EINNet: Optimizing Tensor Programs with Derivation-Based Transformations

Liyan Zheng, Haojie Wang, Jidong Zhai, Muyan Hu, Zixuan Ma, Tuowei Wang, and Shuhong Huang, Tsinghua University; Xupeng Miao, Carnegie Mellon University; Shizhi Tang and Kezhao Huang, Tsinghua University; Zhihao Jia, Carnegie Mellon University
Abstract and Introduction

• The paper introduces EINNET (Efficient Inference Network), a derivation-based optimizer for tensor programs, which are at the core of DNN computations.

• Traditional optimization techniques have limitations because they rely on a fixed set of predefined tensor operators, leading to restricted optimization possibilities.

• EINNET expands this by using general tensor algebra expressions, enabling a much larger optimization space and automatically creating new operators required by transformations.
Background and Motivation

The current tensor program optimization works at two levels: operator and graph. Operator-level optimization focuses on performance tuning for specific tensor operators, while graph-level optimization reorganizes DNN computations for efficiency.

However, both approaches are constrained to predefined operator representable (POR) transformations, which EINNET aims to transcend by exploring general tensor algebra transformations.
Key Contributions

• EINNET is distinguished by revealing operator computation semantics and applying derivation rules to tensor algebra expressions, allowing for the reorganization of computation into arbitrary tensor expressions.

• This system can potentially introduce novel program transformations and optimize beyond the capabilities of existing frameworks.
Addressing Optimization Challenges

• EINNET tackles three main challenges: discovering transformations between general expressions, converting expressions back to executable kernels (expression instantiation), and efficiently finding optimizing transformations in the vast space of general tensor algebra transformations.
Figure 5: The derivation process of the example in Figure 3(b), which transforms Conv with Matmul and eOperators
Methodology

- The paper details the derivation rules EINNET employs to transform tensor programs into optimized forms.
- These rules encompass intra-expression derivation.
- The approach combines traversal and summation notations, along with scope-based transformations to optimize computation.
Optimization

- EINNET optimizes tensor programs: i.e.) transforming convolution operations into matrix multiplications and fusing multiple operators into a single one for efficiency

\[
\begin{align*}
\sum_{c=0}^{C} \sum_{r=0}^{R} \sum_{k_0=0}^{K} \sum_{k_1=0}^{K} A[c, k_0] B[k_0, k_1] C[k_1, r] \\
\sum_{c=0}^{C} \sum_{r=0}^{R} \sum_{k_0=0}^{K} \sum_{k_1=0}^{K} A[c', k_1] B[k_1, k_2] C[k_0, r]
\end{align*}
\]

Figure 4: A tensor algebra expression example for two matrix multiplications $A \times B \times C$. The red box highlights a scope that instantiates the intermediate result of $A \times B$. 

- **Traversal notation**
- **Summation notation**
- **Scope**
EINNET has been implemented with over 23,000 lines of code in C++ and Python and has shown significant performance improvements over existing optimizers, with speedups of up to $2.72\times$ on certain hardware.
Overview of EINNET

- The optimizer translates an input tensor program into subprograms, translates these into tensor algebra expressions, applies derivation rules, and generates optimized subprograms.
Conclusion and Final Thoughts

- **Innovation in Optimization**: EINNET represents a significant advancement in tensor program optimization, pushing beyond the constraints of predefined operator representable (POR) transformations by leveraging general tensor algebra expressions.

- **Methodological Breakthrough**: Introduces a derivation-based mechanism that transforms tensor programs into optimized forms, utilizing a set of sophisticated derivation rules that ensure functional equivalence while enhancing performance.

- **Performance Enhancement**: Demonstrated substantial performance improvements over existing optimization frameworks, achieving up to 2.72× speedup.

- **Practical Impact**: EINNET's capabilities can be applied to a variety of real-world applications that utilize DNNs, potentially leading to efficiency gains in critical areas such as autonomous driving, speech recognition, etc.

- **Future Directions**:
  - Expand Optimization Techniques
  - Broader Hardware Compatibility
  - Integration with Existing Frameworks