Binary Weight CNN Hardware Accelerator for ULP Computer Vision

P. Jokic, S. Emery

Convolutional neural networks are the state-of-the-art algorithms for object detection and classification in many computer vision applications. The high accuracy comes at the price of deeper architectures, resulting in large networks and high associated computational requirements. This limits their deployment on wearables, IoT devices and other power-restricted platforms. Recent advances in quantized neural networks have achieved comparable performance at a significantly lower memory footprint and reduced computational complexity. We present a low power convolutional neural network hardware accelerator featuring efficient exploitation of binary weights and elaborate on the benefits of quantized weight networks.

The key components of today's state of the art object detectors are convolutional neural networks (CNN). Applications range from high-speed obstacle detection in autonomous driving to low-power face detection in smart wearables. While cars are already designed to supply large amounts of power to electronic sub-systems, mobile devices are limited by their small batteries, requiring every single sub-system to consume as little power as possible. To enable meaningful computer vision applications on such low power devices, efficient CNN processing is needed. This work presents a binary weight CNN hardware accelerator that efficiently computes binary weight neural networks, which minimize the memory footprint and enable low-power processing of embedded computer vision applications.

Computing a CNN is a highly data intensive task, dominated by millions of multiply-accumulate (MAC) operations, which are needed to convolve input activations with kernels consisting of learned weights. This process is repeated for each network layer, using the outputs of the previous layer as inputs for the following one. Recent improvements in the performance of neural networks were largely a consequence of the transition to deeper networks, leading to a data bottleneck which manifests itself in a large memory footprint, overloaded memory interfaces and a high-power consumption. Reducing the vast amount of data to be processed and simplifying the involved computations therefore became a hot topic in the field of embedded machine learning. One proposed approach is data quantization, aiming for smaller data entries due to compressed data and reduced precision computational elements. Different levels of quantization applied on activations and/or weights have been proposed. Even fully binary networks [1] (BNN), with activations and weights being quantized to 1 bit (representing -1 or 1) have been shown to be viable options, though at the cost of a reduced accuracy.

This work is using binary weight CNNs, which are a trade-off between accurate full-precision CNNs and BNNs, featuring binary weights but standard-precision activations. From a computational point of view, binary weights are very practical for the convolution operation: the summed multiplications of input activations and weights (+/- 1) are simplified to additions. This helps saving power as multipliers can consume more than 50x more power and 25x more area than adders of the same precision [2]. Compared to a non-quantized 16-bit CNN, binary weights can reduce the memory of the weight parameters by 16x, reducing memory leakage and power for accessing data. This is especially beneficial for deep networks, where the memory size for storing weights can be comparable to the activation memory size. We evaluated the memory savings on a real-world 9-layer CNN for face-detection that was ran on our accelerator: the overall memory footprint was reduced by almost 35% when changing from 16 bit to the implemented binary weights.

The implemented CNN accelerator is a standalone system that can directly interface a connected memory, allowing to take over the complete CNN processing from a microcontroller. It consists of 5 main blocks, as shown in Figure 1:

- Control: reads layer settings from memory, configures other blocks, requests addresses of activations and parameters.
- Address generator: computes memory addresses of data.
- Memory interface: buffers requests and interfaces memory.
- Data pre-processing: extracts and sorts data from memory words, buffers operands to keep all parallel MAC units active and writes results back into memory words.
- MAC processing: multiple parallel units performing the MAC operation with binary weights, pooling and output activation.

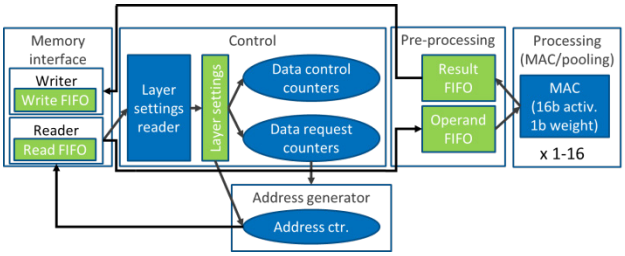

Figure 1: CNN accelerator block diagram.

The number of parallel MAC processing units can be chosen using a generic design parameter, achieving a peak performance of 360 MOPS with a single memory interface and two MAC units. An implementation supporting higher memory bandwidths using multiple memory ports is under development. All network architecture settings, such as layer width or channel depth, can be changed at runtime by configuring the layer settings region in the memory. This allows the accelerator to process any kind of CNN that fits in the memory, enabling different applications to be processed with the same chip. Other fields of use include signal processing, image classification or compression. With an area of 128 kGE (gate equivalents, for a version with 2 parallel MAC units), the accelerator is light-weight and can be easily used to offload data-intensive CNN computations from a microcontroller. The estimated power consumption in a 22 nm FDX process is 2 mW at 180 MHz.

[1] M. Courbariaux, *et al.*, "BinaryConnect: Training Deep Neural Networks with Binary Weights During Propagations," NIPS (2015).

[2] M. Horowitz, "Computing's energy problem (and what we can do about it)," ISSCC (2014).