Assignment I: The softmax function

Luca Lombardo

Results

This section presents a comparative analysis of three softmax function implementations under varying conditions. We evaluate the `softmax_auto` implementation with and without the `parallel` directive and compare performance between AVX2 and AVX512 instruction sets.

Performance

We compare the execution time of three softmax implementations across various input sizes, analyzing the effects of parallelization and vectorization instruction sets on the auto-vectorized implementation. Figures 1 and 2 demonstrate performance without parallelization, while Figures 3 and 4 show results with parallelization enabled.

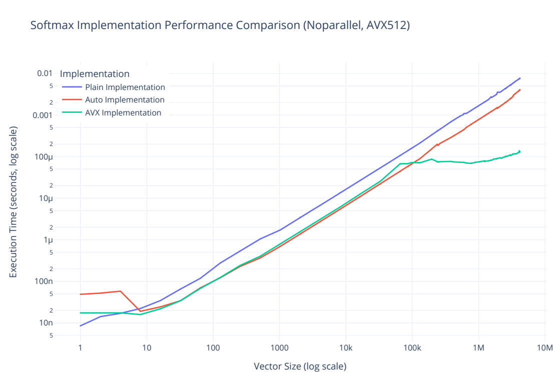


Figure 1: Performance of softmax implementations without parallelization and with AVX512 instructions.

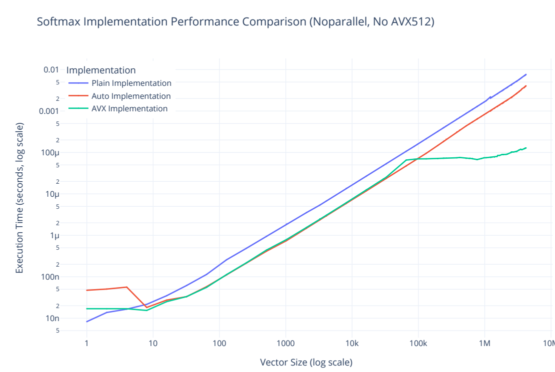


Figure 2: Performance of softmax implementations without parallelization and without AVX512 instructions.

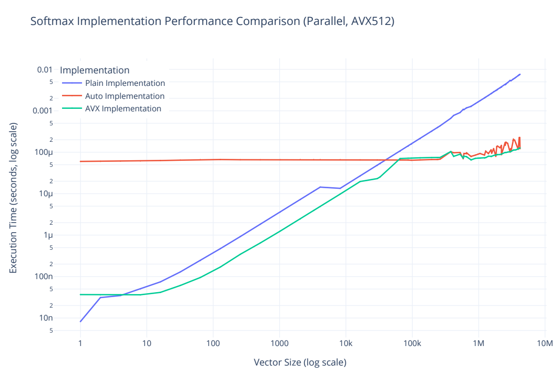


Figure 3: Performance of softmax implementations with parallelization and AVX512 instructions.

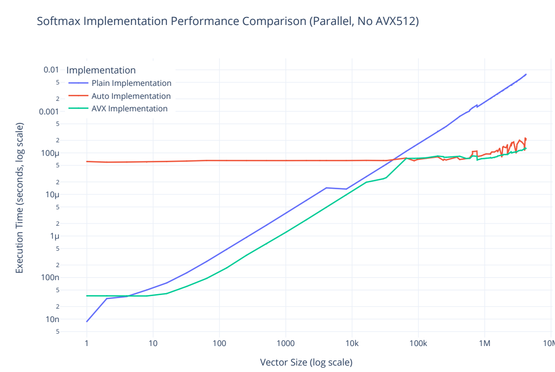


Figure 4: Performance of softmax implementations with parallelization but without AVX512 instructions.

Speedup

We analyze the relative speedup of various configurations compared to the baseline plain implementation. Figures 5 through 8 illustrate the performance gains achieved through different combinations of parallelization and vectorization techniques.

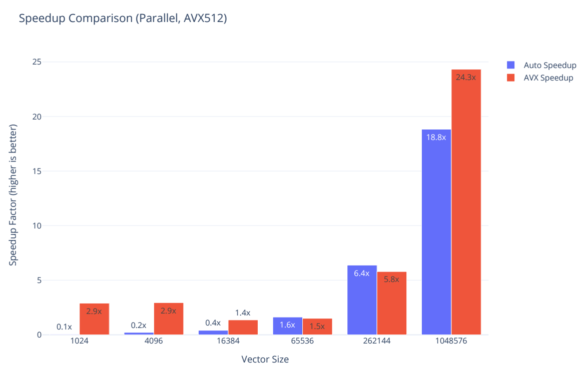


Figure 5: Speedup with parallelization and AVX512 instructions.

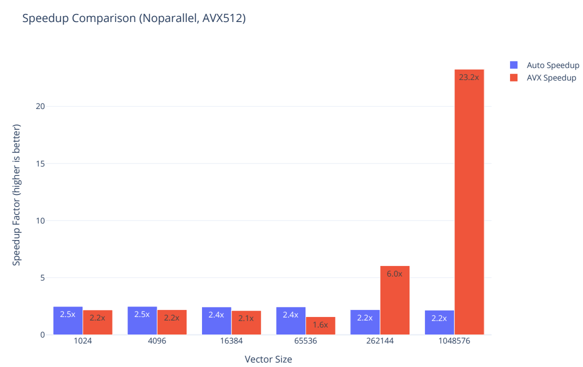


Figure 6: Speedup without parallelization but with AVX512 instructions.

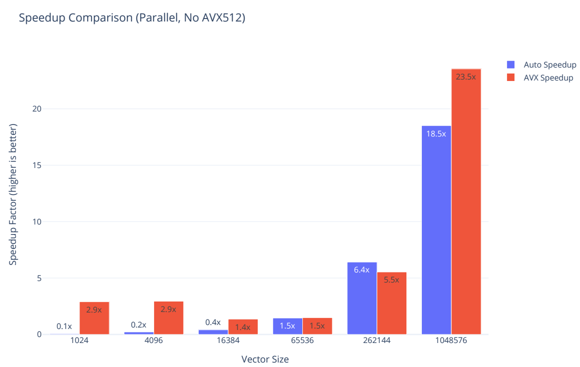


Figure 7: Speedup with parallelization but without AVX512 instructions.

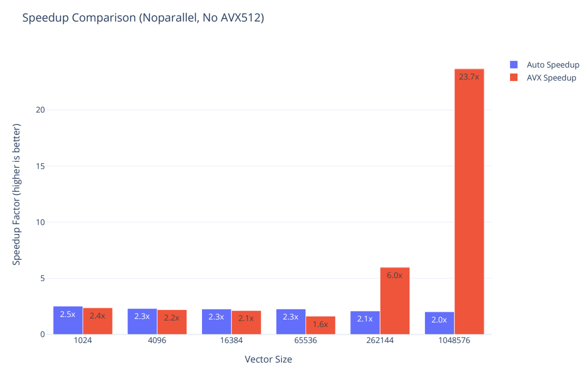
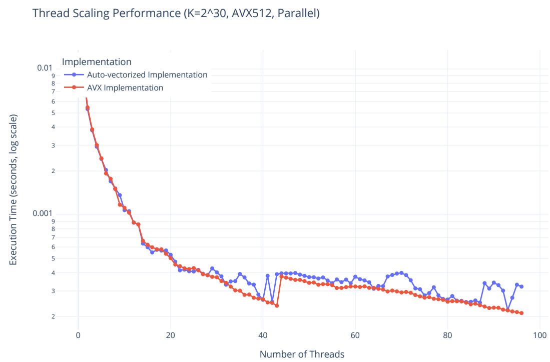


Figure 8: Speedup without parallelization and without AVX512 instructions.

As evidenced in Figures 5 and 7, parallelization significantly enhances the performance of the auto-vectorized implementation. Conversely, the impact of AVX512 instructions appears minimal when comparing Figures 5 with 7 and 6 with 8.

Scalability

We evaluate thread scalability using a fixed large input size ($K = 2^{30}$) while varying thread count from 1 to 96. Figure 9 shows the execution times, while Figure 10 presents the speedup relative to single-threaded execution alongside the theoretical Amdahl's Law prediction. Notably, a performance discontinuity occurs at approximately half the maximum thread count, corresponding to the number of cores in one of the system's two physical processors.



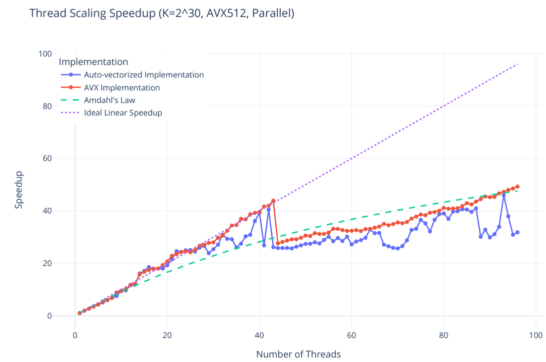
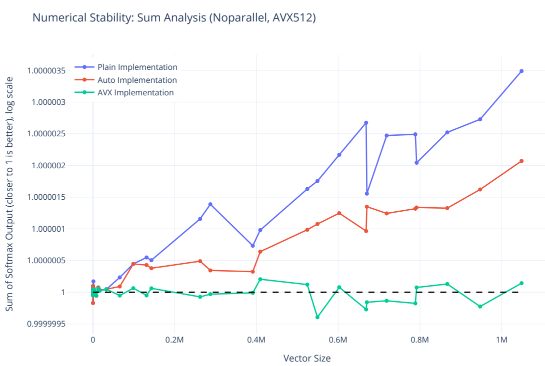


Figure 9: Execution time scaling with thread count for the softmax implementations.

Figure 10: Thread scaling speedup compared to Amdahl's Law prediction.

Numerical Stability

The numerical stability comparison in Figures 11 and 12 demonstrates how each implementation handles numerical challenges across varying input conditions.



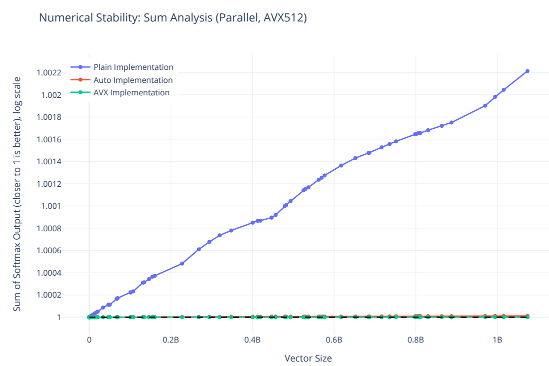


Figure 11: Numerical stability without parallelization.

Figure 12: Numerical stability with parallelization.