A Case for Dynamic Activation Quantization in CNNs

Karl Taht
School of Computing
University of Utah
Salt Lake City, USA
taht@cs.utah.edu

Surya Narayanan
School of Computing
University of Utah
Salt Lake City, USA
surya@cs.utah.edu

Rajeev Balasubramonian
School of Computing
University of Utah
Salt Lake City, USA
rajeev@cs.utah.edu

***Abstract*—It is a well-established fact that CNNs are robust enough to tolerate low precision computations without any significant loss in accuracy. There have been works that exploit this fact, and try to allocate different precision for different layers (for both weights and activations), depending on the importance of a layer's precision in dictating the prediction accuracy. In all these works, the layer-wise precision of weights and activations is decided for a network by performing an offline design space exploration as well as retraining of weights. While these approaches show significant energy improvements, they make global decisions for precision requirements. In this project, we try to answer the question "*Can we vary the inter- and intra-layer bit-precision based on the region-wise importance of the individual input?*".**

The intuition behind this is that for a particular image, there might be regions that can be considered as background or unimportant for the network to make its final prediction. As these inputs propagate through the network, the regions of less importance in the same feature map can tolerate lower precision. Using metrics such as entropy, color gradient, and points of interest, we argue that a region of an image can be labeled important or unimportant, thus enabling lower precision for unimportant pixels. We show that per-input activation quantization can reduce computational energy up to 33.5% or 42.0% while maintaining original Top-1 and Top-5 accuracies respectively.

***Keywords*-Machine Learning; Neuromorphic Architectures; Neural networks**

I. Introduction

Neural networks have gained recent popularity for many algorithms because of their flexibility to solve large classes of problems. This has spawned an advent of hardware accelerators specifically catered to these problems. While different techniques address data movement, heterogeneous requirements and more, most of these accelerators are still built on digital multiplication units to perform basic operations. Two high level solutions have gained recent popularity to address the fundamental computational requirements of these networks: reduced bit precision and analog computation.

While earlier studies with neural networks used 32-bit floating point numbers, 16-bit or even 8-bit fixed point arithmetic is becoming the norm. Regardless of hardware, the physical bit-width lowers the storage requirement. However, if underlying hardware supports variable precision, data movement and computation can be optimized as well. Today, mainstream support for 16-bit float point is widely available [1], and fixed point is becoming increasing popular, but not yet consumer available [2].

Futhermore, researchers are exploring reducing precision even further, and allowing for non-traditional bit-widths. Judd et al. demonstrate weights can be further quantized per-layer, but do so with a brute-force approach [3]. Park et al. propose a solution to reduce the brute force search to a guided search, and separately quantize activations and weights [4]. However, they limit their quantization search to the full network, rather than per-layer optimization. Despite this, they are able to maintain near full precision accuracy with just 5-bit weights and 6-bit quantizations.

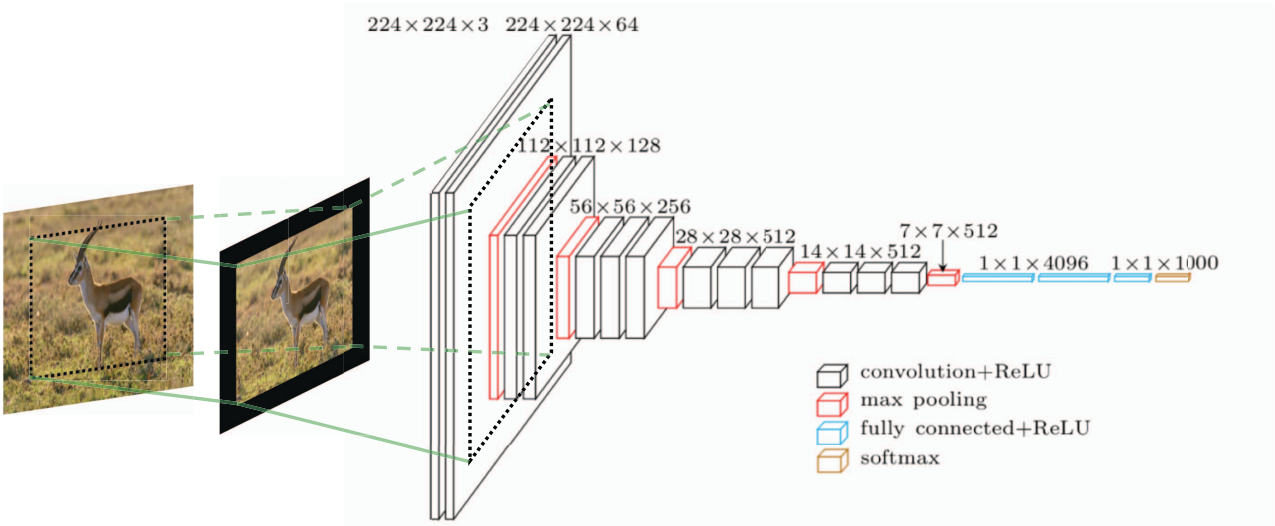
Orthogonally, architects are exploiting new technology to fundamentally change arithmetic circuits. Moshovos et. al propose new digital units which not only scale with bit-width, but also avoid ineffectual neuron bits [5]. Kull et al. showed that HP's memristive technology is capable of being configured as a highly parallel multiply-and-accumulate (MAC) circuit, providing orders of magnitude improvement over traditional CMOS [6]. The crossbar network leverages basic principles of circuits to perform multiplications and additions in the analog domain, and an analog-to-digital converter to bring results back into the digital domain. Shafiee et. al. leverage this circuit in *ISAAC* [7], along with 16-bit fixed point computation.

II. Proposal

We propose *AQuA*, an Active Quantization Approach to extend reduced precision techniques to individual input images. We analyze potential energy savings exploited on a per-input basis, as well as a predictive framework for weight quantization. We show that such efforts combined net up to 46% energy savings depending on input. Specifically, we present:

- A technique for input and activation cropping
- An initial study on the accuracy and energy impacts of proposed techniques
- A mapping to the ISAAC architecture to support various precision levels and crops

978-1-5386-7367-6/18/$31.00 ©2018 IEEE
DOI 10.1109/EMC2.2018.00009



Prediction A	Prediction B	Prediction C
Gazelle	Gazelle	Gazelle
Pr = 0.7188	Pr = 0.8351	Pr = 0.6289

Figure 1: Motivational Example of Image Cropping

A. Motivational Example

We begin with a motivational example for intra-layer quantization and cropping on a particular input image. Intuitvely, we can see in Figure 1 the majority of the image is background. Firstly, we show that by cropping the entire edges of the image (making the pixel values 0), our test neural network (Vgg16) is still able to classify the image (Prediction B). In this particular case, the crop actually improves the prediction confidence over the baseline prediction A! However, due to the fact that the images are normalized to mean pixel values in our baseline, the 0 values do not actually propagate through the network. To simulate omitting these computations completely, we perform an experiment in which we crop activations (note that crop size reduces as the layer goes deeper). Even in this instance (Prediction C), Vgg16 is still able to correctly classify the example image.

To net the benefits of fewer computations, we must firstly analyze the image and decide which computations can be either omitted completely or computed at a lower precision. We will describe the trade-offs between the two approaches in detail during our discussion of activation quantization. Secondly, we must provide an architecture which supports variable precision computation. Finally, we must demonstrate the energy savings associated with these techniques. We first explain our input analysis technique, followed by a brief description of the architecture.

B. Intra-layer Activation Quantization

As shown in the motivation example, many images have irrelevant information encoded near their edges. However, each image has some ideal crop which can come from any combination of sides (top, left, bottom, right) and be for any amount of pixels or activations. While a more aggressive crop increases energy savings, it has the potential to cause a misclassification that might have otherwise been correct. Therefore we propose a lightweight image-preprocessing step in which an input image is analyzed, and particular cropping is predicted. The prediction algorithm can be tuned to favor energy savings or accuracy. While we leave out explicit discussion of such a predictor for this work-in-progress, we note that object localization within an image is well-studied. In particular, good inputs for a predictor might consist of characteristics such as color gradient and points-of-interest can be a good guide for object localization.

III. Architectural Implementation

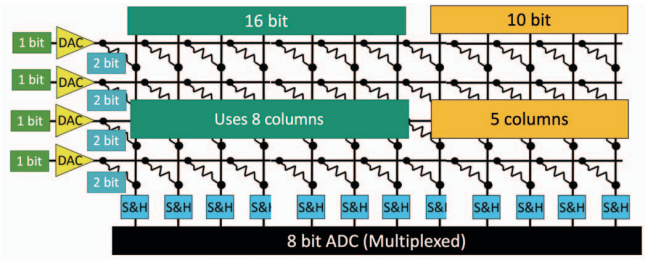


Figure 2: Remapping of Weights for Lower Precision

The main motivation behind AQuA is that, reducing the precision of weights and activations reduces the energy required to process a network. To quantify the impact of AQuA on energy per classification, we choose to evaluate it on the ISAAC [7] architecture. ISAAC is a memristor based in-situ CNN accelerator flexible enough to handle changes in activation and weight precision without the need for any architectural changes. By default, ISAAC considers 16-bit fixed point activations and weights

In ISAAC, an input (activations) is provided to a crossbar over N_A iterations (N_A being activation precision). As AQuA reduces intra- and inter-layer activation precision, processing activations with lower precision means processing for lesser number of iterations. This means, compared to baseline ISAAC, the energy consumed by ADCs is reduced by 2x when a layer has 8 bit activations. Similiarly, when AQuA crops a portion of a feature-map it is equivalent to a quantized precision of 0. Finally, because of how activations are mapped to crossbars, it is possible to vary this precision even within feature maps.

Weights in ISAAC are split into $N_W/2$ chunks (N_W being weight quantization), and 2-bits of information is programmed in a memristor cell. As showed in Figure 2, only 5 columns are needed when a layer has 10-bit weights, compared to 8 cells in baseline ISAAC. Programming quantized weights requires lesser number of cells, hence reducing the xbar+ADCs required in mapping the kernels. Hereby we show that AQuA can reduce the ADC overhead, which consumes 58% of the tile power in ISAAC [7]. In the results section we describe the energy savings unlocked by AQuA in detail.

IV. Results and Analysis

In order to validate cropping activation layers, we analyze the accuracy and energy impacts of activation cropping. We analyze a naive approach which applies the same type of crop to all images, as well as an ideal approach which applies the most aggressive crop to each image while maintaining the original accuracy. We conservatively assume that any image the network originally misclassified cannot be cropped.

A. Search Space

We consider cropping from each edge of the image: top, left, bottom, and right. While ultimately each cropping could be of any size and quantization level, we leave that design space exploration as future work. For the scope of this work we consider 16 different activation crops with the cropped area quantized to 0-bits. Different cropping techniques are noted in Tables I and II. Note that when we refer to *crops* or *cuts*, the actual 0-bit quantization occurs in the feature maps of the network, not on the image itself.

We apply each of these different cropping techniques to four different weight sets: full precision (32-bit floating point), 16-bit fixed point, 16-bit varied, and 12-bit fixed point. The *16-bit varied* weight set consists of variable precision between layers, with 16 refering to the maximum allowed precision per layer [2]. Conceptually, our goal is to demonstate that activation cropping can be applied in conjunction with other weight quantization techniques, allowing end users to stack the benefits of both techniques.

Table I: Naive Cropping results for ImageNet validation with 16-bit fixed point weights. Cut bit vector refers to T-Top, B-Bottom, L-Left, R-Right, where 1 indicates the corresponding side has been cropped.

Cut Size	Relative Energy	Cut [T B L R]	Top-1	Top-5
0	1.0x	[0 0 0 0]	65.09%	85.86%
1	0.861x	[0 0 0 1]	64.71%	85.77%
		[0 0 1 0]	64.66%	85.66%
		[0 1 0 0]	63.72%	84.97%
		[1 0 0 0]	63.97%	85.24%
2	0.743x	[1 0 1 0]	63.17%	84.77%
		[1 0 0 1]	63.22%	84.72%
		[0 1 1 0]	62.46%	84.41%
		[0 1 0 1]	62.69%	84.45%
		[1 1 0 0][1]	62.15%	84.05%
		[0 0 1 1]	63.59%	84.95%
3	0.624x	[1 1 1 0]	60.70%	83.07%
		[1 1 0 1]	60.83%	83.08%
		[1 0 1 1]	61.44%	83.51%
		[0 1 1 1]	61.05%	83.16%
4	0.526x	[1 1 1 1]	58.57%	81.64%

Table II: N_x and N_y refers to the number of rows cropped in x and y dimension of respective feature maps. For example, top-right cut of Conv3_1 is 48x48x256, whereas for all-but-left cut of Conv3_1 would be 48x40x256.

Layer	Original Feature Map Size	Crop Size (N_x, N_y)
Conv1_1	224x224x64	25,25
Conv1_2	224x224x64	25,25
Conv2_1	112x112x128	10,10
Conv2_2	112x112x128	10,10
Conv3_1	56x56x256	8,8
Conv3_2	56x56x256	8,8
Conv3_3	56x56x256	8,8
Conv4_1	28x28x512	5,5
Conv4_2	28x28x512	5,5
Conv4_3	28x28x512	5,5
Conv5_1	14x14x512	2,2
Conv5_2	14x14x512	2,2
Conv5_3	14x14x512	2,2

B. Methology

For this work we focus our efforts on VGGNet, specifically VGG16 (VGG16 has 13 convolutional layers, 3 fully connected layers) [8] using TensorFlow [9]. While all networks have slight variations in the number of layers, filter sizes, and more, the general structure for CNNs is similar. To measure the effects of reduced precision computation, we

[2]Layer-wise Weight Bit Widths: 11-13-13-13-13-12-12-12-11-11-12-12-12

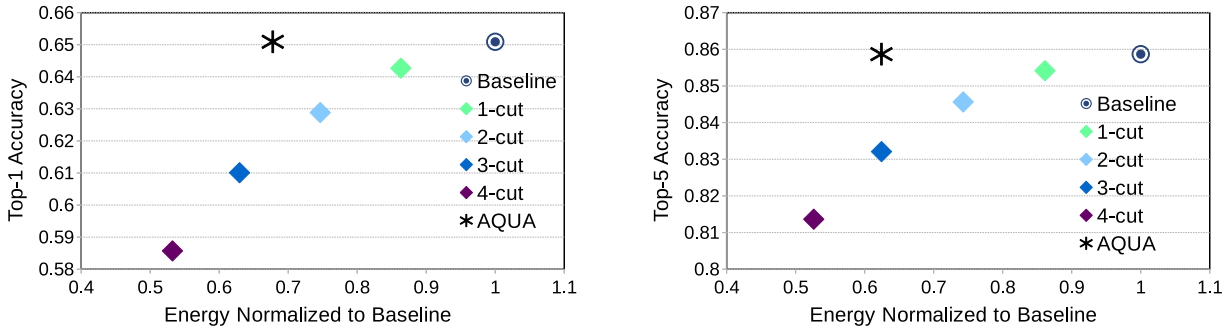


Figure 3: Energy vs Accuracy Trade-offs with 16-bit fixed point weights

measure the accuracy with the ILSVRC2012 validation set which comprises of 50,000 images. Note that our analysis uses full precision activations, and quantizing activations is left to future work. Additionally, we do not retrain weight sets, but rather generate our quantized weight sets by simply rounding the full precision weights.

C. Analysis

As mentioned earlier, we claim that certain regions of the activations can be considered unimportant, and hence can be cropped out. Though the region of unimportance depends on the input image, we start our analysis by testing the accuracy for all kinds of crops. Table I shows the top-1 and top-5 accuracies corresponding to the naive approach of applying same crops to all the 50K images. As shown in the table, we further classify the 16-crops into 4 classes: 1-cut, 2-cut, 3-cut, and 4-cut, with 0-cut being the original uncropped image and activation. As expected, with an increase in the crop size, the energy savings comes at a cost of reduced accuracy. As can be seen in the table, different types of crops tend to have different impact on the top-1 and top-5 accuracy.

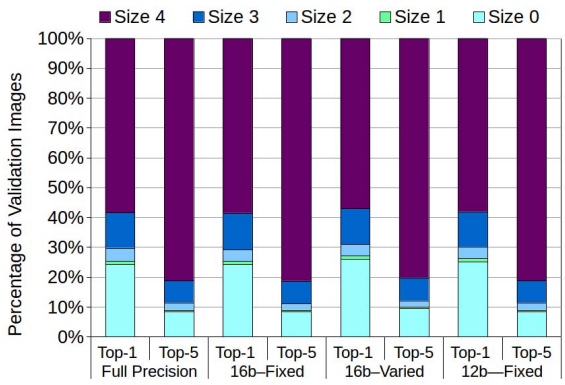


Figure 4: Max Cropping allowed on validation set images to retain original accuracy with a given weight set. Size refers to number of edges cropped.

Next, we perform an experiment to quantify the number of cuts each image can tolerate without incurring a drop in accuracy. As shown in Figure 4 We find that over 50% of the images tested are robust to even 4-cuts for both top-1 and top-5 accuracy. While the original image is required for approximately 20% of the inputs to maintain an accurate top-1 prediction, only around 10% of the inputs need it to make an accurate top-5 prediction (exact numbers vary depending on the weight set used). This clearly proves the need for an intelligent predictor to choose the appropriate crop to maximize the accuracy-energy trade-off for an image.

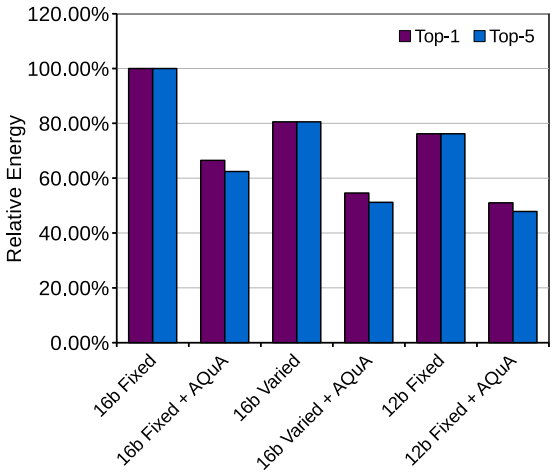


Figure 5: Energy comparison of different weight-sets for maintaining top-1 and top-5 accuracy, both with and without AQuA (activation cropping). Energy is shown relative to 16b Fixed, the baseline comprising of a 16-bit fixed point weight set and uncropped activations.

Figure 3 plots the accuracy versus energy for all possible cuts, and an oracle version AQuA. It can be clearly seen that each crop saves signficant energy compared to the baseline, at the cost of accuracy. However, AQuA bridges this gap by performing the crop adaptively based on input image such that original classification accuracy is retained while

providing an energy savings of up to 33.5% and 42.0% for top-1 predictions and top-5 predictions, respectively. Finally, we show that AQuA can also be applied in conjunction with other quantization techniques in Figure 5. Activation cropping energy savings are nearly identical, even with differently quantized weight sets.

V. Conclusion and Future Work

In this work we show that adaptive activation quantization techniques can signficantly reduce the number of computations. Our proposed energy saving techniques can be applied in conjunction with existing quantization approaches. We show that such an approach has the potential for energy improvements of up to 41.8% with simple reconfiguration of a CNN accelerator, ISAAC. Moving forward, we expect to do a deeper dive on variable levels of precision within activations, rather than just quantization to 0-bits of precision for unimportant regions. We also seek to build a predictor and analyze the accuracy-energy trade-offs between aggressive and conservative policies.

References

[1] M. H. Luke Durant, Olivier Giroux and N. Stam, "Inside Volta: The World's Most Advanced Data Center GPU," 2017, https://devblogs.nvidia.com/inside-volta/.

[2] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers *et al.*, "In-Datacenter Performance Analysis of a Tensor Processing Unit," 2017.

[3] P. Judd, J. Albericio, T. Hetherington, T. M. Aamodt, and A. Moshovos, "Stripes: Bit-Serial Deep Neural Network Computing," in *Proceedings of MICRO-49*, 2016.

[4] E. Park, J. Ahn, and S. Yoo, "Weighted-entropy-based quantization for deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] J. Albericio, A. Delmás, P. Judd, S. Sharify, G. O'Leary, R. Genov, and A. Moshovos, "Bit-pragmatic deep neural network computing," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2017, pp. 382–394.

[6] L. Kull, T. Toifl, M. Schmatz, P. A. Francese, C. Menolfi, M. Brandli, M. Kossel, T. Morf, T. M. Andersen, and Y. Leblebici, "A 3.1 mW 8b 1.2 GS/s Single-Channel Asynchronous SAR ADC with Alternate Comparators for Enhanced Speed in 32 nm Digital SOI CMOS," *Journal of Solid-State Circuits*, 2013.

[7] A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. Strachan, M. Hu, R. Williams, and V. Srikumar, "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," in *Proceedings of ISCA*, 2016.

[8] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, " TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/