Less is More: Quantization of Deep Neural Networks

Brady Steele
Georgia Institute of Technology
Atlanta, GA

Yuan Shi
Georgia Institute of Technology
Atlanta, GA

Prithwish Mukherjee
Georgia Institute of Technology
Atlanta, GA

Abstract

This project investigates the effects of weight quantization on deep neural network performance, focusing on image datasets. Weight quantization reduces the precision of model weights which leads to reduction in model size and computational requirements, making them suitable for resource-constrained devices. The project explores the impact of these techniques on model accuracy, training efficiency, and generalization capabilities. The research is motivated by the need to develop efficient and effective deep learning models for image classification tasks that can be easily deployed in real-world applications.

1. Introduction

This project investigates the impact of weight quantization and pruning on deep learning models applied to image classification tasks. Weight quantization involves reducing the numerical precision of model weights, while pruning selectively eliminates less impactful connections. These techniques aim to address the challenge of deploying deep learning models in resource-constrained environments, such as mobile or embedded devices, where computational power and memory are limited. By reducing model size and computational complexity, quantization and pruning can enable real-world applications that were previously infeasible due to hardware limitations.

The development of efficient deep learning models has far-reaching implications across various domains. In healthcare, for instance, compact models could be deployed on portable medical devices, enabling faster and more accessible diagnostics. In the realm of autonomous vehicles, optimized models could enhance real-time decision-making capabilities. Moreover, the proliferation of smart devices and the Internet of Things (IoT) necessitates the deployment of lightweight models that can operate efficiently on edge devices. The successful implementation of quantization and pruning techniques would thus unlock a wide array of possibilities for innovative applications.

Current state-of-the-art deep learning models often exhibit substantial size and computational demands, hindering their deployment on resource-constrained devices. While existing quantization and pruning methods have demonstrated promising results, their impact on model accuracy, training efficiency, and generalization capabilities remains an active area of research. This project aims to contribute to this ongoing exploration by evaluating the performance of established algorithms, namely Binary Connect and Incremental Network Quantization (INQ), on the MNIST image dataset [5]. The findings of this research will shed light on the trade-offs between model compression and performance, informing the development of future techniques for efficient deep learning.

2. Dataset

The MNIST (Modified National Institute of Standards and Technology) dataset [5] is a widely used benchmark for evaluating image classification algorithms. Comprising 60,000 training and 10,000 test images of handwritten digits, this dataset provides a standardized and accessible platform for assessing model performance. Its simplicity, consisting of 28X28 gray-scale images with limited variability, allowed focusing on the core algorithms and techniques under investigation, making it an ideal choice for evaluating quantization and pruning methods in this project.

The MNIST dataset offers several advantages for this study:

- **Standardization**: As a well-established benchmark, MNIST allows for direct comparisons with existing

literature, providing context for our results within the broader field of deep learning research.

- **Size and Complexity**: With 70,000 total images, MNIST is large enough to be meaningful yet small enough to allow for rapid experimentation and iteration.

- **Class Balance**: The dataset is nearly balanced across its 10 classes (digits 0-9), minimizing the impact of class imbalance on our analysis of model performance.

- **Preprocessing**: MNIST images are pre-processed (centered, size-normalized), reducing the need for extensive data preparation and allowing us to focus on the core aspects of model compression.

- **Low-dimensional Input**: The 28×28 pixel format results in a manageable input size of 784 features per image, making it feasible to experiment with various network architectures without excessive computational demands.

While MNIST's simplicity is advantageous for our study, it's important to note its limitations. The dataset's lack of color information, limited variability, and relatively simple classification task may not fully represent the challenges faced in more complex real-world scenarios. Future work could extend this research to more challenging datasets such as CIFAR-10 or ImageNet to validate the effectiveness of the quantization and pruning techniques in more demanding contexts.

Despite these limitations, MNIST serves as an excellent starting point for our investigation, providing a solid foundation for understanding the fundamental impacts of weight quantization and pruning on neural network performance.

3. Approach

In this project, we investigated the impact of weight quantization on deep learning models for image classification. Our approach involved utilizing the MNIST dataset, a well-established benchmark for evaluating image classification algorithms.

Two distinct weight quantization schemes were employed:

- **Incremental Network Quantization (INQ)** [11]: This scheme progressively quantizes weights using power-of-two values, striking a balance between model size reduction and accuracy preservation.

- **Binary Connect (BC)** [4]: This method quantizes weights to binary values (-1 or 1), drastically reducing memory requirements and enabling efficient inference.

To evaluate the effectiveness of these quantization techniques, we established baselines for the following two network architectures - simpler (CNN + Fully Connected) architecture and a more complex (CNN + ResNet + Batch Normalization) architecture. This is to understand any differences in the impact of quantization because of the network structure.

- **CNN + Fully Connected** *(to be referred as CNN model)*: A traditional convolutional neural network (CNN) followed by a fully connected layer (FC). The network begins with a convolutional layer containing 32 filters, each 3x3 in size, followed by a ReLU activation function and a 2x2 max-pooling layer. This is repeated with another convolutional layer, this time with 64 filters. The output of these layers is flattened into a 1D vector with 1600 elements and passed through two fully connected layers with 120 and 84 neurons respectively, both using ReLU activation. Finally, a fully connected output layer with 10 neurons produces the predicted class probabilities.

- **CNN + ResNet + Batch Normalization** *(to be referred as CNN+RES model)*: A deeper network incorporating residual connections and batch normalization for improved performance. It begins with two CNN layers, each with 3x3 filters and padding to maintain spatial dimensions. The first layer has 32 filters and the second has 64. Batch normalization is applied after each convolutional layer to stabilize training, followed by ReLU activation. The network then introduces a residual block (RES), where the input is added back to the output after passing through two additional convolutional layers (both with 64 filters). This helps with training deeper networks by allowing gradients to flow more easily. The residual block is also followed by batch normalization and ReLU activation. Next, a 2x2 max-pooling layer reduces the spatial dimensions, and an adaptive average pooling layer transforms the output into a 1D vector with 64 elements. Finally, a fully connected layer (FC) with 10 neurons produces the predicted class probabilities.

We also present variations of the standard quantization algorithms mentioned before.

- **INQ-Binary**: This is a hybrid of **BC** [4] and **INQ** [11] algorithms where it follows the same progressive quantization policy as **INQ** but instead of reducing to powers-of-two, this method quantizes weights to binary values (-1 or 1) based on **BC** quantization technique. *The goal is to understand the relative performance of INQ-Binary compared to the established BC algorithm.*

- **Layerwise Quantization**: Restrict the quantization to specific layers (e.g. CNN only, only shallow layers, only deep layers etc.). *The goal is to understand the importance of specific layers and corresponding bias w.r.t overall network performance*.

4. Experiment Methodology

We modified existing code respositories [1] and [2] for INQ and Binary Connect algorithms respectively. The code required for obtaining the experimentation results is available here. For both the network structures, we created baseline models using Cross Entropy Loss, 0.001 learning rate, 0.001 L2 regularization and Adam Optimizer [8]. For INQ, we use a 5 step quantization, we used [0.2, 0.4, 0.6, 0.8, 1.0] for all the experiments - where we started with quantization of 20% weights in the 1st iteration, 40% in the 2nd, 60% in the 3rd, 80% in the 4th and all the weights in the final step. The dataset was split into an 80:20 ratio for training and validation (using 64 as the batch size), ensuring a robust evaluation of our quantization methods.

For each quantization scheme and network architecture, we meticulously evaluated the following aspects:

- Loss: The difference between predicted and actual labels, indicating model performance.
- Accuracy: The proportion of correctly classified images, a direct measure of classification capability.
- Clip Range Difference: The difference between the maximum and minimum values of weights before clipping, influencing quantization error.
- Bias Term Quantization: The quantization of bias terms, affecting model expressiveness.
- Bit Width or WB (only applicable for INQ): The number of bits used to represent each weight, reflecting the degree of quantization. *We experimented with 2, 3 and 5 for INQ algorithm.*
- Quantization Epochs (only applicable for INQ): The number of epochs to retrain the network after each quantization step. *We experimented with 2, 5 and 10 for INQ algorithm.*

These quantization decisions directly impact the final quantization loss. Additionally, we examined the weight and bias distributions within hidden layers to analyze the trade-off between clipping and rounding errors. By comprehensively evaluating these factors, we aim to gain insights into the optimal strategies for achieving efficient and accurate quantized models for image classification tasks.

Overall, our approach provides a systematic framework for evaluating the impact of different quantization schemes and network architectures on model performance. Through rigorous experimentation and analysis, we strive to advance the field of model quantization and contribute to the development of efficient deep learning models for real-world applications.

5. Results

5.1. Accuracy and Training Time

Model	Test Acc.	Train Acc.	Valid Acc.	Time (hrs.)
Baseline	97.84	97.83	97.56	0.25
INQ[NB][P]	97.65	97.64	97.45	0.28
INQ[NB][R]	97.50	99.12	98.47	0.28
INQ[B][P]	97.63	99.16	98.55	0.29
INQ[B][R]	97.59	97.64	97.51	0.28
INQ(CNN)[P]	98.78	99.46	98.53	0.32
INQ(FC)[P]	98.93	99.43	98.63	0.32
INQ-Binary[P]	16.36	17.52	17.62	0.30

Table 1. Captures results of INQ quantization on CNN+FC only model for 3 WB. P indicates pruning strategy and R indicates random strategy. NB indicates excluding bias and B indicates including bias. CNN indicates only CNN layer has been quantized. FC indicates only FC layer is quantized.

Model	Test Acc.	Train Acc.	Valid Acc.	Time (hrs.)
Baseline	96.97	97.18	97.04	0.40
INQ[NB][P]	94.62	97.93	97.80	2.28
INQ[NB][R]	93.99	98.28	97.99	1.55
INQ[B][P]	87.82	97.83	97.60	1.57
INQ[B][R]	91.74	98.39	98.23	1.57
INQ(CNN)[P]	98.21	98.53	98.23	1.56
INQ(CNN+RES)[P]	98.54	98.86	98.52	1.56
INQ(FC)[P]	98.42	98.64	98.38	1.56
INQ(FC+RES)[P]	98.40	98.68	98.43	1.55
INQ-Binary[P]	10.20	48.42	48.30	1.56

Table 2. Captures results of INQ quantization on CNN+RES model for 3 WB. P indicates pruning strategy and R indicates random strategy. NB indicates excluding bias and B indicates including bias. CNN indicates only CNN layer has been quantized. FC indicates only FC layer is quantized. CNN+RES indicates only CNN and RES layers have been quantized. FC+RES indicates only FC+RES layers have been quantized.

The CNN model (refer Table 1), with a focus on quantizing only the fully connected (FC) layer (INQ(FC)[P]), demonstrated the highest test accuracy (98.93%) and a good balance between training (99.43%) and validation accuracy (98.63%), suggesting minimal overfitting. This configuration slightly outperformed quantizing only the convolutional (CNN) layers (INQ(CNN)[P]). Interestingly, includ-

ing bias in the quantization process led to marginally better results than excluding it, albeit with a slight increase in training time. The INQ-Binary[P] model, using binary quantization, performed poorly, indicating its unsuitability for this task.

The CNN+RES model (refer Table 2), incorporating residual connections, generally outperformed the CNN model in terms of test accuracy. Quantizing both CNN and residual (RES) layers together (INQ(CNN+RES)[P]) yielded the highest test accuracy (98.54%), followed closely by quantizing only the FC and RES layers (INQ(FC+RES)[P]). Similar to the CNN model, including bias slightly improved performance compared to excluding it. The INQ-Binary[P] model again performed poorly. Notably, the training time for the CNN+RES models was considerably longer than for the CNN models, suggesting increased computational demands due to the residual connections.

Model	Test Acc.	Train Acc.	Valid Acc.	Time (hrs.)
CNN BC[S]	91.4	91.29	90.5	0.06
CNN BC[D]	98.45	98.82	98.14	0.06
CNN BC[I]	82.36	82.78	82.61	0.08
CNN+RES BC[S]	81.46	81.47	80.84	0.62
CNN+RES BC[D]	91.72	92.03	91.70	0.60
CNN+RES BC[I]	95.89	95.83	95.52	0.64

Table 3. Captures results of BC quantization on CNN and CNN+RES model. S indicates first half of the network is quantized, D indicates only the second half of the network is quantized, I indicates incrementally all layers have been quantized.

For the standard CNN (refer Table 3), quantizing only the second half (BC[D]) yields the highest test accuracy (98.45%), suggesting that the initial layers benefit from full precision. This aligns with the understanding that shallow CNN layers capture low-level features that are more sensitive to precise representations. However, quantizing the first half (BC[S]) still results in a respectable 91.4% accuracy, demonstrating the potential for compression. Interestingly, incrementally quantizing all layers (BC[I]) leads to the lowest accuracy (82.36%), hinting at the importance of preserving precision in certain layers. In the CNN+RES model, the highest test accuracy (95.89%) is achieved when incrementally quantizing all layers (BC[I]), unlike the CNN. This implies that the residual connections might mitigate quantization loss. The residual connections may also help in effective gradient propagation during training. Quantizing only the second half (BC[D]) or the first half (BC[S]) results in lower accuracy, suggesting that the entire model might benefit from a balanced quantization approach. Notably, the training and validation accuracies are closely aligned across both models, indicating minimal overfitting. This suggests that quantization acts as a form of regularization, potentially impacting the model's capacity to memorize training data. However, the CNN+RES model generally requires significantly longer training times, possibly due to its deeper architecture and residual connections.

5.2. Weight and bias expressiveness

To investigate accuracy loss during quantization, we evaluated the weight and bias distributions of INQ-Binary models against full-precision 32-bit (FP32) models. Figure 1 illustrates these distributions for the second convolutional layer of a CNN, comparing FP32 with a quantized model (WB=3 bits), both with and without bias quantization. Notably, Table 1 shows only a marginal decrease in test accuracy after quantization. INQ-Binary employs variable-length encoding to represent weights and biases as zero or powers of two. For WB=3, this allows a maximum of nine distinct values: 0, $(\pm 2^{-1})$, $(\pm 2^{-2})$, $(\pm 2^{-3})$, $(\pm 2^{-4})$.

However, our analysis reveals that only three values are needed for weights and five for biases, with a significant concentration of weights at zero. This suggests that the model's expressive power is concentrated in a few non-zero weights and biases, achieving comparable accuracy to the FP32 model with a reduced representation. Attempts to further compress the model using binary quantization (INQ-Binary with WB=1) resulted in a catastrophic accuracy loss of approximately 90%, irrespective of whether bias was included. This indicates that binary representation severely limits the model's expressiveness. Notably, the previously predominant zero weights were replaced by either +1 or -1, failing to compensate for the information loss incurred during quantization. The following section explores this phenomenon in greater depth.

5.3. Non-uniform contribution of weight and bias

While the previous section highlighted the limited expressiveness of binary weights and biases, we now examine where the catastrophic loss occurs during bit-width reduction. Figure 2 depicts the training accuracy curve for a binary INQ-Binary model, revealing a sudden collapse at around 10% of weight quantization.

INQ-Binary's group quantization, implemented with either Gaussian or random selection, aims to distribute information uniformly across the weight and bias tensors. However, the early accuracy drop suggests that information remains concentrated in a small subset of parameters, and cross-layer quantization fails to redistribute this information effectively. This observation aligns with findings in [9], which also reported concentrated weight distributions across layers.

Table 3 further supports this notion. Quantizing only the first half of the model results in 91.40% test accuracy, while quantizing the second half yields 98.45%. This demonstrates that the model's expressive power is not evenly dis-

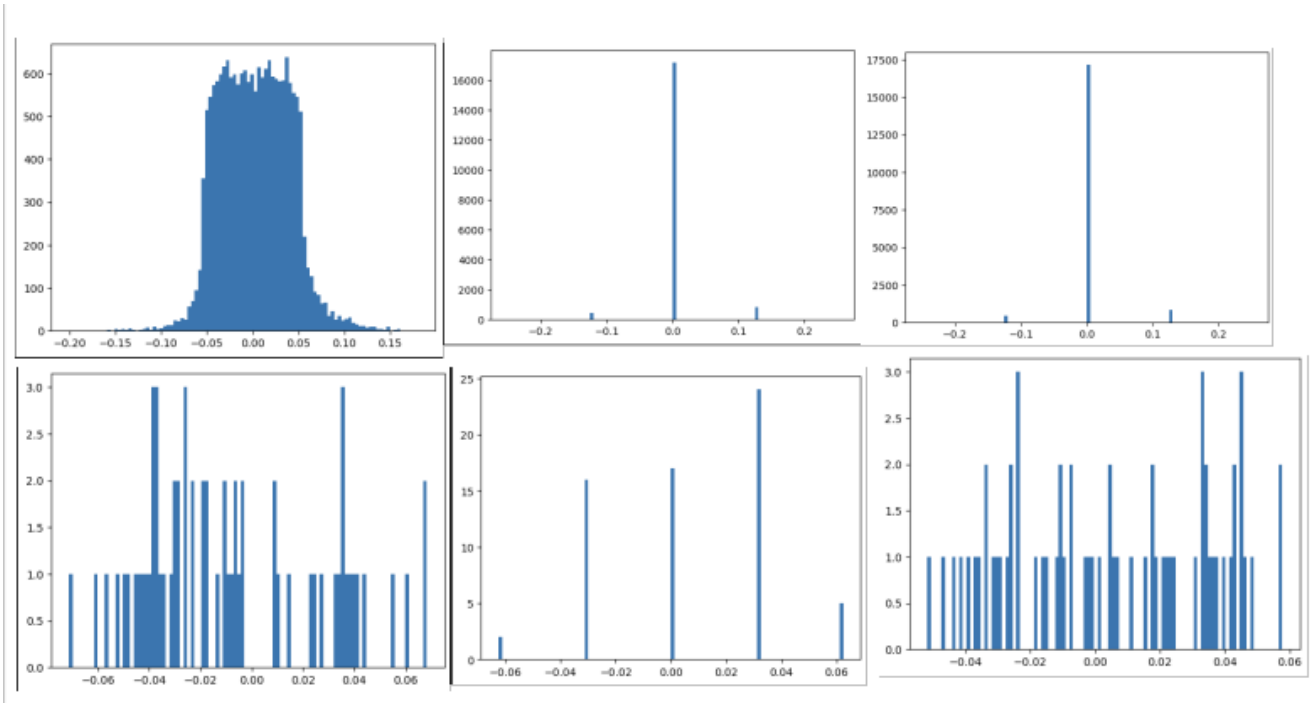

Figure 1. All stats extracted from second CNN layer in the CNN model comparing with FP32 model against weight bit=3 (WB3) quantized model, left upper: FP32 weight distribution , left lower : FP32 bias distribution, middle upper : WB3 quantized weight distribution, middle lower : WB3 quantized bias distribution, right upper : WB3 quantized weight distribution, right lower : FP32 bias distribution. Although WB=3 could represent at most 9 different values, quantized weight only used 3 and bias need 5 to achieve similar accuracy as FP32, exclude bias to quantization lead to same weight distribution but increase varity in bias distribtution.

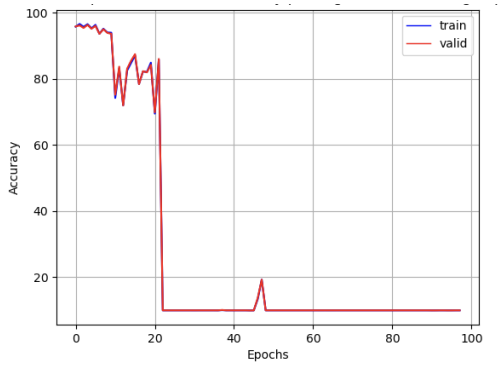

Figure 2. Binary INQ group quantization using bit width=2, and 50 quantization groups with each group training of 2 Epochs showed the accuracy collapse at about 10% of the parameter being quantized (50 groups x 2 epoch per group = 100 epochs)

tributed across layers. A layer-wise incremental retraining approach for INQ-Binary, where only a subset of layers is quantized at each step, significantly improves accuracy compared to the per-model quantization scheme.

5.4. Visualization of quantized model

CKA [6] is used in this experiment to compare different quantized model to compare the latent feature spaces' difference. In figure 4, bit width = 3 quantized model including bias term and excluding bias term were compared to observe the similarity in all layers latent feature space , the result showed no similarity in the two models, which after combining result from section 5.2 and Figure 3 showed that with limited bit width representation, model's layer distribution will vary signicantly if bias term were included vs excluded. Albeit both representation can predict to almost as good as FP32 model, but the distribution of the weight and bias tensor has changed completely.

To inspect how does the representation power get transferred to quantized model, we leveraged GradCAM [10] implementation [7] to visualize the activation map for each layer before and after quantization, in Figure 3, four experiments result were compared, left is FP32 model and middle left is from bit width of 5 quantized model and middle right is from binary quantized model with full layer wise incremental retraining, right is just entire model binary quantization. It can be observed that the activation from FP32 to bit width of 5 slightly shifted, this can show the quantized model adapting changed bit width, and was redistributing weights, however, in binary quantization, the activation reduced its intensity as well as lost focus on previous quantized model's activating location, indicating the model has some difficulty to adjust with limited representation power.

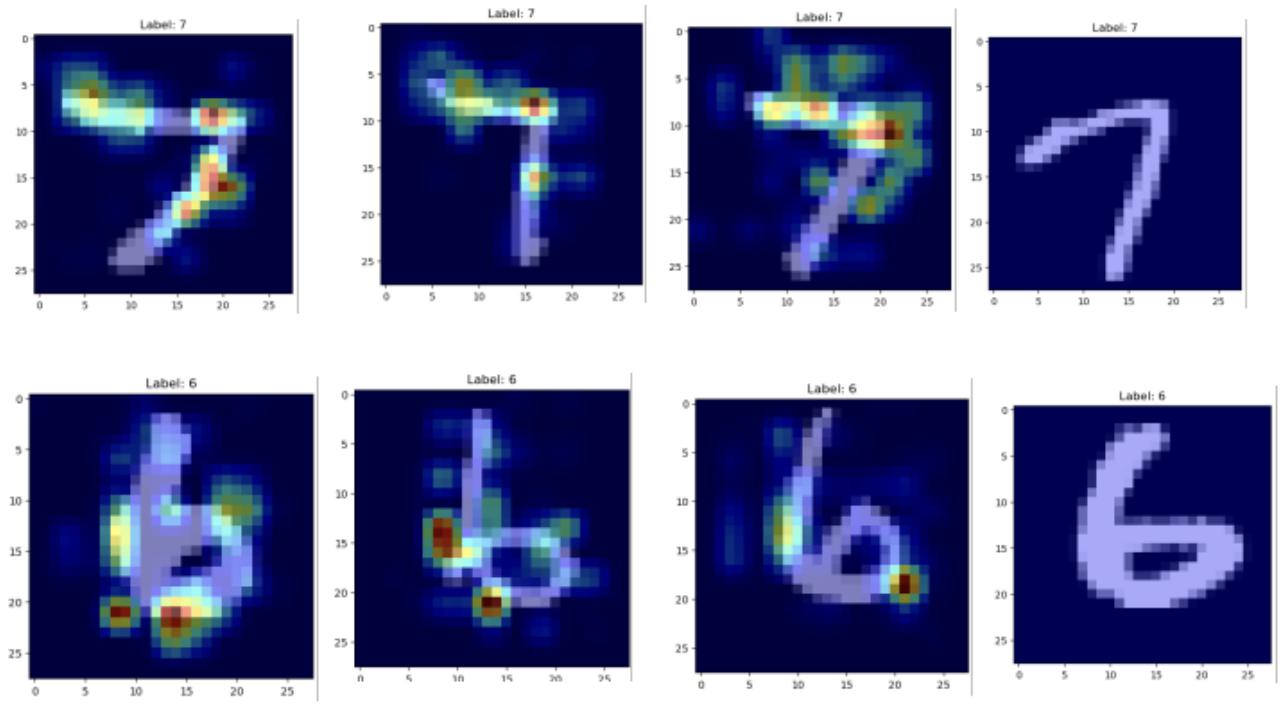

Figure 3. Activation of second CNN layer in the CNN model for digits ”6” and ”7” from models from left to right : FP 32, WB=5 INQ , INQ-Binary with model weight retraining starting from WB=2, WB=2 INQ algorithms with one shot conversion into +1/-1

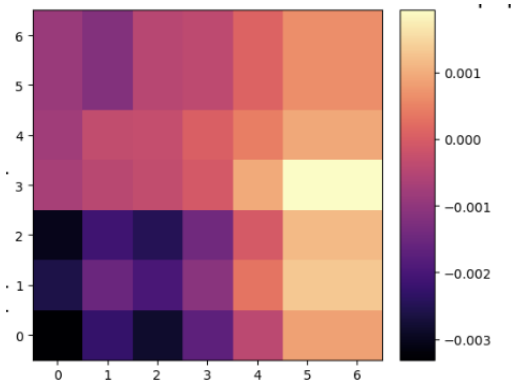

Figure 4. CKA analysis on INQ-Binary excluding bias vs INQ-Binary including-bias per tensor retraining statistics

In the last when model cannot produce any strong response and stuck in a total confusion.

6. Conclusion

A common theme across both INQ and BC quantization is the close alignment between training and validation accuracy, suggesting minimal overfitting. This could imply that quantized networks are not inherently more prone to overfitting and might simply require more extensive training to reach their full potential. This observation warrants further exploration, potentially by training quantized models for longer durations or employing techniques like early stopping to prevent overfitting.

Further research directions for optimizing these models include fine-grained analysis of quantization layer placement and exploring alternative quantization methods (DoReFa-Net [12], PACT [3]) across more diverse (e.g. multimodal) datasets to identify the optimal balance between compression and accuracy. This could involve experimenting with mixed precision quantization, varying quantization bit widths, and refining binary quantization techniques for improved accuracy. Additionally, evaluating quantized models on hardware accelerators, testing on larger and more diverse datasets, and exploring techniques for accelerating training times (e.g., distributed training or model parallelism) for the CNN+RES model could provide valuable insights for further advancements.

References

[1] Maxim Bonnaerens. Inq pytorch. https://github.com/Mxbonn/INQ-pytorch, 2020. 3

[2] Ghouthi Boukli Hacene. Binary connect. https://github.com/eghouti/BinaryConnect, 2020. 3

[3] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for

quantized neural networks. In *International Conference on Learning Representations*, 2018. 6

[4] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. *Advances in neural information processing systems*, 28, 2015. 2

[5] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 1

[6] Kim Dongwan and contributors. Pytorch library for cam methods. https://github.com/numpee/CKA.pytorch, 2022. 5

[7] Jacob Gildenblat and contributors. Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam, 2021. 5

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[9] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. arxiv 2021. *arXiv preprint arXiv:2106.08295*. 4

[10] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 5

[11] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards lossless cnns with low-precision weights. In *34th International Conference on Machine Learning*, pages 3004–3012, 2017. 2

[12] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. In *Advances in Neural Information Processing Systems*, 2016. 6