Anatomy Of High-Performance Deep Learning Convolutions On SIMD Architectures

Hao Deng

January 2024

1 Reading Reflection

This paper mainly proposes a convolution solution to achieve the best possible single node performance on CPUs. This paper elaborates how to attain this goal using JIT-ing strategy.

Notes: JIT(Just-in-time) Compilation is a type of compilation during the execution of a program (i.e. runtime). Generally, A JIT has access to dynamic runtime information whereas a standard compiler doesn't and can make better optimizations like inlining functions that are used frequently.

This paper talks about the following aspect:

1. Derive and define fast direct convolution
2. Combine JIT and layer/execution graph strategy to deal with the demand of a large number of required kernels. Improve the locality.
3. Evaluate the CNN training performance on recent CPU architectures on kernel and multi-node level.

The most important part of this paper is a series of optimization it implements, including:

1. Vectorization and register blocking
2. Cache blocking and loop ordering
3. Generating microkernel in JIT style.
4. Prefetching: Two levels–L1 and L2
5. Parallelization
6. Layer fusion: Normally, non-convolution layers have low operational intensity-¿bandwidth bound. Layer fusion decomposes them such than they operate on sub-tensors.
7. Kernel Stream: Sometimes, the a tensor's spatial dimensions (i.e. height and width) can not be diveded by register blocking factors. Instead of reducing the blocking factors and sacrificing the performance, creating another microkernel would be more beneficial.
8. Backward propagation implementation
9. Weight gradient update implementation
10. Reduced Precision: It's the current trend to increase the performance by reducing the precision of data.

This paper evaluates its algorithm from two aspects:

1. Kernel efficiency for various topologies
2. End-to-end fully-integrated CNN training performance

I think this can be a good example when we are thinking about evaluate our models.