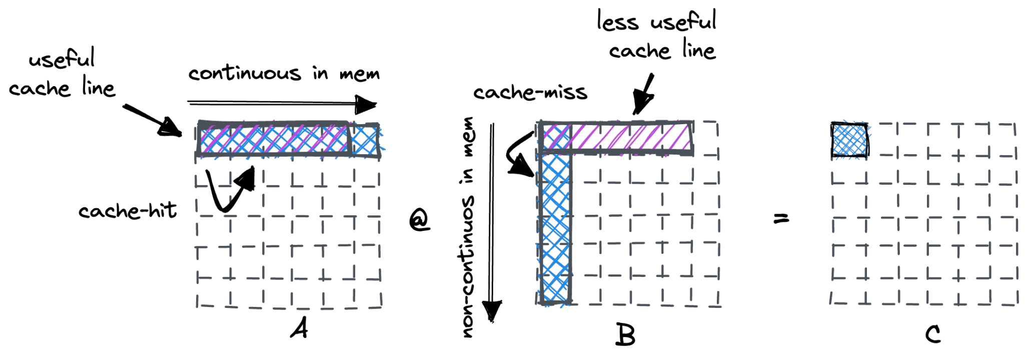
Matrix multiplication and dense neural networks

Prof. Luca Formaggia, Matteo Caldana

September 22, 2023



1 Introduction

The simple matrix-matrix multiplication is the operation that stands at the basis of almost all neural networks, however an efficient implementation is by no means trivial.

2 Objectives

Write a code that performs the matrix-matrix multiplication [1]. Your code should support both `float` and `double`. The size of the matrix should be able to be specified at run-time. Run strong and weak scalability test and report the results. Compare the results and elapsed time of your code with the ones of the OpenBLAS library (available in the mk modules) [2]. Employ the following techniques to make your code go faster, quantify the improvements you get with each of them:

- Exploit the SIMD instruction [3].
- Exploit loop unrolling [4].
- Make sure that your code is cache friendly [5,6,7].
- Use OpenMP to parallelize the code.

Use the matrix-matrix multiplication to build and train a dense feed-forward neural network. You can use a simple dataset of your choice and hard-code the computations for the gradient and use a simple gradient descent as optimizer.

3 Ideas for an exam project

- Parallelize the code with CUDA.
- Make your code more general: automatize the computations of the gradient, handle different activation functions, implement more sophisticated optimizers (such as momentum or ADAM).

References

[1] https://inst.eecs.berkeley.edu/ cs61c/fa17/lec/18/L18%20SIMD%20(1up).pdf
[2] https://github.com/OpenMathLib/OpenBLAS
[3] https://en.wikipedia.org/wiki/Single_instruction,_multiple_data
[4] https://en.wikipedia.org/wiki/Loop_unrolling
[5] https://en.wikipedia.org/wiki/Cache_performance_measurement_and_metric
[6] https://en.wikipedia.org/wiki/Cache_hierarchy
[7] https://siboehm.com/articles/22/Fast-MMM-on-CPU