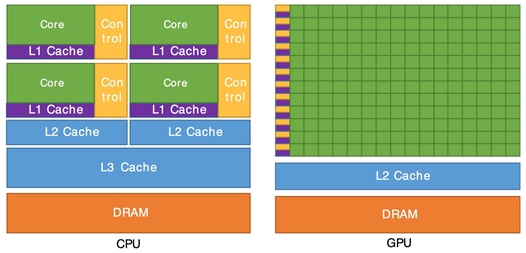
Python vs Julia: Computational Speed of Various Matrix Operations

Yug Patel and Raghav Gn.
Department of Industrial and Manufacturing Engineering

Introduction

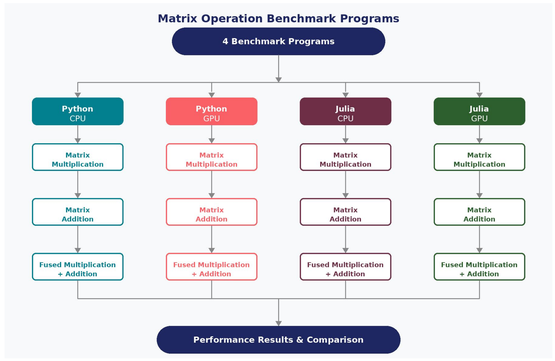
- Neural networks power large language models (LLMs) such as ChatGPT or Claude.
- They rely on on large matrix operations like addition and multiplication.
- Graphical processing units (GPUs) excel at parallel calculations which makes them well suiting for large matrix operations.
- CUDA units on GPUs allow for fused multiplication and addition operations improving operational efficiency.



- Python is commonly used for LLMs due to its extensive ecosystem and relative simplicity [1].
- Python struggles to compete with languages such as C++ without libraries.
- Julia was developed to integrate the fast speeds of C and the simple syntax of Python.
- Julia uses just-in-time (JIT) compilation to convert code into machine instructions before-hand.
- This project aims to compare the computational speeds of Python (with libraries) to Julia when performing large matrix addition, multiplication and fused multiplication+addition.

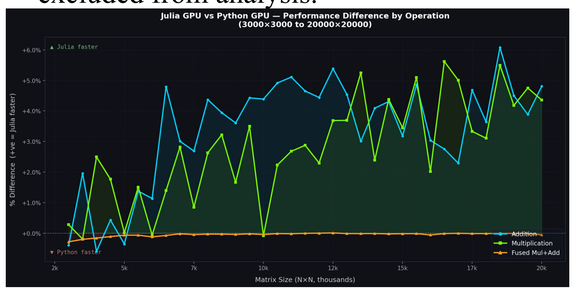
Methodology

- Two programs were developed — one in Python, one in Julia — each performing addition, multiplication, and fused multiplication/addition operations.
- Two equivalent programs were also created for CPU comparison.
- Operations were performed on matrices ranging from 500×500 to 20000×20000 in increments of 500, with 1000 operations per interval (100 operations for CPU due to time constraint).
- Matrices were pre-allocated with empty slots to prevent reallocation, then filled with random numbers before each set of operations.
- 20 warmup repetitions were performed before timing to ensure accurate timing.
- Timing was within the program, capturing the time taken only for 1000/100 operations and excluding random number generation and other intermediate functions.
- Programs were run on a Linux workstation with an Nvidia RTX 5000 Ada Generation GPU and an AMD Ryzen Threadripper PRO 7985WX CPU (64 cores).
- A separate Python script was used to visualize the collected data.

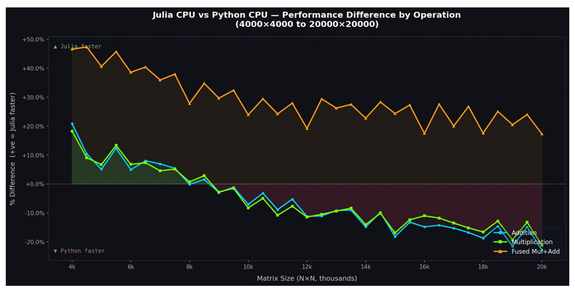


Results

- GPU-based results show Julia slightly outperforming Python for multiplication and addition on matrices larger than 3000×3000.
- The performance gap widens as matrix size increases which suggests Julia scales more efficiently with larger matrix sizes.
- For fused multiply-add operations on the GPU, Python slightly outperformed Julia for matrices below 6500×6500 after which both languages performed at the same level.
- Results for matrices below 3000×3000 were highly variable and unreliable and thus excluded from analysis.



- CPU results show Julia consistently outperforming Python for fused multiply-add operations across all matrix sizes.
- For CPU-based multiplication and addition, Julia outperforms Python for smaller matrices (below 8000×8000), however this trend reverses as matrix size surpasses 8000x8000.



Conclusion

- For fused multiply-add operations on the GPU, Python is the preferred language, especially for smaller matrices.
- Julia is slightly advantageous for large matrix addition and multiplication on the GPU, while Python is the stronger choice for CPU operations [2].
- Caution is advised when applying Julia to neural networks due to its relatively underdeveloped ecosystem, making it harder to apply in a wide variety of hardware and situations [1].

Future Recommendations

- Benchmark Julia and Python on established neural networks to measure performance in more practical situations.
- Extend benchmarking beyond 20000×20000 matrices to further measuring scaling effects of Julia and Python.
- Measure power consumption and other operating costs alongside computational speed.
- Expand benchmarking to specialised neural network GPUs, such as Nvidia's Blackwell architecture (H100/H200), to evaluate performance on more modern hardware.

Works Cited

[1] Korobeynikov et al., "The State of Julia for Scientific Machine Learning," *arXiv*, 2024.
[2] Bishnu et al., "Comparing the Performance of Julia on CPUs versus GPUs," *Geoscientific Model Development*, 2023. DOI: 10.5194/gmd-16-5539-2023