CNN Model Pruning and Quantization for FPGA

Chuxuan Hu, Hanlin "Asher" Mai, Jiaru Zou, Joseph Rejive, William Eustis, Volodymyr Kindratenko

Department of Electrical and Computer Engineering, Grainger College of Engineering, University of Illinois at Urbana-Champaign

INTRODUCTION

Convolutional Neural Network (CNN), such as AlexNet, VGGNet, and GoogLeNet have been shown effective to achieve high accuracies for image classification tasks, given a large enough labeled training data set. However, the convolutional layers inside these CNNs consume enormous amounts of memory to store these convolutional kernels. This is partly due to the sheer amount of 32-bit floating point numbers these kernels store and use during training and inference. By zeroing some of the lowest value numbers in each 3x3 kernels (i.e. pruning) and turning the 32-bit floating point numbers into 8-bit integers (i.e. quantization), we can effectively reduce memory usage by 4 times while making inferences faster purely due to the fact that integer multiplication and addition are much faster than corresponding floating point operations.

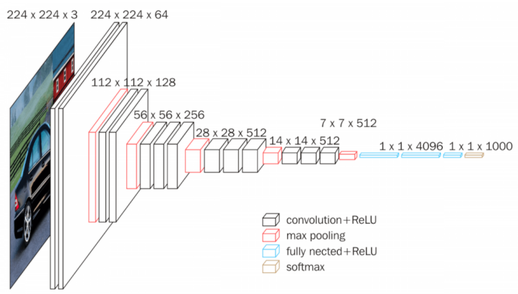
AIM

We apply pruning and quantization techniques to reduce the memory usage of large CNNs, so that they can be run on hardware with low memory availability, such as a Field Programmable Gate Array (FPGA), which is a lightweight device with constrained recourses, and are more widely accessible than GPUs. The CNN is trained on HAL GPU cluster from the National Center for Supercomputing Applications (NCSA). The trained convolutional kernels and linear layer weights will be pruned, quantized and copied over to an FPGA implementation for the inferencing step of image classification. We aim to speed up the computation with integer operations instead of floating point operations

METHOD

Convolutional Neural Network

We use VGG16 to conduct our experiment, which contains 13 convolutional layers and 3 fully connected layers.



Pruning

We first train our VGG16 network on the Cifar-10 Dataset, then for each 3x3 convolutional kernel in the i-th convolutional layer, we zero out n_i of the 9 floating point numbers with the lowest absolute value, where n = [2, 2, 3, 4, 5, 6, 6, 7, 7, 7, 7, 8, 8].

$$\begin{matrix} -0.54 & -0.13 & 0.54 \\ -0.57 & 0.38 & 0.78 \\ 0.34 & 0.24 & 0.49 \end{matrix} \longrightarrow \begin{matrix} -0.54 & 0.0 & 0.54 \\ -0.57 & 0.38 & 0.78 \\ 0.0 & 0.0 & 0.49 \end{matrix}$$

Quantization

After the pruning step, we employ a Post Training Static Quantization, which converts all 32-bit floating point convolutional kernels and fully connected layer weights into 8-bit integers. We use per-channel quantization rather than per-tensor quantization. For each channel in each conv layer, we calculate a scaling factor S and zero-point Z, using the following equations:

$$S = \frac{\beta - \alpha}{255}, \qquad \mathrm{Z} = -\frac{\alpha}{S}$$

where α and β are the minimum and maximum that the inputs can be, which can be calibrated during forward propagation. We then use these per-channel S and Z to compute our quantized 8-bit conv kernel weights with the following:

$$w_{int8} = round\left(\frac{w_{float32}}{S} + Z\right)$$

$$\begin{matrix} -0.54 & 0.0 & 0.54 \\ -0.57 & 0.38 & 0.78 \\ 0.0 & 0.0 & 0.49 \end{matrix} \longrightarrow \begin{matrix} -87 & 0 & 88 \\ -93 & 62 & 127 \\ 0 & 0 & 80 \end{matrix}$$

EMPIRICAL EVALUATIONS

We conducted extensive experiments on Cifar-10 Dataset elaborate the efficacy our pruned and quantized model. Cifar-10 Dataset contains images of 10 categories (airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks).

Cifar-10 Dataset: each category composes of images of 3(channels) 32(width) * 32(height) pixels.*

Our quantization and pruned model is pre-trained with a portion of the dataset and fully tested.

Compared to the original VGG16 network, our method has greatly improved running time and memory efficiency of the convolutional neural network (CNN)s without sacrificing the accuracy of the model for image classifications tasks.

Effectiveness

Our pruned and quantization model achieves **87.57%** correctness in image classification. There's almost no accuracy loss comparing to the original VGG16 network, where the accuracy is 88.20%.

Efficiency

Our quantized and pruned model shows superiority over baseline VGG16 model in both running time and memory.

Running time efficiency. Our pruning and quantization technique accelerates the image classification process by **2.5 times**. Specifically, it takes our network 51.99 seconds to generate results while the original model runs for 136.13 seconds.

Memory efficiency. Our pruned and quantized network takes up only **5.42%** memory spaces of the original VGG16 network for storage.

CONCLUSIONS

In this research, we propose a novel pruning and quantization technique that effectively compresses a deep neural network (VGG16) without affecting its performance. The method prunes convolutional layer weights with smallest absolute values, quantizes float64 weights and inputs into int8 leveraging our specially-developed quantization technique for pruned network and self-adjusts activation functions based on post-training quantization (PTQ) to maintain correct classification results. We demonstrate the superiority of our compressed model over the original VGG16 network through extensive experiments. The pruning and quantization method can also be extended to various other networks, e.g., ResNet, Transformer, e.t.c..

FUTURE WORK

The pruned and quantized VGG16 model has only been tested on CPU and GPU implementations, and the accuracy has not yet been verified on the FPGA implementation. There is still more work that needs to be done to ensure the FPGA implementation of the model matches the performance of ones tested on the CPU and GPU.

ACKNOWLEDGEMENTS

NCSA

ILLINOIS