Learning Route Planning for 25 Spring

@Bao Zhuhan

GPU Operator Kernel Development, Performance Optimization & CUDA Programming

1. C/C++ and Operating System Principles (no specific focus)
2. Computer Architecture and Parallel Computing Concepts (the CMU-15-418 course)
3. Integration with Deep Learning?
    - Explore how to write custom CUDA Kernels to accelerate certain operators in deep learning, such as convolution, normalization, or other common operations.
    - Attempt to integrate custom CUDA Kernels in PyTorch, and understand how the framework calls acceleration code at the lower level.

Deep Learning Algorithm Principles and PyTorch Direction

1. Get started with PyTorch and complete the official tutorial examples.
2. Begin intermediate-level projects, such as image classification or simple object detection tasks; try customizing model layers and parameter tuning.
3. Study cutting-edge research papers, design a comprehensive project (国创), combining data preprocessing, model training, and model optimization.

Learning Path

- **First 4 weeks**:
    - **Main focus**: Learn deep learning fundamental theories; Get started with PyTorch, complete official tutorial examples.
    - **Secondary focus**: Use spare time to complete CUDA programming basics, continue with the CMU-15-418 course

- **Weeks 5-8**:
  - **Main focus**: Tasks such as image classification or simple object detection; Try customizing model layers and parameter tuning.
  - **Secondary focus**: Write simple CUDA examples, attempt to analyze performance bottlenecks, gradually understand memory optimization strategies.
- **Weeks 9-12**:
  - **Main focus**: Study cutting-edge research papers in depth, design a comprehensive project (implement and refine 国创), combining data preprocessing, model training and model optimization.
  - **Secondary focus**: Try integrating CUDA Kernels into PyTorch projects, optimize key operators, use performance analysis tools for tuning.