EE 367 Project Proposal

Haley So

February 25, 2022

Abstract

To be filled out later.

1 Introduction

CNNs have become ubiquitous in many computational imaging tasks. However, these can take ample time to run and large amounts of memory to store the models. Recently, there's also been a push to bring the computation onto the focal plane and to extract only the salient information, not only for a speedup, but also to tackle the limited bandwidth problem when it comes to transferring data.

The goal is to bring CNN computation onto a sensor, such as the SCAMP-5, and have this output from the network inform how we send data out, or how we capture new data, essentially doing adaptive sensing. To make this happen, we first have to tackle a few challenges.

SCAMP-5, along with other sensors, are limited in memory. SCAMP-5 has 12 digital registers and 6 analog registers. In regular CNNs, the weights for each layer are in 32bit, but it would be impossible to save these all these thousands of weights on the SCAMP or probably on any other sensor due to the limited memory.

In the past few years, researchers have started looking into binarized networks or highly quantized networks to achieve tasks like image classification or image segmentation. It's challenging to get as good results as the full precision networks, but there are teams that created training strategies and networks that work relatively well for small datasets. The work ends up being implemented on FPGAs etc but not quite in the same way as we would like, which is to inform how we capture the next image.

In this project, I want to do a deep dive into binary networks to understand the tradeoff between precision and memory footprint, and also to see if I can improve on this field of work of binary neural networks or highly quantized networks.

This project can later be extended as well to a more general concept. The idea that the previous frame can inform what layer weights to use is essentially adaptive pruning on the focal plane. With the ability to go as fast as 100,000fps, we could do some really high speed applications.

2 Timeline

We want to answer a few questions in this project: 1. How does one train a good binary neural network? 2. How much precision do you lose by going from full precision to binary? But what do we gain in memory and time? 3. For the same memory footprint, how deep can our binary network go and how high can we get the precision? Looking at accuracy, memory footprint, inference time, I will compare the following on the CIFAR10 dataset [3] to answer the above questions.

Week 8: train baselines with different methods such as the straight-through estimator (STE), gumbel softmax, and different training strategies

1. 2 layer CNN full precision with and without bias
2. 2 layer CNN binarized weights and activations using STE with and without bias
3. 2 layer CNN binarized weights and activations using gumbel with and without bias
4. different BNN training methods on the same sort of architecture
5. Additional Baselines – other state of the art BNNs or quantized neural networks such as IR-net, XNOR-net, Bi-Real, BENN

Week 9: Memory footprint study: With a good binarized network training strategy, compare deeper binary networks to full precision networks with the same memory footprint to understand the trade-off.

1. increase number of binary layers or filters up to the same memory footprint as full precision network
2. change parameters like size of kernels, depth, binary type (0,1 or -1, +1)

Week 10: Potential additional memory study

1. try models with different kinds of quantization: binary, ternary... quantized at different levels.
2. The final goal is to understand how to train binary networks and to understand the tradeoffs between accuracy and memory so we can better inform our future adaptive sensing projects.

3 Related Work

(work in progress) Binary Networks related work: IR-Net [5], XNOR-net [6], Regularized Binary Network Training [7]

Quantized Network related work: TBN [8]

CNNs on SCAMP or other sensors related work: Fast cnns on the scamp [1], [4]

4 Experiments

5 Discussions

[2]

References

[1] Bose, L., Dudek, P., Chen, J., Carey, S. J., and Mayol-Cuevas, W. W. Fully embedding fast convolutional networks on pixel processor arrays. In *Computer Vision – ECCV 2020* (Cham, 2020), A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Springer International Publishing, pp. 488–503.

[2] Bulat, A., Liu, Z., Martinez, B., Leontiadis, I., and Tzimiropoulos, G. Binary networks for computer vision cvpr 2021 workshop, 2021.

[3] Krizhevsky, A. Learning multiple layers of features from tiny images.

[4] Liu, Y., Bose, L., Chen, J., Carey, S., Dudek, P., and Mayol-Cuevas, W. High-speed lightweight cnn inference via strided convolutions on a pixel processor array. British Machine Vision Virtual Conference ; Conference date: 07-09-2020 Through 10-09-2020.

[5] Qin, H., Gong, R., Liu, X., Shen, M., Wei, Z., Yu, F., and Song, J. Forward and backward information retention for accurate binary neural networks. In *IEEE CVPR* (2020).

[6] Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Computer Vision – ECCV 2016* (Cham, 2016), B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Springer International Publishing, pp. 525–542.

[7] Sajad Darabi, Mouloud Belbahri, M. C. V. P. N. Regularized binary network training. In *NeurIPS19 Workshop on Energy Efficient Machine Learning and Cognitive Computing* (2019).

[8] Wan, D., Shen, F., Liu, L., Zhu, F., Qin, J., Shao, L., and Tao Shen, H. Tbn: Convolutional neural network with ternary inputs and binary weights. In *The European Conference on Computer Vision (ECCV)* (September 2018).