Course Experiment

Triton Conv2D kernel

Overview

In this experiment, students will implement a 2D convolution operation using **Triton**, a Python-based framework for writing custom GPU/MLU kernels. The goal is to gain hands-on experience with low-level tensor computation, kernel programming, and performance optimization. Students will compare their Triton implementation with PyTorch's native `Conv2d` to validate correctness and measure performance.

Objectives

By completing this experiment, students will be able to:

1. Understand how to implement a basic 2D convolution kernel in Triton.
2. Learn how to write and launch Triton kernels for tensor computation.
3. Compare the output of a custom Triton kernel with PyTorch’s built-in `Conv2d` to ensure correctness.
4. Measure and analyze the performance difference between Triton and PyTorch implementations.

Environment

- **Hardware Platform:** DLP Cloud Platform environment.
- **Image**：mlu370_ubuntu22.04-for-student:v8_ds
- **Software Environment:** PyTorch 2.5, Triton 3.0.0

Implementation

Students are required to implement a custom 2D convolution using Triton by writing a kernel function that performs element-wise computation over the input and kernel tensors and applies the bias. A Python wrapper function should allocate the output tensor and launch the kernel with the proper grid configuration. The implementation is then validated by comparing the output against PyTorch's `Conv2d` and measuring the execution time to analyze performance differences.

Code

Complete the provided templeate `conv2d_template.py`

Evaluation Criteria

Grading will be based on the following aspects:

- **Correctness of Implementation**: Proper implementation of the Triton conv2d kernel and wrapper function, including correct memory indexing and bias addition. Outputs should match PyTorch's `Conv2d` .
- **Performance Analysis**: Ability to measure and interpret execution time for both Triton and PyTorch implementations, and discuss any observed speedup or bottlenecks. Higher performance and meaningful speedup will receive higher scores, and outstanding acceleration results may be awarded full grades.
- **Conceptual Understanding**: Clear demonstration of understanding of Triton kernel programming, grid configuration, and convolution computation.
- **Clarity of Report**: Well-organized presentation of the implementation process, validation results, and performance analysis.

327字