week-21

JX-Ma

2025/1/4

1 INTORDUCE

This week's work is as follows:
This week, I mainly tried index precomputation and changed the data layout on im2win.
My previous idea was to record the first address of each thread handling elements plus the first address of each thread handling data each time. Firstly, recording the first address of each thread handling element requires a large amount of constant memory, for example, conv1 requires a (256x55x55) large int array in the N direction to record. And for each thread, its first address for handling data only needs to be calculated once. Therefore, I only record the first address of each thread's data transfer into the constant memory.
For transferring data from global memory to shared memory through threads, my approach is to have one or more threads responsible for moving a convolution window. For a convolution kernel, one or more threads are responsible for moving a batch, but for an input tensor, each thread is responsible for moving an element convolution window.
Due to the read-only nature of constant memory, it is necessary to calculate the index on the CPU and write it into constant memory using the CUDACPySymbol function for GPU use.
Due to the possibility of discontinuity in element handling in the input tensor convolution window, I added padding filling on top of the original data layout.
Later, I found that in the later layers of convolution, filling too much data caused many invalid calculations. So I modified the original data layout to NHWC and tested TFLOPS.
Experimental results: There is a significant improvement on conv1-conv3, approaching the implicit matrix multiplication in CuDNN. conv7 surpasses the implicit matrix multiplication in CuDNN, but the performance is not as good in other layers.