FusionBench: Analyzing Kernel Fusion in Vision Models

Kamaula Rowe[1], Angélica Aparecida Moreira[2], Tanvir Ahmed Khan[3]
[1]Princeton University, [2]Microsoft Research, [3]Columbia University

Introduction: From data centers to mobile devices, Machine Learning (ML) drives a variety of use cases including computer vision[1], natural language processing[2], and speech recognition[3]. ML benchmarks are crucial for evaluating the performance of ML models, software, and hardware. Established benchmarks from TorchBench[4] and HuggingFace[5] evaluate ML tasks but do not focus on kernel fusion optimizations. Kernel fusion combines consecutive operators into a single kernel, improving efficiency by minimizing memory access overhead and enhancing computational performance[6]. Despite its significance, to the best of our knowledge, there is a lack of benchmark suites dedicated to this technique, creating a gap in comparison and innovation. Reliable metrics are needed to understand, optimize, and automate kernel fusion, particularly for tensor operations like convolution. Without these benchmarks, the potential advantages and effects of kernel fusion remain underexplored, hindering progress in optimizing ML model performance and automation in ML compilers. In this work, we propose FusionBench, a novel benchmark suite to study the effectiveness of kernel fusion in improving performance of various ML workloads.

Methods: We selected representative workloads by using pre-trained models from the TorchBench suite and the HuggingFace community, focusing on their popularity in vision applications. Diverse model architectures were also taken into consideration, with the suite containing transformers, smaller convolution neural networks (CNNs), and larger CNNs. We compiled each model using PyTorch's TorchInductor compiler before inference, and the PyTorch profiler was used to evaluate data before and after the compiler's fusion optimizations.

Results: We developed a benchmark suite of nine image classification models as a baseline for kernel fusion evaluation. The execution times of each model's inference for a single image was recorded. We show these results in Figure 1. As Figure 1 shows, kernel fusion optimization decreased execution times for some models, while increasing it for others. The most common fusions performed within the suite are convolution + rectified linear unit (ReLU) and batch normalization + ReLU.

Our contributions include:

- Development of a benchmark suite focused on kernel fusion optimizations.
- A comprehensive evaluation of kernel fusion's impact on inference performance for pre-trained models.
- Insights into the most common kernel fusions and their effects on model performance.

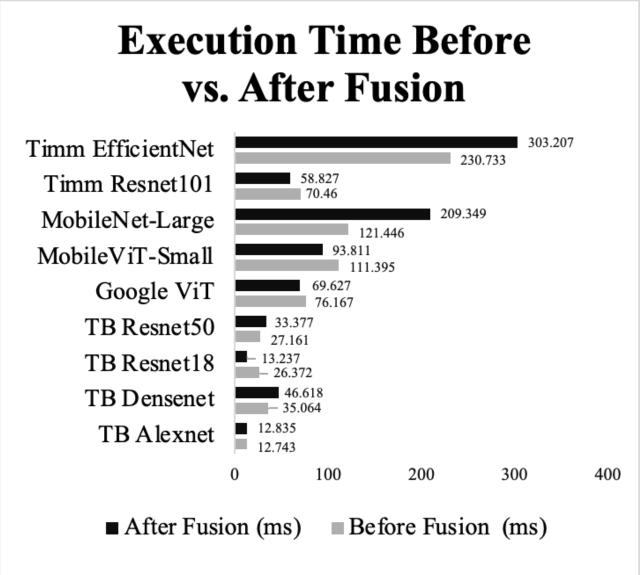

Figure 1. Execution times before and after compiler optimizations. As shown, fusion helps speed up some models while slowing down other models.

Conclusions: Our ultimate goal is to pinpoint areas for optimization within the TorchInductor compiler and to develop a new compiler strategy that can automate kernel fusion in machine learning applications. We plan to do this by investigating the reasons behind the observed discrepancies between the execution times by further analyzing how each model is compiled, as speedups for all models were expected. Analysis into memory usage before and after each model is compiled is also an area for future observation.

References:

1. Szeliski, R. (2022). Computer vision: algorithms and applications. Springer Nature.
2. Liddy, E. D. (2001). Natural language processing.
3. L. Deng and X. Li, "Machine Learning Paradigms for Speech Recognition: An Overview," in IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 5, pp. 1060-1089, May 2013, doi: 10.1109/TASL.2013.2244083.
4. Hao, Y., Zhao, X., Bao, B., Berard, D., Constable, W., Aziz, A., & Liu, X. (2023). Torchbench: Benchmarking pytorch with high api surface coverage. arXiv preprint arXiv:2304.14226.
5. Jain, S.M. (2022). Hugging Face. In: Introduction to Transformers for NLP. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-8844-3_4
6. W. Sun, A. Li, S. Stuijk and H. Corporaal, "How much can we gain from Tensor Kernel Fusion on GPUs?," in IEEE Access, doi: 10.1109/ACCESS.2024.3411473.
7. O'shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

Acknowledgements: We thank the Columbia University Summer Undergraduate Research Experience (SURE) Program sponsored by Amazon as Kamaula Rowe completed this work as her summer undergraduate research project.