Project 3: Improved Matrix Multiplication in C

Matrix multiplication, a very basic operation in linear algebra, plays an important role in deep learning. Please implement matrix multiplication in C and try your best to improve its speed.

Requirements

1. Only `float` is needed since it is the most commonly used data type in deep learning.
2. Implement a function `matmul_plain()` in a straightforward way using several loops as the benchmark.
3. Implement a function `matmul_improved()` using SIMD, OpenMP and other technologies to improve the speed. You can compare it with `matmul_plain(`)`.
4. Test the performance using 16x16, 128x128, 1Kx1K, 8Kx8K and 64Kx64K matrices. You can generate some matrics with random values to test your function.
5. Compare your implementation with the matrix multiplication in OpenBLAS (https://www.openblas.net/). The results of your implementation should be the same or very similar to those of OpenBLAS. Can your implementation be as fast as OpenBLAS?
6. [Optional] Test your program on X86 and ARM platforms, and describe the differences.

Tips

1. You can define a structure for a matrix, such as

```
struct Matrix
{
    size_t rows;
    size_t cols;
    float * data;
};
```

2. Please implement your functions professionally. At least to check the parameters before using them.
3. Be careful with the memory management. No memory leak.
4. It is meaningless to measure the efficiency of a program that is generated in the DEBUG mode. The DEBUG mode is the default mode of many IDEs. If you compile a C program with the option `-O3`, you will be surprised.

Rules

1. **Deadline**: Please submit your project report before its deadline. The deadline is **23:59, April 28**. After the deadline (even 1 second), **0 score!** No deadline extension for any students.

2. **Format**: Please submit the files as: *.pdf, xxx.c, xxx.c, xxx.h. Please do **NOT** put the files into a compressed one. To uncompress a lot of files will cost time for instructors. PDF format can make the layout of the report to be consistent on any computer.

3. **Report**: Your score will also depend on the quality of your source code **and your report**. Your report should be easy to understand and describe your work well, especially the highlights of your work. If you do not know how to write the report, just assume you are a developer in a company and are reporting to your manager that you have developed a good calculator.

4. **Code style**: Please pay more attention to your code style. This is not ACM-ICPC contest. You have enough time to write code with both correct results and a good code style. You will get a deduction if your code style is terrible. You can read Google C++ Style Guide (http://google.github.io/styleguide/cppguide.html) or some other guide for code style. You may lose points on your project if your code is ugly.

5. **ChatGPT**: You can use ChatGPT as your personal coach. Such as you can use ChatGPT or some other similar tools to polish the text in your report. But ChatGPT can also generate some rubbish that looks good but contains nothing. It will waste the time of the instructor to review those empty words, and also make you lose points on your project. ChatGPT can also generate a framework of the source code, and help improve the quality of your source code. If you use it for that, please mention it in the report. You are responsible for all materials you submit, not ChatGPT.