Espresso: Efficient Forward Propagation for Binary Deep Neural Networks

Fabrizio Pedersoli, George Tzanetakis, Andrea Tagliasacchi

University of Victoria

Presenter: Eamon Collins

https://qdata.github.io/deep2Read

Overview

Motivation

- Make binary networks even faster
  - Leverage bit-packing and bit-wise operations
- End-to-end software framework
  - Easy to apply to existing networks

Prior Work

- Quantization and specifically binarized networks well-studied
- BinaryNet

  - Includes a binary-optimized version of matrix multiplication, 7x faster than baseline
  - Replace floating-point multiply and add with XNOR and bitcount

Contributions

- Bit-packing in network layers
- Memory layout optimization
- Custom CUDA kernels optimized for binary weights and activations

Binarization and Bit-Packing

Binarization

A BDNN is composed of a sequence of $k = 1, \ldots, L$ layers whose weights W_k^b and activations a_k^b are binarized to the values -1, +1.

$$x^b = sign(x) = \begin{cases} +1 & x \geq 0 \\ -1 & otherwise \end{cases}$$

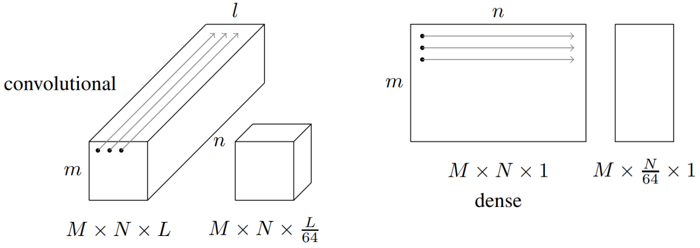
Bit-Packing

Store weights in a 64-bit word. We can compute a dot-product of 64 element vectors by using just one XNOR and one bit-count. Assuming binary vectors vectors $a, b \in B^{1 \times N}$ where N is a multiple of 64,

$$a \cdot b \equiv ((\sum_{i=1}^{N/64} \text{bitcount}(\text{XNOR}(a_i, b_i))) \ll 1) - N$$

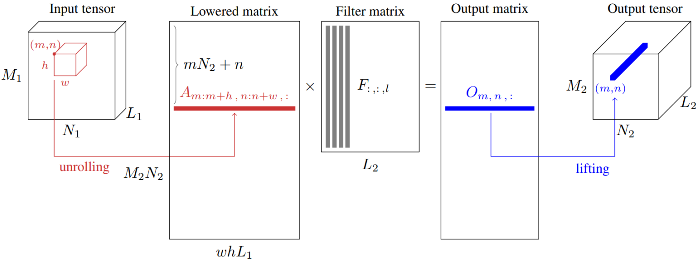
Memory Layout

- For convolutional layers, bit-packing is performed along the *l* dimension
- Otherwise, along the *n* dimension
- Allows for good cache locality



Memory Layout

- Unroll: transform a tensor into a matrix where each row is formed by unrolling the tensor data contained in each convolution sliding volume
- Otherwise, along the *n* dimension
- Allows for good cache locality



Experiments

Baselines

1. BinaryNet
2. Optimized BDNN implemented in the Intel Nervana Neon framework
3. Espresso GPU (cache locality but no XNOR dot product)
4. Espresso GPU^{opt}

Tasks

1. Matrix multiplication
2. MLP on MNIST
3. CNN on CIFAR-10

Results

Table 1: Averaged time of binary optimized matrix multiplication.

BinaryNet	Espresso *GPU*^{opt} 32-bit	Espresso *GPU*^{opt} 64-bit
88 ms	16 ms (5.5×)	11 ms (8×)

Table 2: Average prediction time of the BMLP.

BinaryNet	Nervana/Neon	Espresso *CPU*	Espresso *GPU*	Espresso *GPU*^{opt}
18 ms	17 ms	37.4 ms	3.2 ms (5.6×)	0.26 ms (68×)

Table 3: Average prediction time of the BCNN.

Espresso *CPU*	Espresso *GPU*	Espresso *GPU*^{opt}
85.2 ms	5.2 ms (16×)	1.0 ms (85×)

Takeaways

- Demonstrate great speedups over BinaryNet and Nervana for MLP
  - Difficult to understand where the unoptimized GPU implementation gains the speedup on MLP
- Significant speedups from both memory layout and bit-wise operations
- No binarized CNN implementation to compare to, would likely do well though.
  - No mention of XNOR-net?

References

. Courbariaux, I. Hubara

Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or 1

In International Conference on Learning Representations, 2017.

atthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David.

Training deep neural networks with low precision multiplications.

arXiv preprint arXiv:1412.7024, 2014.