Investigating the Efficacy of Shortcut Connections in ResNet Variants with Hybrid Binary Convolutions for Architecture Pruning Purposes in Edge Computing Applications

Mani Amani
UCSD

Abstract

With advancements in edge computing and robotics, the need for computationally cheap neural networks is ever increasing (Shi et al. (2016)). Methodologies have been since introduced such as architecture design, quantization and pruning to create smaller and more efficient models. Many different approaches exist such as MobileNet (Howard et al. (2017)), EfficientNet (Tan and Le (2020)), Binarized Nerual networks and XNOR-Nets (Rastegari et al. (2016)). Pruning has proven to be an extremely effective method of neural network acceleration (Molchanov et al. (2017)). In this paper we investigate the effects of pruning shortcut connections in Binary and Full Presicion ResNet variants. Our results support partial pruning of hybrid binary ResNet models resulting in competitive accuracies with state of the art models while decreasing model size by 2x.

1 Introduction

Deep Learning and neural networks have opened has had a massive effect on schools of study such as Computer Vision, Natural Language Processing, biology and many other fields (Schmidhuber). End devices such as smartphones and Internet-of-things sensors generate certain data that can be used effectively using deep learning methodologies, however both inference and training of these models are very computationally taxing which can cause issues for real life edge computing requirements (Chen and Ran). The applications of Machine Learning specifically in resource constrained areas such as Non-volatile memories and SSD mappings that utility of more state of the art deep learning algorithms has been pulled back due to high computational costs (Sun et al. (2023)). One of the proposed methods that alleviates the computational requirements of Deep Learning methods is the binarization of weights and activation functions that turn 32-bit parameters into 1-bit parameters which are either +1 or -1, decreasing both memory and inference requirements by lowering computational complexity (Courbariaux et al. (2016)). However, traditional BNN's tend to lose significant expressive capabilities due to the 1-bit parameters . Other models such as XNOR-Nets and BNN+ in order to optimize current architecture through binarizing weights and activation functions. All different BNN's use 1-bit binarized weights however different processes could be added such as regularization and gain terms, however the existence of these processes add to the computational complexity of the model (Simons and Lee). Other downsides of more complex BNN models such as XNOR-Net is that they need specialized kernels and software to take full advantage of the bit-wise operations (Xu and Pedersoli (2019)). ResNets have been one of the most effective architectures for computer vision which most BNN researchers use as a benchmark (Simons and Lee). ResNet utilizes shortcut connections in residual blocks in order to mitigate the effects of vanishing gradients in deep neural networks (He et al.). However DNN's such as ResNet have millions of parameters that render them computationally taxing which are non suitable for edge devices. Training BNN's have also proven to be more difficult and slower, in addition, the convergence of the weights to either a positive or negative values in BNNs, many of the gradients that are calculated have no effect on the loss or the accuracy of the model, implying that there is a significant amount of redundant parameters adding to the computational requirements of the models (Tang et al.). Given the trade off between accuracy and speed there has been some research on the effects of hybrid binarized networks that utilize both 32-bit and 1-bit parameters (Chakraborty et al. (2019)). While quantization of the weights is a popular approach, another methodology to model compression is pruning, deleting non-critical and redundant by deleting redundant and non critical

processes to reduce computational insensitivity (Ye et al. (2018)). One pruning avenue we plan to explore is the removal of shortcut connections in binarized residual networks, We argue that in binarized residual networks the effect of the binary convolved identity mappings would be less effective in solving the vanishing gradient problem due to the convergence to either -1 or +1 numbers which would have smaller expressiveness. By also decreasing the shorcut connections, we reduce the MACs of the model which is the biggest factor in computational complexity (Mei et al.). Since Neural Networks have millions or even billions of parameters, such large models are prone to overfitting, so reducing these parameters can lead to more satisfying results (Ye et al. (2018)).In this paper we train and compare training losses, accuracies and variants of the ResNet architecture with different shortcut connections and binarized weights.

2 Methods

2.1 ResNet Architecture

The ResNet architecture was first introduced in the paper "Deep Residual Learning for Image Recognition" by He et al. (He et al.). as a solution to the bottleneck problem in deep neural networks. This architecture secured the first place in ImageNet challenge (ILSVRC 2015) with 3.57% error, which for first time, the CNN achieve error rate more better than human perception (Elhassouny and Smarandache).

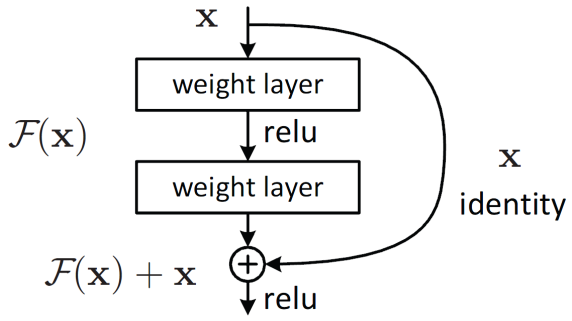

Figure 1: Residual Learning Blocks

In this paper we experiment with a 14 layer residual network as visualized in Figure 2. The reason for a 14 layer residual network would be we would see the effect of binarization of the weights and the removal or addition of shortcut connections in a more shallow architecture. The ResNet 14 model in question starts with a Full-Precision convolution into batch normalization and ReLu activation, followed by three residual blocks. Each residual block has two convolutions into batch normalization into ReLu activation. The convolutions can be either fully binarized or Full Precision depending on the variant. In the shortcut connections, A 1x1 convolution exists for computational optimization and mapping the output channels to a new space which has proven to be effective (Szegedy et al. (2017)). For the binarized and hybrid binarized variations a binary convolutions used and full precision if the block uses full precision weights. After the 1x1 convolution the data is passed through a batch normalization and ReLu activation which then gets added back to the output for identity mapping. After the residual blocks, the model utilizes and adaptive 2D average pooling then connecting to a fully connected layer for the final classification. For the hybrid residual variants, the same architecture from Figures 2–3–4, however the first and third layer would not have shortcut connections and for the non-residual layer would not have any shortcut connections in both full precision and binarized variants.

2.2 Binary Convolutions

There are multiple ways to approach Binary convolutions such as XNOR and popcount (Rastegari et al. (2016)). In this paper, we utilize the sign function inspired by traditional BNNs (Courbariaux et al. (2016)):

$$w_{ib} = sign\left(w_i\right) \tag{1}$$

This binarization would theoretically help with memory storage reduction by quantizing the 32-bit parameters into 1-bit and make the calculations less expensive.

2.3 Assessing Theoretical Memory Usage

Due to hardware and software differences in real life applications, a fair way to compare the memory requirements is through effective 32-bit parameters. In order to have a measurement of how much memory is required for a model, one good method is to calculate the effective 32-bit parameters. The model parameters could be in different bit counts meaning each 32-bit parameter requires the same amount of memory as 32 1-bit parameters. In our case, we utilize a mix of 1-bit and 32-bit parameters. We convert every 32 1-bit parameter into *1 effective 32-bit parameter.*

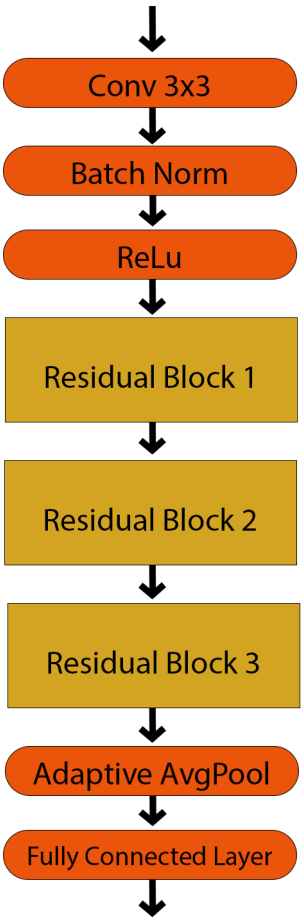

Figure 2: Implemented ResNet Architecture

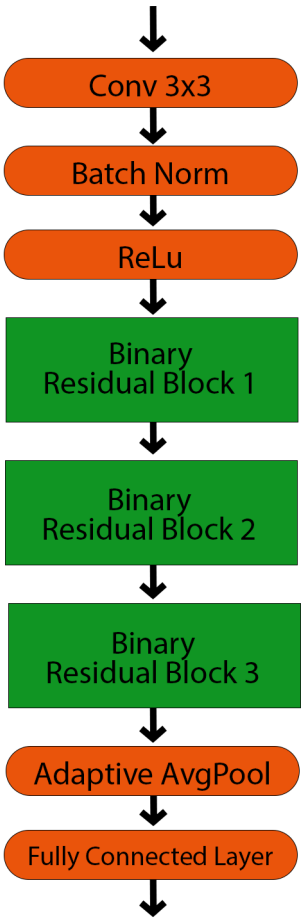

Figure 3: Implemented Fully Binarized ResNet Architecture

2.4 Training

Since Binarized Neural Networks are notoriously hard to train (Chen et al. (2021)),We use different training procedures for each of the variants. All Variants utilize and Adam Optimizer, L2 regularization and a dropout layer in the residual blocks to prevent overfitting. The full-precision fully residual models were more prone to overfitting so it used a L1 regularization in addition to the rest for better training. The initial learning rate was set at 0.001, however that was changed as an adjustment to the models performance of each separate model. All models were trained for 100 epochs for a fair comparison. All models were trained on a Nvidia GeForce RTX 2060 SUPER GPU.

3 Results

Figures 5–6–7 show the validation loss during training. The results show that hybrid binarization does not have a drastic effect on accuracy. As can be seen on Tables 1–**??**–**??**, The drop of accuracy is significantly more drastic from Hybrid Binary to Full Binary than from Full Precision to Hybrid Binary.

4 Discussion

The effects of shortcut connections will vary in model depth. This model is relatively shallow in order to be order to better isolate and evaluate the effects of the shortcut connections while being able to control other parameters. As it can be seen even the existence of one shortcut connection drastically improves the accuracy of the model, implying the diminishing returns of having shortcut connections in every residual block making it a worthwhile endeavour to asses the application in even full precision models. This also raises the possibility that there many shortcut connections in deeper ResNet models such as ResNet50 and ResNet101 which would have many extra shortcut connections, however more research is necessary to understand the exact implications. Pruning shortcut connections

Model	Effective 32-bit Param	Accuracy
ResNet14	2.777×10^6	91.52
Hybrid Res ResNet14	2.744×10^6	88.31
No Residual ResNet14	2.735×10^6	85.17

Table 1: Full Precision ResNet Variant's Max accuracy and Effective 32-bit Parameters

Model	Effective 32-bit Param	Accuracy
Hybrid Binary ResNet14	6.056×10^5	88.59
Hybrid Binary Hybrid Res ResNet14	6.051×10^5	88.49
Hybrid Binary No Residual ResNet14	6.025×10^5	86.96

Table 2: Hybrid Binary ResNet Variant's Max accuracy and Effective 32-bit Parameters

Model	Effective 32-bit Param	Accuracy
Binary ResNet14	1.893×10^4	48.08
Binary Hybrid Res ResNet14	1.891×10^4	49.09
Binary No Residual ResNet14	1.828×10^4	43.89

Table 3: Full Binary ResNet Variant's Max accuracy and Effective 32-bit Parameters

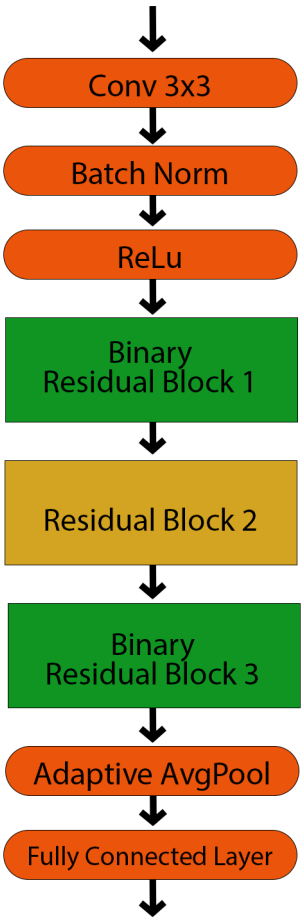

Figure 4: Implemented Hybrid Binarized ResNet with Full Residual Connection Architecture

can be used in theory be used in any model that uses them regardless of XNOR or bit-count operations and models with software and hardware requirements, which makes pruning techniques extremely versatile. This experiment merely scratches the surface of the applicability of these modifications in real life scenario. Pruning ResNet is not necessary for all applications. Certain potential applications can include some edge computing algorithms such as Faster R-CNN use ResNet architectures.(Wang et al.).

The existence of shortcut connections or lack there of in Hybrid Binary Neural Networks seems to have the least effect on the model's generalizability. This was to be expected since the connection utilize binary convolutions while floating-point parameters are carrying the bulk of the model capacity. This notion is also supported by the idea that most parameters are not effective in model capacity in binary neural networks (Tang et al.). The implications of these could be the partial pruning

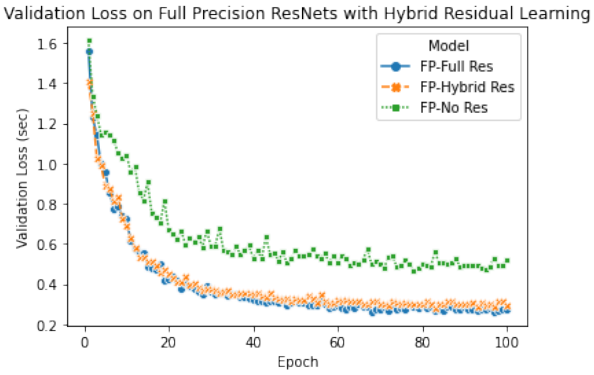

Figure 5: Full Precision 32-bit ResNet14 Training

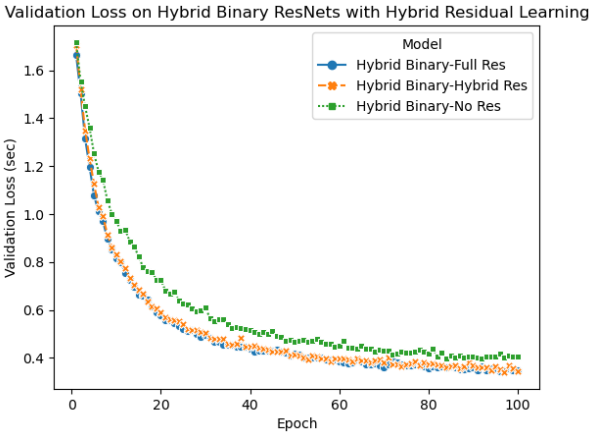

Figure 6: Hybrid Binarized ResNet14 Training

of certain shortcut connections to current hybrid binary implementations.

The Hybrid Binary Model raises a novel way of approaching low-cost models. While weight quantization is a well established methodology for increasing efficiency, the mixing an matching of different bit-counts is not researched at the time of writing. However one downside could be that the model would more likely than not need specialized software and kernels to take advantage of the quantized weights. The Hybrid Binary model also seems to fall nicely in between the speed-accuracy spectrum. Compared to open-source implementations of fully binarized networks such as Binarized VGG (Hubara et al. (2016)) and XNOR-NIN (Cao et al. (2017)), the model has less parameters and competitive accuracy. The next step in our research is properly implementing this model either as a backbone or as is in edge computing devices and embedded systems to test their efficacy in a real-world setting

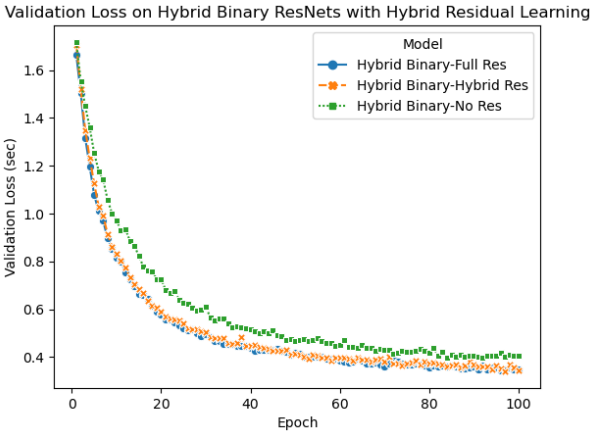

Figure 7: Fully Binarized ResNet14 Training

References

Jie Cao et al. 2017. Xnor-net-pytorch. https://github.com/jiecaoyu/XNOR-Net-PyTorch.

Indranil Chakraborty, Deboleena Roy, Aayush Ankit, and Kaushik Roy. 2019. Efficient hybrid network architectures for extremely quantized neural networks enabling intelligence at the edge.

Jiasi Chen and Xukan Ran. Deep learning with edge computing: A review. 107(8):1655–1674. Conference Name: Proceedings of the IEEE.

Tianlong Chen, Zhenyu Zhang, Xu Ouyang, Zechun Liu, Zhiqiang Shen, and Zhangyang Wang. 2021. "BNN - BN = ?": Training Binary Neural Networks without Batch Normalization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4614–4624, Nashville, TN, USA. IEEE.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1.

Azeddine Elhassouny and Florentin Smarandache. Trends in deep convolutional neural networks architectures: a review. In *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, pages 1–8.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized Neural Networks. In *Advances in Neural Information*

Processing Systems, volume 29. Curran Associates, Inc.

Linyan Mei, Mohit Dandekar, Dimitrios Rodopoulos, Jeremy Constantin, Peter Debacker, Rudy Lauwereins, and Marian Verhelst. Sub-word parallel precision-scalable MAC engines for efficient embedded DNN inference. In *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 6–10.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning convolutional neural networks for resource efficient inference.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. 61:85–117.

Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646.

Taylor Simons and Dah-Jye Lee. A review of binarized neural networks. 8(6):661. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

Jinghan Sun, Shaobo Li, Yunxin Sun, Chao Sun, Dejan Vucinic, and Jian Huang. 2023. Leaftl: A learning-based flash translation layer for solid-state drives. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS 2023, page 442–456, New York, NY, USA. Association for Computing Machinery.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Mingxing Tan and Quoc V. Le. 2020. Efficientnet: Rethinking model scaling for convolutional neural networks.

Wei Tang, Gang Hua, and Liang Wang. How to train a compact binary neural network with high accuracy? 31(1). Number: 1.

Yanbo J. Wang, Ming Ding, Shichao Kan, Shifeng Zhang, and Chenyue Lu. Deep proposal and detection networks for road damage detection and classification. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5224–5227.

Xianda Xu and Marco Pedersoli. 2019. A computing kernel for network binarization on pytorch. *CoRR*, abs/1911.04477.

Jianbo Ye, Xin Lu, Zhe Lin, and James Z. Wang. 2018. Rethinking the Smaller-Norm-Less-Informative Assumption in Channel Pruning of Convolution Layers. ArXiv:1802.00124 [cs].