Nallatech
a molex company

White Paper

FPGA Acceleration of Binary Neural Networks

Deep Learning

Until only a decade ago, Artificial Intelligence resided almost exclusively within the realm of academia, research institutes and science fiction. The relatively recent realization that Deep Learning techniques could be applied practically and economically, at scale, to solve real-world application problems has resulted in a vibrant eco-system of market players.

Now, almost every application area is in some way benefiting from Deep Learning – the leveraging of Artificial Neural Networks to learn from vast volumes of data to efficiently execute specific functions. From this field of neural network research and innovation, Convolutional Neural Networks (CNNs) have emerged as a popular deep learning technique for solving image classification and object recognition problems. CNNs exploit spatial correlations within the image sets by using convolution operations. CNNs are generally regarded as the neural network of choice – especially for low-power applications because they have fewer weights and are easier to train compared to fully connected networks which demand more resources.

Neural Networks

One approach to reduce the silicon count and therefore power required to execute a high performance neural network is to reduce the dynamic range of floating-point calculations. Using 16-bit floating-point arithmetic instead of 32 bits has shown to only slightly impact the accuracy of image classification. Furthermore, depending upon the network, the accuracy of the calculation can be reduced even further to fixed point or even single bits. This trend of improving overall efficiency through implementation of reduced calculation accuracy has led to the use of binary weights i.e. weights and input activations that are binarized with only two values: +1 and -1. This new variant is known as a Binary Neural Network (BNN). It reduces all fixed-point multiplication operations in the convolutional layers and fully connected layers to 1-bit XNOR operations.

Flexible FPGAs

Established classes of conventional computing technologies have attempted to evolve at pace to cater for this dynamic market. NVIDIA, for instance, has not only adapted the underlying GPU architecture and tools, but also their product strategy and value proposition. GP-GPUs, previously marketed as the ultimate double precision floating-point engines for graphics and demanding HPC applications are now being re-positioned for the Deep Learning CNN market where half-precision arithmetic support is critical for success.

Google, one of the strongest proponents of AI, has created its own dedicated hardware architecture, the Tensor Processing Unit (TPU), which is tightly coupled with their Machine Learning framework, TensorFlow. Other industry leaders, including hyperscale innovator Microsoft, have selected Field Programmable Gate Arrays (FPGAs) for their "Brainwave" AI architecture – a pipeline of persistent neural networks that promises to deliver real-time results. This choice is no doubt linked to the confidence they gained from the highly successful (and market disrupting) use of Intel-based Arria-10 FPGAs for Bing search indexing.

FPGA Acceleration of Binary Neural Networks

This white paper explains why FPGAs are uniquely positioned to address the dynamic roadmap requirements of neural networks of all bit ranges – in particular, BNNs

Binary Neural Networks

Processing convolutions within CNN networks requires many millions of coefficients to be stored and processed. Traditionally, each of these coefficients are stored in a full single precision representation. Research has demonstrated that coefficients can be reduced to half precision without any material change to the overall accuracy while reducing storage capacity and memory bandwidth. More significantly, this approach also shorten the training and inference time. Most of the pre-trained CNN models available today use partial reduced precision.

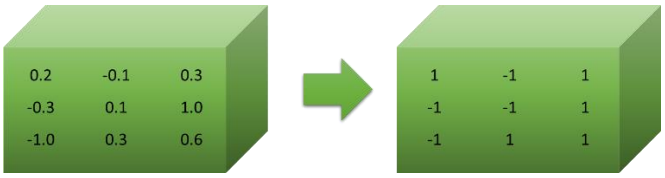

Figure 1 : Converting weights to binary (mean = 0.12)

By using a different approach to the training of these coefficients the bit accuracy can be reduced to a single bit, plus a scaling factor 1. During training, the floating-point coefficients are converted to binarized values and scaling a factor by averaging all output feature coefficients and subtracting this average from the original value to produce a result that is either positive or negative, represented as either 1,0 in binary notation (
Figure 1). The output of the convolution is then multiplied by the mean.

FPGA Optimizations

Firstly, binarization of the weights reduces the external memory bandwidth and storage requirements by a factor of 32. The FPGA fabric can take advantage of this binarization as each internal memory block can be configured to have a port width ranging from 1 to 32 bits. Hence, the internal FPGA resource for storage of weights is significantly reduced, providing more space for parallelization of tasks.

The binarization of the network also allows the CNN convolutions to be represented as a series of additions or subtractions of the input activations. If the weight is binary 0 the input is subtracted from the result, if the weight is binary 1 it is added to the result. Each logic element in an FPGA has addition carry chain logic that can efficiently perform integer additions of virtually any bit length. Utilizing these components efficiently allows a single FPGA device to perform tens of thousands of parallel additions. To do so the floating-point input activations must be converted to fixed precision. Given the flexibility of the FPGA fabric, we can tune the number of bits used by the fixed additions to meet the requirement of the CNN. Analysis of the dynamic range of activations in various CNNs shows that only a handful of bits, typically 8, are required to maintain an accuracy to within 1% of a floating-point equivalent design. The number of bits can be increased if more accuracy is required.

Converting to fixed point for the convolution and removing the need for multiplications via binarization dramatically reduces the logic resources required within the FPGA. It this then possible to perform significantly more processing in the same FPGA compared to a single precision or half precision implementation.

Deep Learning models are becoming deeper by adding more and more convolution layers. Having the capability to stack all these layers into a single FPGA device is critical to achieving the best performance per watt for a given cost while retaining the lowest possible latency.

FPGA Implementation

The Intel FPGA OpenCL framework was used to create the CNNs described in this paper. To optimize the design further, the Nallatech research center developed IP libraries for the binary convolution and other bit manipulation operations. This provides a powerful mix programmability and efficiency.

[1] https://pjreddie.com/media/files/papers/xnor.pdf

FPGA Acceleration of Binary Neural Networks

Layer(s)	Size	Filter size	No Filters
CONV x2	416x416	3x3 & 1x1	32,64
CONV x 3	208x208	3x3 & 1x1	64,128
CONV x 5	104x104	3x3 & 1x1	64,128
CONV x 17	52x52	3x3 & 1x1	128,256
CONV x 17	26x26	3x3 & 1x1	256,512
CONV x 15	13x13	3x3 & 1x1	512x1024
Up-sample & route	26x26	-	256
CONV x7	26x26	3x3 & 1x1	256,512
Up-sample & route	52x52	-	128
CONV x7	52x52	3x3 & 1x1	128,256

Table 1 : Approximate Yolo V3 layers

The network targeted for this white paper was the Yolo v3 network (Table 1). This network consists largely of convolution layers and therefore the FPGA has been optimized to be as efficient at convolutions as possible.

To achieve this, the design uses a HDL block of code to perform the integer accumulations required for binary networks, making for an extremely efficient implementation.

IP	ALUTs	Registers	Equivalent FP operations
32x32 8Bit Binary Convolution	17953 (2.1%)	10239 (0.6%)	2048 (67%)

Table 2 : Resource requirements of BNN IP (% Arria 10 GX 1150)

Table 2 lists resource requirements for the accumulation of the 8-bit activation data when using binary weights. This is equivalent to 2048 floating-point operations, but only requires 2% of the device. Note, there is extra resource required by the FPGA to restructure the data (see Table 3), so it can be processed this way, however it does illustrate the dramatic reduction in resources that can be achieved versus a floating-point implementation.

The FPGA is also required to process the other layers of Yolo v3 to minimize the data copied over the PCIe interface. These layers require much less processing and therefore less of the FPGA resource is allocated to these tasks. In order for the network to train correctly, it was necessary for activation layers to be processed with single precision accuracy. Therefore, all layers other than the convolution are calculated at single precision accuracy.

The final convolution layer is also calculated in single precision to improve training and is processed on the host CPU. Table 3 details the resources required by the OpenCL kernels including all conversions from float to 8-bit inputs, the scaling of the output data and final floating-point accumulation.

Kernel	ALUTs	Registers	M20K memories
Convolution (Binary Weights)	109763 (15%)	94447 (6%)	560 (24%)
Coefficient Controller	18478 (2.1%)	22599 (2%)	172 (7%)
Activation Layer	8098 (1%)	10488(1%)	80(3%)
Route Layer	4140 (1%)	7495 (1%)	59 (2%)
Shortcut Layer	14911 (2%)	17998 (1%)	119 (5%)
Up-sample Layer	7522 (1%)	9728(1%)	59(2%)

Table 3 : Resource requirements for full Yolo v3 CNN kernel (% Arria 10 GX 1150)

FPGA Accelerator Platforms

The FPGA device targeted in this whitepaper is an Intel-based Arria-10. It is a mid-range FPGA fully supported within the Intel OpenCL Software Development Kit (SDK). Nallatech delivers this flexible, energy-efficient accelerator in the form of either an add-in PCIe card or integrated rackmount server. Applications developed in OpenCL are mapped onto the FPGA fabric using Nallatech's Board Support Package (BSP) enabling customers (predominantly software rather than hardware focused) to remain at a higher level of abstraction than is typically the case with FPGA technology.

Nallatech's flagship "520" accelerator card shown below features Intel's new Stratix-10 FPGA. It is a PCIe add-in card compatible with server platforms supporting GPU-class accelerators. Ideal for scaling Deep Learning platforms cost effectively.

FPGA Acceleration of Binary Neural Networks

Performance

Each convolution block performs 2048 operations per clock cycle or ~0.5 TOPS per second for a typical Arria 10 device. 4 such kernels allow Yolo v3 to be run at a frame rate of ~8 frames sec for a power consumption of 35 Watts. This is equivalent to 57 GOPS/Watt.

XNOR Networks

It is possible to further reduce compute and storage requirements of CNNs by moving to a full XNOR network. Here both the weights and activations are represented as binary inputs. In this case a convolution is represented as a simple bitwise XNOR calculation, plus some bit counting logic. This is equivalent to the binary version described earlier except that activations are now only a single bit wide.

Speed-up of such networks is estimated at 2 orders of magnitude when running on FPGA. This disruptive performance improvement enables having multiple real-time inferences running in parallel on power efficient devices. XNOR networks require a different approach to training, where activations on the forward pass are converted to binary and a scaling factor.

Whereas binary networks show little degradation in accuracy, XNOR networks show 10-20%[2] difference to a floating-point equivalent. However, this is using CNNs not designed specifically of XNOR calculations. As research into this area increases, it's likely the industry will see new models designed with XNOR network in mind, that will provide a level of accuracy close to the best CNNs, while benefiting from the tremendous efficiency of this new approach.

Conclusion

This whitepaper has demonstrated that significant bit reductions can be achieved without adversely impacting the quality of application results. Binary Neural Networks (BNNs), a natural fit for the properties of the FPGA, can be up to thirty times smaller than classic CNNs - delivering a range of benefits including reductions in silicon usage, memory bandwidth, power consumption and clock speed.

Given their recognized strength for efficiently implementing fixed point computations, FPGAs are uniquely positioned to address the needs of BNNs. The inherent architecture flexibility of the FPGA empowers Deep Learning innovators and offers a fast-track deployment option for any new disruptive techniques that emerge. XNOR networks are predicted to deliver major improvements in image recognition for a range of cloud, edge and embedded applications.

Nallatech, a Molex company has over 25 years of FPGA expertize and is recognized as the market leader in FPGA platforms and tools. Nallatech's complimentary design services allow customers to successfully port, optimize, benchmark and deploy FPGA-based Deep Learning solutions cost-effectively and with minimal risk.

Please visit www.nallatech.com or email contact@nallatech.com for further information.

This work has been partly developed as part of the OPERA project to provide offloading support for low powered traffic monitoring systems: www.operaproject.eu

[2] https://pjreddie.com/media/files/papers/xnor.pdf