What Causes a Disparate Impact in a Quantized Model?

Abhimanyu Bellam
Computer Science
North Carolina State University

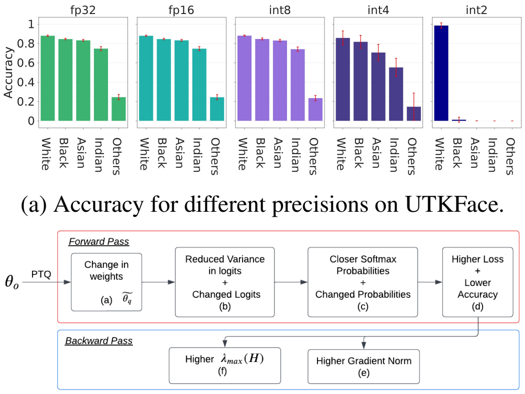
Jung-Eun Kim[*]
Computer Science
North Carolina State University

Abstract

Post Training Quantization (PTQ) is widely adopted due to its high compression capacity and speed with minimal impact on accuracy. However, we observed that disparate impacts are exacerbated by quantization, especially for minority groups. Our analysis explains that in the course of quantization, the changes in weights and activations cause cascaded impacts in the network, resulting in logits with lower variance, increased loss, and compromised group accuracies. We extend our study to verify the influence of these impacts on group gradient norms and eigenvalues of the Hessian matrix, providing insights into the state of the network from an optimization point of view.

1 Introduction

With the onset of edge devices running deep neural networks for various tasks ranging across several domains, the demand for faster computation and model lightness has become more pronounced. To aid this, compression methods such as pruning [5] and quantization [8] have taken the lead, producing little to no loss of accuracy with considerable memory and speed gains. Notably, [11] demonstrated that quantization outperforms pruning-based strategies when similar model sizes and resource footprints are considered. Furthermore, quantization is prominent for Large Language Models (LLMs) due to their large parameter sizes and requirement for reduced energy consumption [10, 4, 2]. Nevertheless, these methods do not account for the possible disparate impact they cause, and have been shown to have adverse effects on minority groups and exacerbate the shortcomings of their dense, counterpart model [14, 7].



(a) Accuracy for different precisions on UTKFace.

(b) The impact flow of quantization.

Figure 1: Group accuracy changes & impact flow.

We observed that quantization can exacerbate disparity of a model, especially for the minority group (Fig. 1a). The leftmost chart is pre-quantization. As the precision is reduced, the disparity is exacerbated further. When the model is quantized to `int2`, the disparity is extreme. In this work, we identify the factors impacting the disparity and optimization state via forward and backward passes.

Post Training Quantization (PTQ) modifies the weights of the network while setting several weights to absolute zeros, thereby, inducing sparsity, which together brings in disparate impacts of a model. Consequently, the logits suffer from a reduction in variance, similar to using high temperature scaling, while undergoing magnitude changes that lead to misclassifica-

[*] Correspondence.

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).

tions. These factors finally alter the softmax probabilities and skew their distributions closer to the decision boundary towards low confidence regions, causing higher loss and group disparity. Also, PTQ shifts the model to a worse position in the optimization space, with larger gradient norms and eigenvalues of the Hessian matrix for minority classes, implying a potential for further optimization.

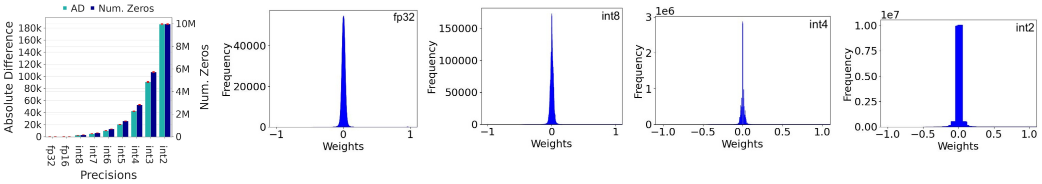
2 Problem Formulation

Consider a classification task involving a dataset D with M input samples $X = \{x_1, x_2, \cdots, x_i, \cdots, x_M\}$ and corresponding classes $Y = \{y_1, y_2, \cdots, y_i, \cdots, y_M\}$ where $y_i \in G$ groups (classes). The objective is to learn a classifier f_θ with parameters $\theta \in \mathbb{R}^K$, where K is the number of parameters in the network. The risk function obtained by using cross-entropy as the loss function to measure the discrepancy between the predicted and actual labels under empirical risk minimization (ERM) [15] is: $L(\theta; D) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{g=1}^{G} y_{ig} \cdot \log(p_\theta(x_i))_g$, where $p_\theta(x_i) = \sigma(f_\theta(x_i))$ and $\sigma(z_g) = \frac{e^{z_g}}{\sum_j e^{z_j}}$. The best solution to this optimization problem is given by, $\theta_o = \underset{\theta}{\text{argmin}}\ L(\theta; D)$. Note that this definition pertains to an uncompressed model. Subsequently, let θ_q be the weights of a quantized network such that $\theta_q = T(\theta_o)$, where T is a quantization function and q is the number of bits used to represent the weights of the network. For example, if the network was quantized to use 8-bit representations, the network parameters are denoted by θ_8. Let $\widetilde{\theta_q}$ denote the dequantized weights obtained by scaling θ_q to floating point numbers, $\widetilde{\theta_q} = S.\theta_q$, where S is the set of scaling factors. As a result, the risk functions for the original and compressed models are given by $L(\theta_o; D_g)$ and $L(\widetilde{\theta_q}; D_g)$, respectively.

3 Factors Impacting Fairness

The impact of quantization occurs through multiple stages, as shown in Fig. 1b. During the forward pass, the effect of the changes in weights propagates throughout the network and leads to changes in logits, whose behavior is reflected in the softmax probabilities and, therefore, the loss. To better understand and visualize the effects of higher loss on the network weights, we use backpropagation without actually updating the weights, motivated by the second order Taylor Series expansion of the loss function at point x_c, $L(x) = L(x_c) + \nabla L(x_c) \cdot (x - x_c) + \frac{1}{2}(x - x_c)^T H(x - x_c)$. Here, ∇L represents the gradient G. Now, for every group and precision, we study the gradient norm $||G_g^L||$ and the largest eigenvalue of the hessian matrix $\lambda_{max}(H_g^L)$ for the loss function L. The gradient norm helps us understand how far away the solution is from a better state in the solution space. Whereas, eigenvalues of the Hessian matrix provide crucial information about the steepness in the loss surface. Quoting from [14], the maximum of the eigenvalues indicates how well the solution can separate the groups. [9, 12] support that top eigenvalues aid in understanding the loss landscape.

Changes in Weights The root cause of the impact flow of quantization is the change in weights of the network. The absolute difference in the weights is measured as, $AD = \sum_{k=1}^{K} |\widetilde{\theta_{q,k}} - \theta_{o,k}|$. However, the impact does not only include the absolute difference, but also involves the fraction of "zero" weights induced by quantization. While the former quantifies how much the weights have deviated from the original values, the latter is indicative of the loss of information due to sparsity, measured by $\frac{1}{K}\sum_{k=1}^{K} I(\theta_{.,k} = 0)$. Here, I denotes the indicator function, $\theta_{.} \in \{\theta_o, \theta_q\}$. The absolute difference is controlled by the reduction in the precision of the weights. For example,



(a) Changes in weights

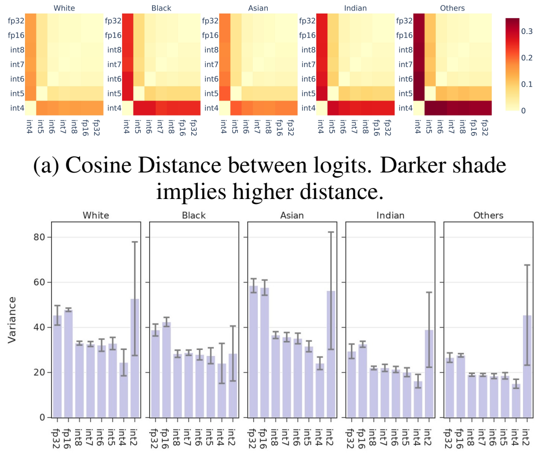
(b) Weight Distributions

Figure 2: The first piece of the impact flow: changes in weights

θ_4 has 28 lesser bits to represent the weights in comparison to θ_{32}, which persists even after scaling by S. Whereas, sparsity increases when the weights are pushed to the '0 bin' during quantization which continues to remain as 0s even after scaling. While achieving higher compression, this effect is similar to (unstructured weight) magnitude pruning [6, 18, 3], where some of the weights of the network are changed to 0.

Fig. 2a illustrates an increase in both absolute difference in weights and sparsity as precision reduces. On the other hand, Fig. 2b shows the weight distribution of $\widetilde{\theta_q}$ for different precisions, indicating a distribution shift towards the center with reducing precision. Clearly, reducing the precision increases the sparsity of the network, therefore, making it more like a pruned network (by weight magnitude).

The Effect on Logits and Probabilities We study the change in numerical values using cosine distance, defined as, $CD(A, B) = 1 - \frac{A \cdot B}{\|A\| \cdot \|B\|}$, where A and B are two vectors of equal length. Let the average cosine distance between $f_{\widetilde{\theta_q}}$ and f_{θ_o} across the samples of a group be, Average cosine distance $= \frac{1}{|G|} \sum_i^{|G|} CD(f_{\widetilde{\theta_q}}(x_i), f_{\theta_o}(x_i))$. Fig. 3a shows that the angle between different quantization levels is largest for the minority class `Others` and the least for the majority class `White`. CD captures the changes that occur in the logits due to quantization. Note that we are not able to show θ_2 and θ_3 as they produce null vector logits for some images which makes cosine distance inapplicable. The variance in the logits or softmax probabilities indicates how well the model has learned to differentiate between the groups. The mean variance among logits within each group, represented as, Mean variance of logits $= \frac{1}{|G|} \sum_i^{|G|} \text{Var}(f_\theta(x_i))$, decreases with decreasing precision, as observed in Figure 3b. Notably, the group `White` exhibits the highest variance, while the `Others` group demonstrates the least variance. This reduction indicates that the separability of groups worsened due to quantization.

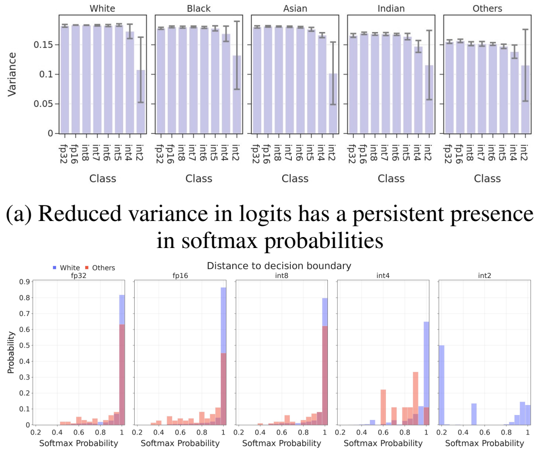


(a) Cosine Distance between logits. Darker shade implies higher distance.

(b) Decrease in precision leads to variance drop in logits

Figure 3: Logits analysis

At lower precisions, there is a substantial decrease in variance across all groups, with the `Others` group being affected the most, as illustrated in Fig. 4a. This reduced variance is analogous to the output-softening nature of the high-temperature scaling, which softens the logits of the network. Further, the disruption in the softmax probability distribution links to the inability of the precision to capture the original model's behavior. The softmax probability can be viewed as a Distance To the Decision Boundary ($DTDB$). We define $DTDB_{i,g}$ as the softmax probability obtained for each sample i belonging to group g, and that is plotted in Fig. 4b. If $DTDB_{i,g} > DTDB_{i,g'}$, then group g is farther away from the decision boundary than g', which implies an easier classification. Fig. 4b shows a strong leftward shift of distribution for `Others` unlike `White`, indicating that reduced precision induces uncertainty in the model for minority classes.



(a) Reduced variance in logits has a persistent presence in softmax probabilities

(b) The probability distribution of the distance to decision boundary (softmax probability).

Figure 4: The effect on Softmax probabilities

Contribution to Loss and Accuracy The reduced variance in softmax probabilities, together with the changed values, adversely affect the loss and accuracy of the model as depicted in Fig. 5.

The per-group loss is highest for `Others` and least for `White`. In addition, it is reflected as a direct impact on the accuracy of the model, as observed in Fig. 1a. These circumstances indicate a clear, unfavorable movement of the model in the optimization space for all the classes, due to quantization, with the most affected being `Others`. To better understand this degraded position, we backpropagate the loss and observe how the gradient norm and Hessian are affected.

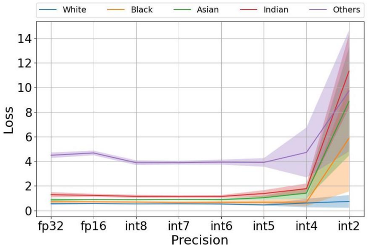
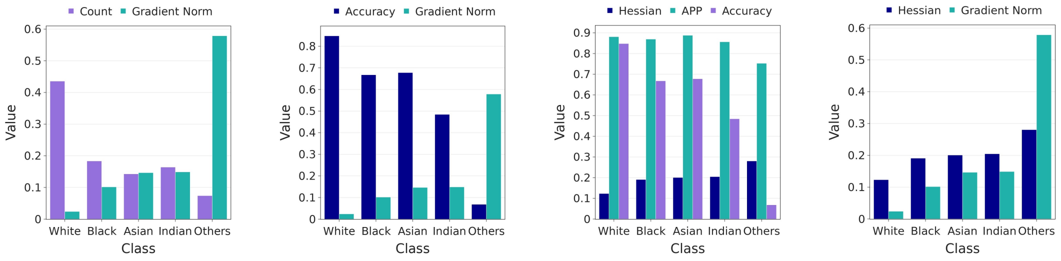


Figure 5: Higher group loss for `Others` after PTQ

Observing Unfairness through Gradient norms The gradient norm provides insight into the convergence of the optimization problem, indicating the proximity of the solution in the optimization space to a local minima [17]. We find the group gradient norm for a quantized network using the gradients obtained by passing the test set (without weight updates) and evaluating the ℓ_2 norm, given by, $G(\widetilde{\theta}_q; D_g) = \sqrt{\sum_{k=1}^{K} \left(\frac{\partial L(\widetilde{\theta}_q; D_g)}{\partial \widetilde{\theta_{q,k}}} \right)^2}$. This measure also signifies the extent of gradient updates necessary for the model to improve its prediction. Consider the situation when D is passed as a single batch for gradient updates. Initially, the averaged gradients are dominated by classes with a higher number of samples. This effect persists even when there are mini batches, although lower in impact. Therefore, the gradients are also controlled by the class distributions and batch size. In addition, the initial gradients are heavily dependent on the initialization of θ. However, the effects of batch size (if moderate) and initialization dampen as the network trains further. We therefore look at the effects of per group sample counts of the test set on the gradient norm. Fig. 6a shows an inverse trend between the gradient norm and group sizes for θ_4. Notice the huge disparity between the gradient norm of `White` and `Others`. It further reflects an inverse trend with the accuracy of the model in Fig. 6b.

Reflection of unfairness on the Hessian $\lambda_{max}(H_g^l)$ helps explain the steepness of the loss surface at that point in the solution space for a particular group. Fig. 6c shows that λ_{max} and accuracy move in opposite directions, indicating a larger λ_{max} for the minority group. This implies that the steepness is the highest for `Others`, and a corresponding update to the weight would cause a higher reduction in the loss as compared to any other group. To capture the average of the highest softmax prediction probabilities across the groups, we define, Avg. prediction prob.$(APP) = \frac{1}{|G|} \sum_{i}^{|G|} max(\sigma(f_\theta(x_i)))$ We also observe in Fig. 6c that the average prediction probability is lowest for the group with the highest λ_{max} and vice versa. Fig 6d shows gradient norm and λ_{max} moving toward the same direction, indicating that quantization induces a combined effect on them.



(a) Gradient norm vs. group size

(b) Gradient norm vs. accuracy

(c) λ_{max} is inverse to accuracy and APP

(d) λ_{max} aligns with the gradient norm

Figure 6: Trends of gradient norm and λ_{max} against group size (normalized) and accuracy ($\widetilde{\theta}_4$)

4 Conclusion

The disparate impact caused by PTQ is explained by an impact flow that passes across stages in the forward pass, whose effects can be visualized as a shift of the model to a sub-optimal state in the optimization landscape, using gradient norms and eigenvalues of the Hessian matrix. Future work will explore the effect of example difficulty and propose solutions to mitigate the disparity.

References

[1] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 7950–7958, 2019.

[2] Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: k-bit inference scaling laws. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7750–7774. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/dettmers23a.html`.

[3] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[4] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

[5] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[6] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[7] Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

[8] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *arXiv e-prints*, pages arXiv–1609, 2016.

[9] Nitish Shirish Keskar, Jorge Nocedal, Ping Tak Peter Tang, Dheevatsa Mudigere, and Mikhail Smelyanskiy. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[10] Jeonghoon Kim, Jung Hyun Lee, Sungdong Kim, Joonsuk Park, Kang Min Yoo, Se Jung Kwon, and Dongsoo Lee. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. *arXiv preprint arXiv:2305.14152*, 2023.

[11] Andrey Kuzmin, Markus Nagel, Mart van Baalen, Arash Behboodi, and Tijmen Blankevoort. Pruning vs quantization: Which is better? *arXiv preprint arXiv:2307.02973*, 2023.

[12] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

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

[14] Cuong Tran, Ferdinando Fioretto, Jung-Eun Kim, and Rakshit Naidu. Pruning has a disparate impact on model accuracy. In *NeurIPS*, 2022.

[15] V. Vapnik. Principles of risk minimization for learning theory. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, page 831–838, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc. ISBN 1558602224.

[16] Song Yang Zhang Zhifei and Qi Hairong. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[17] Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International Conference on Machine Learning*, pages 26982–26992. PMLR, 2022.

[18] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

A Appendix

Setup For the investigations presented in this paper, we use per tensor uniform post-training quantization (PTQ) [13] for weights, based on the implementation in [1] for integer quantization. In particular, for `fp16` experiments, we used half-precision computation from the PyTorch library. Note that the integer weights are scaled to floating points during inference. The following experiments are on UTKFace dataset [16] with the task of classifying the ethnicity using a ResNet18 architecture, where the weights are quantized to 16, 8, 4, and 2 bits. The original network's precision is 32 bits.