FFT-based convolutional neural networks for wide-SIMD multi-core CPUs

Convolutional Neural Network

For many, including in the tensor-contraction domain, fast computation of 2D convolution and related functions is at the core of the neural network. In particular, for convolutional neural networks, the forward pass is dominated by a series of 2D generic convolutions. Many strategies have been developed to efficiently compute these convolutions on Intel Skylake architectures.

**Convolutional Neural Network Execution Patterns:**

- **Forward Convolution:**
  - Data movement across rows
  - Redundancy in the input data
  - Need for efficient data movement and storage

- **Batch FFT:**
  - Efficient computation of FFTs
  - Reduction in computational complexity

- **Performance:**
  - Forward Convolution
  - Batch FFT

**Contributions:**

- A novel implementation strategy for 2D FFT on Intel Skylake architectures
- A highly tuned component that includes batched 2D FFT, CGEMM-like matrix-matrix multiplication, and work-stealing thread scheduler

**Acknowledgements:**

This material is based upon work supported by the U.S. National Science Foundation (NSF) under Grant No. 1617111.

**Footnotes:**

1. For an implementation on modern hardware, the batched FFT-based convolution can be significantly more efficient.

**Performance:**

- **Forward Convolution**
  - Speedup over MKL on the sizes relevant for convnets
  - Achieved by aggressively merging butterflies with other transformations
  - Speedup of up to 51 on VGG-A and 31 on VGG-B

**Options C and D:**

- Option C: No shuffles needed
- Option D: Requires particularly many shuffles for real-to-complex FFT
- Option C: Low register pressure: can easily fit 3 16-element rows into registers
- Option D: FFT butterflies require shuffles

**Complex SIMD:**

- **AS vs AOA:**
  - AOS PERM and AOS PERM formats to cut the storage in half
  - Uses PERM and AOS PERM formats to cut the storage in half

**Complex FFT Layout:**

- Uses structured data movement to reduce the computational overhead
- Reduces the number of floating-point operations required

**Performance:**

- FFT-based convolutional neural networks for wide-SIMD multi-core CPUs
- Fourier transforms for real and complex signals
- High performance on modern hardware

**FFT Across Rows:**

- Loads parts of different rows into SIMD registers
- FFT Within Rows loads a whole image row into SIMD registers
- High performance on modern hardware

**FFT Within Rows vs Across Rows:**

- FFT fabrics handle a whole image row into SIMD registers
- FFT Across Rows loads parts of different rows into SIMD registers

**FFT butterflies require shuffles:**

- In option C, the shuffles are grouped inside transposition, creating a bottleneck in pipeline.
- Option D all shuffles are in transposition, where they can be interleaved with arithmetic operations.

**Note:**

- The performance of FFT-based convolutional neural networks is significantly improved by optimizing data movement and arithmetic operations.
- The implementation strategy is effective on modern hardware, achieving high speedups over CPU-based implementations.