Understanding the Performance of Small Direct Convolution Operations for CNN on Intel Architecture

Alexander Heinecke1, Evangelos Georganas1, Kunal Banerjee2, Dhiraj Kalamkar2, Narayanay Surdaram1, Anand Venkat3, Greg Henry1, Hans Pabet1
1Intel Corporation, Intel Labs, Mission College Boulevard 2200, Santa Clara 95054, CA, USA
2Intel Technology India Private Limited, Intel Labs, #23-500, Devarahalli Hall, Outer Ring Road, Bangalore 560011, India
3Intel Corporation, Software and Services Group, 2111 NE 28th Avenue, Hillsboro 97124, OR, USA
4Intel Siemens AG, Software and Services Group, Badenwittensstrasse 540, 8046 Zurich, Switzerland

Introduction

LIBXSMM is a library targeting Intel Architecture for small, dense or sparse matrix multiplications, and small convolutions. LIBXSMM is available as free software at https://github.com/hfp/libxsmm.

- Work targets Convolutional Neural Networks (CNNs), and other deep neural networks.
- Convolution operators are thread-library agnostic (fphrases, OpenMP, C+ threads, Cilk, etc.)
- Direct convolution and Winograd transformation implemented.
- Convolution layer is integrated into TensorFlow.
- Optimized for x86 (Xeon Phi and Xeon server).

This work leverages LIBXSMM's infrastructure to generate executable code Just-In-Time (JIT) by assembling the instructions in-memory. For reproducible results and for general use, we show how the JIT optimizations integrate with a high-level domain-specific language such as TensorFlow.

Performance

We have evaluated the performance in the context of standalone convolution operators using direct convolutions and Winograd for 3x3 convolutions. We selected the most important layer operations from Alexnet, Resnet-50 and Google's Inception-v3 topologies. The later two are the most modern topologies in production.

Our test platform is a single-socket Intel Xeon Platinum 8180 (SKX) with 28 cores and a Intel Xeon Phi-7250 (KNL) with 68 cores. On both platforms Intel Turbo Technology was enabled. On the Xeon Platinum SGEMM runs at 3.2 TFLOPS where as the Xeon Phi-7250 delivers 4.6 TFLOPS for SGEMM. We also present measurements on a Knights Mill processor (skipped for submission). All measurements are based on Version 1.9 of LIBXSMM, https://github.com/hfp/libxsmm/releases.

Implementation Details

Kernel Streams
- Convolutions consist of seven nested loops, each one corresponding to the parameters N, C, K, H, W, R, S. In the innermost loop we call the JIT-ed kernel.
- The kernel takes six arguments: 3 addresses for the input, weight and output blocks to be convoluted in the current iteration and 3 addresses for the input, weight and output blocks to be prefetched for following iteration.

Winograd
- The implementation is split into 4 kernels: input transform, weight transform, batched GEMM, output transform.
- For best performance we implemented a JIT-based batched GEMM.

Integration into Tensorflow
- LIBXSMM leverages Tensorflow's thread pool and offers native support for TF's data format (which eliminates costly data transformations).
- Current integration speeds up Tensorflow by 1.5x running Inception-v3 model on an Intel Xeon Platinum CPU.

Summary/Outlook

- Achieved optimal performance for wide range of convolution operators on latest Intel Xeon Platinum (code-named Skylake) and Intel Xeon Phi (code-named Knights Landing) processors.
- Integrated into Tensorflow and leverages Tensorflow's execution model.
- Already offers support for Intel's Machine Learning oriented chip code-named Knights Mill.
- Freely available and includes examples for usage.

Current Research:
- Adding a runtime auto-tuning component for both dispatching and micro-kernel composition.
- Providing LSTM/RNN modules based on small blocked GEMMs.

References