Benchmarking On-Device Machine Learning on Apple Silicon with MLX

Oluwaseun A. Ajayi Ogundepo Odunayo

BACKGROUND

The recent widespread adoption of Large Language Models (LLMs) and machine learning in general has sparked research interest in exploring the possibilities of deploying these models on smaller devices such as laptops and mobile phones. To address this need, Apple's machine learning research team introduced MLX (Hannun et al., 2023), a framework optimized for ML computations on their proprietary silicon devices, facilitating easier research, experimentation, and prototyping.

This study presents a performance evaluation of MLX, focusing on inference latency of transformer models. We compare the performance of different transformer architecture implementations in MLX with their Pytorch counterparts. For this research we create a framework called MLX-transformers which includes different transformer implementations in MLX.

Our study benchmarks different transformer models on two Apple Silicon macbook devices against an NVIDIA CUDA GPU. Specifically, we compare the inference latency performance of models with the same parameter sizes and checkpoints.

RELATED WORK

Apple Silicon Performance in Scientific Computing:

(Kenyon and Capano, 2022) explored the potential of Apple Silicon processors, particularly the M1 and M1 Ultra, in scientific computing. Their study found that Apple Silicon processors outperform state-of-the-art data-center GPUs in single-precision computing tasks, despite lacking double precision GPU capabilities.

MLX-Based Libraries and Frameworks: The release of MLX has sparked a surge of research and development efforts aimed at leveraging the framework's capabilities on Apple Silicon devices. Notable libraries and frameworks include::

- DiffusionKit: Facilitates running Diffusion Models on Apple Silicon with Core ML and MLX.
- MFLUX (MacFLUX): A line-by-line port of the FLUX implementation from the Huggingface Diffusers library to Apple MLX.
- MLX-VLM: Enables the execution of Vision Language Models on Apple Silicon devices using MLX.
- FastMLX: A production-ready API for hosting MLX models, including both VLMs and LMs.
- Local Chat: A private AI chatbot built on MLX, allowing users to run the latest open language models on Mac, iPad, and iPhone.

These projects collectively illustrate the growing ecosystem around MLX, spanning image generation, vision-language models, and educational resources.

EXPERIMENTS

Our experiments aimed to benchmark the latency performance of MLX operations and transformer models on various hardware configurations:

Hardware:

- 8GB Apple M1 Macbook Pro
- 32GB Apple M2 Max Macbook Pro
- NVIDIA A10 (24 GB PCIe) on AWS EC2 instance (30 vCPUs, 205.4 GB RAM, 1.5 TB SSD)

Benchmarks:

1. Operations Benchmarking: We evaluated each MLX implementation of the machine learning operations on both the GPU and CPU of Apple Silicon devices. We then compared these results with their PyTorch equivalents on the CUDA GPU, performing five iterations per test to ensure accuracy and reliability.

2. Model Inference Benchmarking: We focused on BERT, RoBERTa, and XLM-RoBERTa models. We measured the inference times of the MLX implementations on both the GPU and CPU backends of the Apple Silicon devices and compared them with the PyTorch equivalents on the CUDA GPU. We varied the input lengths to 50, 100, 200, and 500 characters, and adjusted the batch sizes to 1, 16, and 32. Each test was conducted over ten iterations.

RESULTS: Operation Benchmarks

Our benchmarking of the machine learning operations compared the performance of MLX on Apple Silicon (M1) against PyTorch on the NVIDIA A10 CUDA GPU, focusing on operations crucial for transformer architectures. While the CUDA GPU consistently outperformed the Apple M1, the M1's performance remains noteworthy for a consumer device.

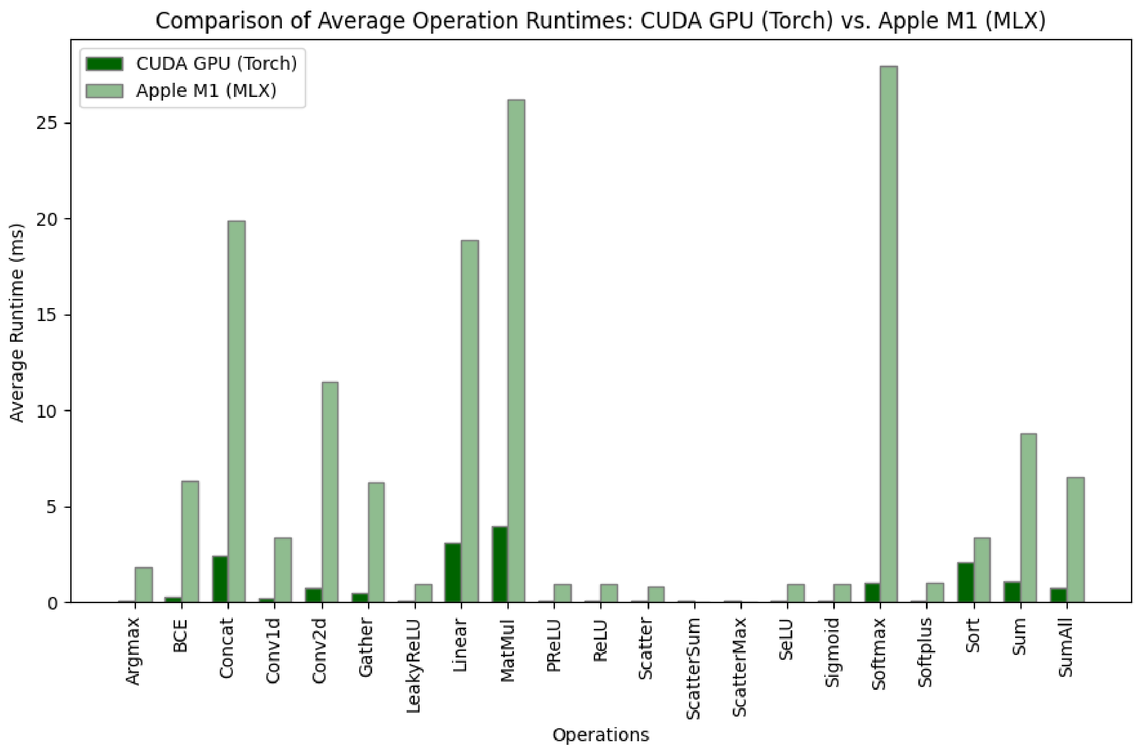

Figure 1. Comparison of operation runtimes (in milliseconds) between MLX on Apple M1 GPU and PyTorch on the CUDA GPU. Lower bars indicate better performance.

Key observations:

- Matrix multiplication: 3.96ms (CUDA) vs. 26.19ms (M1)
- Linear transformations: 3.11ms (CUDA) vs. 18.88ms (M1)
- Softmax: 1.06ms (CUDA) vs. 27.91ms (M1)

These results demonstrate that while the CUDA GPUs maintain a performance edge, Apple Silicon devices offer promising capabilities for on-device machine learning tasks, particularly for research, experimentation, and applications that don't require the absolute highest performance.

RESULTS: Model Inference Benchmarks

Our model inference benchmarking compared MLX performance on Apple Silicon devices (M1 and M2 Max) against PyTorch on a CUDA-enabled GPU (NVIDIA A10). We computed the inference latency of BERT (base and large), RoBERTa (base), and XLM-RoBERTa (base) models across various input lengths and batch sizes.

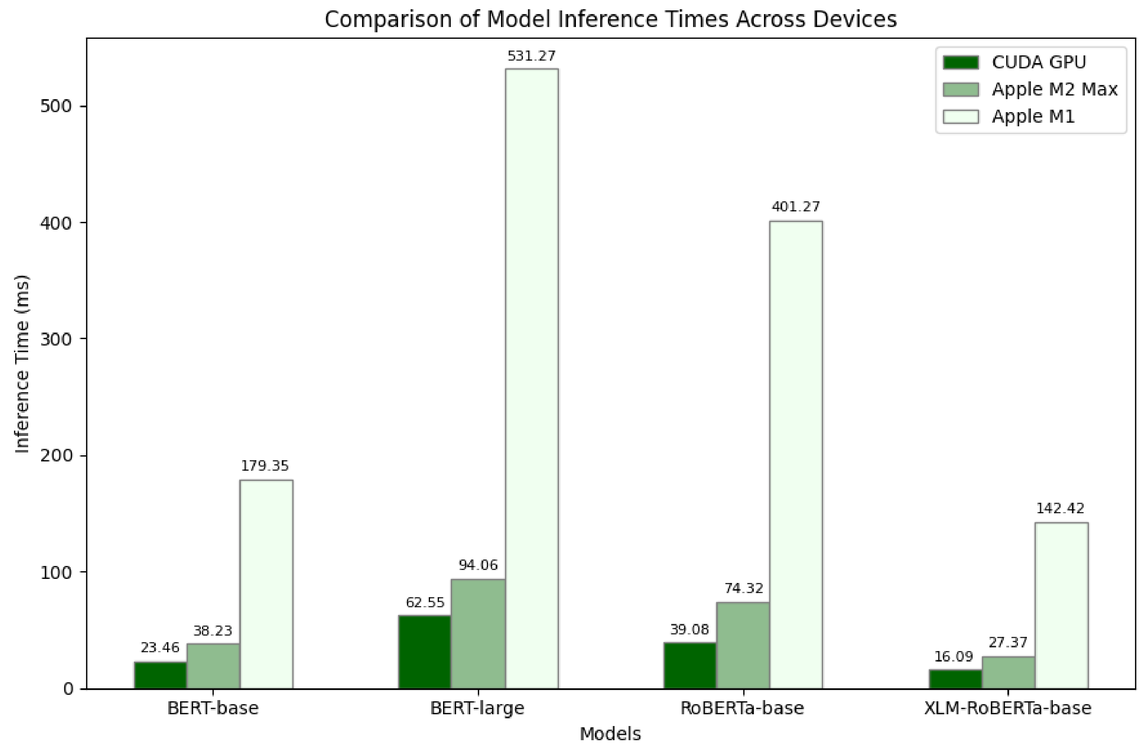

Figure 2. Comparison of average inference times (in milliseconds) for different models across CUDA GPU, Apple M1, and Apple M2 Max. Lower bars indicate better performance.

Key observations:

- The CUDA GPU outperformed both Apple Silicon devices
- M2 Max significantly narrowed the gap compared to the M1
- M2 Max's superior performance over M1 partly due to better hardware configuration (32GB vs. 8GB RAM). Consequently, more recent M-chips, like the M3 series and the forthcoming M4, are expected to deliver even better performance due to their advanced hardware configurations.

RESULTS: Influence of Input Length and Batch Size

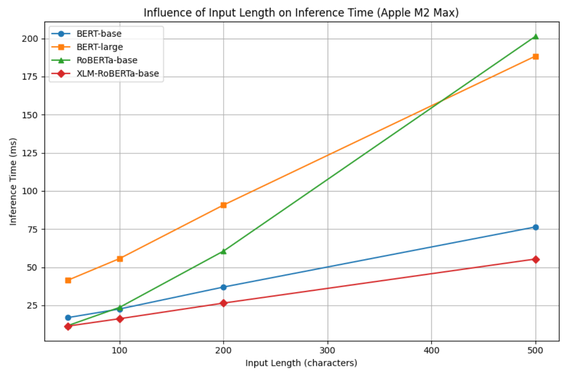

Figure 3. Influence of input length on inference times for different models on Apple M2 Max. The x-axis represents input lengths, and the y-axis shows inference times in milliseconds.

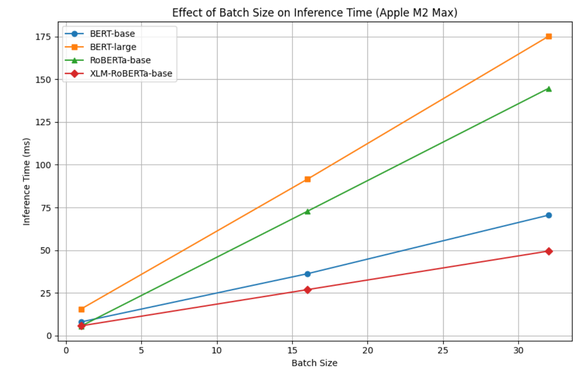

Figure 4. Effect of batch size on inference times for different models on Apple M2 Max. The x-axis represents batch sizes, and the y-axis shows inference times in milliseconds.

Input Length Impact:

Inference times increased with input length across all models, with varying rates of increase. On the M2 Max, BERT-base showed a relatively linear increase from 16.93ms (50 chars) to 76.40ms (500 chars), while RoBERTa-base exhibited a more pronounced increase from 11.65ms to 201.36ms.

Batch Size Impact:

All models showed increased inference times with larger batch sizes, but the scaling was not linear. For instance, on the M2 Max, BERT-base inference times increased from 8.02ms (batch size 1) to 70.48ms (batch size 32), an approximately 9x increase for a 32x batch size increase. This sublinear scaling suggests efficient parallel processing capabilities of the M2 Max.

CONCLUSION AND FUTURE WORKS

While CUDA GPUs remain the preferred choice for performance-critical applications, the combination of MLX and Apple Silicon devices presents a compelling alternative for many machine learning tasks, particularly for on-device inference and experimentation. Their accessibility, cost-effectiveness, and improving performance make them an increasingly attractive choice in the AI and machine learning landscape. This advancement contributes to the democratization of AI technologies, enabling broader access to powerful machine learning capabilities on consumer-grade hardware.

Future research should focus on optimizing performance for larger and more diverse models, as well as exploring the potential for on-device training of smaller models, further expanding the capabilities of machine learning on consumer devices.

We plan to expand this benchmark to include models of different modalities, we also intend to include include the latest Apple Silicon devices, such as the M3 and forthcoming M4 series, for a more comprehensive evaluation of MLX's capabilities

References

[1] Awni Hannun, Jagrit Digani, Angelos Katharopoulos, and Ronan Collobert. 2023. MLX: Efficient and flexible machine learning on Apple silicon. https://github.com/ml-explore.

[2] Connor Kenyon and Collin Capano. 2022. Apple Silicon Performance in Scientific Computing. arXiv:2211.00720 [cs.DC] https://arxiv.org/abs/2211.00720

oluwaseunaajayi@student.oauife.edu.ng