Tiling
- Vectorization (SIMD)
- Array packing

Record the performance of your implemented optimization approaches, and discuss your findings.

Useful Materials:

- MATLAB im2col
- Making faster
- ConvNets in practice
- GEMM Optimization on CPU

Tips: You should learn the code style from the sample code to build your project.