An Underexplored Dilemma between Confidence and Calibration in Quantized Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Modern convolutional neural networks (CNNs) are known to be overconfident in terms of their calibration on unseen input data. That is to say, they are more confident than they are accurate. This is undesirable if the probabilities predicted are to be used for downstream decision making. When considering accuracy, CNNs are also surprisingly robust to compression techniques, such as quantization, which aim to reduce computational and memory costs. We show that this robustness can be partially explained by the calibration behavior of modern CNNs, and may be improved with overconfidence. This is due to an intuitive result: low confidence predictions are more likely to change post-quantization, whilst being less accurate. High confidence predictions will be more accurate, but more difficult to change. Thus, a minimal drop in post-quantization accuracy is incurred. This presents a potential conflict in neural network design: worse calibration from overconfidence may lead to better robustness to quantization. We perform experiments applying post-training quantization to a variety of CNNs, on the CIFAR-100 and ImageNet datasets, and make our code publicly available.[1]

1 Introduction

We want predictive models to produce reliable uncertainty estimates, such that better informed decisions can be made based on their predictions. This is especially important in safety-critical applications such as autonomous driving [1] and medical diagnosis [2]. One common way of measuring the quality of predictive uncertainty is model calibration, which in the case of a classification task can be understood as how well the probabilities predicted by a model match its accuracy. For example, a well-calibrated model should be correct 70% of the time, when it predicts a probability of 0.7 for the chosen class. Thus, for a K-class classification problem, we can define a *perfectly calibrated* model [3, 4] as,

$$P(y = \hat{y}|P(\hat{y}|\boldsymbol{x}; \boldsymbol{\theta}) = p) = p, \quad p \in [0, 1], \tag{1}$$

where $\boldsymbol{\theta}$ are the model parameters, $\boldsymbol{x}$ is an input and $y \in \{\omega_k\}_{k=1}^{K}$ is the corresponding class label. $\hat{y}$ is the class predicted by that model and is given by $\hat{y} = \arg\max_{\omega} P(\omega|\boldsymbol{x}; \boldsymbol{\theta})$, and $P(\hat{y}|\boldsymbol{x}; \boldsymbol{\theta})$ is its corresponding predicted probability, or the model confidence for input $\boldsymbol{x}$. In the case of a CNN, probabilities will come from the softmax after the final fully connected layer (or logits).

It has been previously shown that modern convolutional neural networks (CNNs), such as ResNets [5], exhibit poor calibration, and in fact tend to be overconfident [3, 4]. The model confidences are greater than the actual test accuracies achieved, making them unreliable.

On the other hand, neural network quantization is an optimization technique where weights

[1] Code to be released on Github after review.

Under review at ICBINB Workshop @ NeurIPS 2021. Do not distribute.

and activations are stored in a lower bit numerical format compared to what they were trained in, e.g. int8 vs fp32 [6, 7, 8]. This can allow significant reductions in computational and memory costs, resulting in lower latency and energy consumption. Quantization is thus often important for deploying CNNs on resource limited platforms such as mobile phones. In this paper we examine post-training quantization (PTQ), where a fully trained full precision network is mapped to lower precision without further training. It has been noted that CNNs are robust to the noise introduced by quantization; however, to the best of our knowledge, research tends to focus on improving robustness [8, 9, 10] rather than explaining why architectures seem to be inherently robust. There has been other research into understanding how compression methods affect accuracy [11], however, it primarily focuses on the effect of pruning and does not consider predictive uncertainty.

In this paper we highlight a novel insight, that links the above two concepts (calibration, quantization) in deep learning together. We find that

- confidence and calibration are closely linked to the robustness of model accuracy to post-training quantization and,
- overconfidence can potentially improve the robustness of accuracy.

2 How does calibration affect accuracy post quantization?

We can consider the effect of post-training quantization on a CNN's activations and ultimately outputs as adding noise to the original floating point values [9, 12]. Thus, we can separate two questions that determine the change in accuracy after quantization:

1. How easy is it to change the predicted class?
2. Given the prediction has changed, how will the model accuracy be influenced?

In order for a prediction to change, for standard classification CNNs, the quantization noise needs to be sufficient to change the top logit. Intuitively, this suggests that less confident predictions will be easier to change, as their top logit will be closer to the other logits. We would also expect lower precision, and thus greater quantization noise, to result in more swapped top logits as well. We do not investigate the above in detail, as this would require accurately modelling the distribution of logits post-quantization for different architectures conditional on the input. However, we simply state that our empirical results support the intuition presented above.

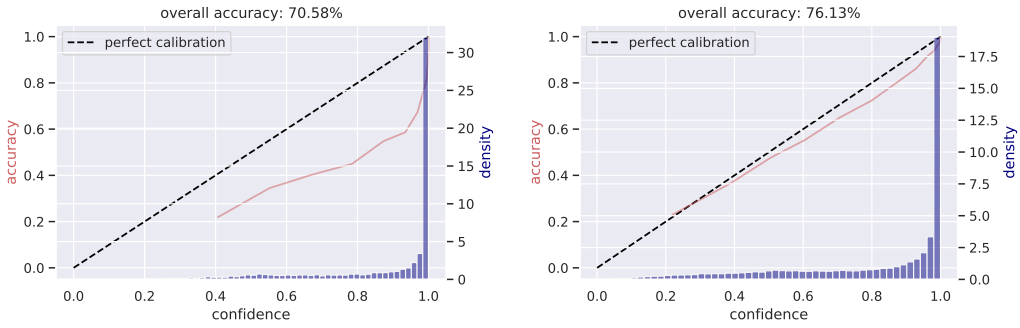
The second question can be directly linked to the calibration of the model, as calibration tells us about the accuracy of predictions at a given confidence. For example, considering a well-calibrated model as defined in Equation 1, for a prediction with confidence $P(\hat{y}|\boldsymbol{x};\boldsymbol{\theta}) < 0.5$ the probability that it will be correct is also < 0.5. This means that it is more likely to already be wrong. If the prediction is in fact incorrect then it will not cause a decrease in the overall accuracy if it changes, since it will either change to another incorrect class, or to the correct class.

It can now be seen how overconfidence, where the model is more confident than it is accurate, $P(y = \hat{y}|P(\hat{y}|\boldsymbol{x};\boldsymbol{\theta}) = p) < p$, may improve robustness to post-training quantization. For predictions with $P(y = \hat{y}) > 0.5$, these are more likely to be correct in the first place, and so it would be better for these to stay the same post quantization. Thus, having a higher confidence would be beneficial. Conversely, for more easily swapped predictions with lower confidence, if the original accuracy of these is lower, then the change in accuracy post quantization will be less.

3 Experiments

We present experimental results for ResNet56 and ResNet50 [5] trained on CIFAR-100 [13] and ImageNet [14] respectively. Additional results for ResNet20 on CIFAR-100 and MobileNetV2 [15] on ImageNet are available in Appendix A. CIFAR models were trained using the regime specified by [5], whilst ImageNet models use pretrained weights available from PyTorch[2] [16].

[2] `https://pytorch.org/vision/stable/models.html`



(a) ResNet56, CIFAR-100

(b) ResNet50, ImageNet

Figure 1: Reliability curves and confidence histograms for full precision (pre-quantization) networks on the corresponding test datasets. ResNet56 on CIFAR-100 is both very overconfident in terms of calibration (confidence $>$ accuracy), and highly confident in general. ResNet50 on ImageNet shows similar behavior but to a lesser extent.

For weights we use uniform per-channel symmetric quantization, where the quantization parameters are determined using minimum and maximum values. For activations we use uniform per-tensor asymmetric quantization, where the quantization parameters are found using PyTorch's default histogram based method that iteratively aims to minimize the mean squared error from quantization [8, 16]. Batchnorm layers are folded into the preceding convolutional layer. This is a relatively standard scheme, as our aim is not to achieve the best performance, but to examine behavior as quantization noise varies. Quantized inference is simulated in PyTorch using the existing backend for this purpose.[3]

3.1 Model calibration before quantization

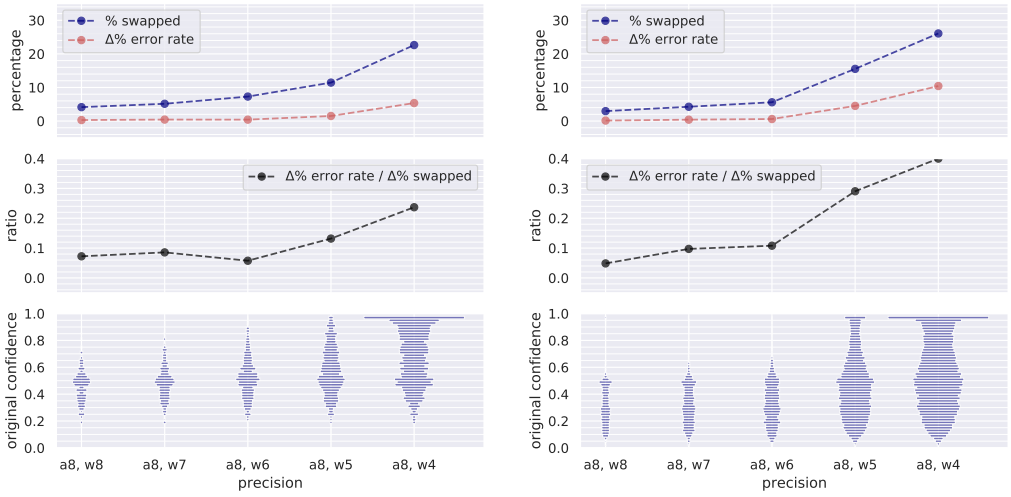
Reliability curves plot accuracy against binned confidence [4, 17], and so not only give an idea of the calibration error, but also of whether a model is over or under confident. Figure 1 shows reliability curves for floating-point models (pre-quantization), alongside histograms showing the distribution of confidence over the test datasets. It can be seen that ResNet56 on CIFAR-100 is very overconfident on the test data, with accuracy much lower than confidence. ResNet50 on ImageNet is better calibrated, but still overconfident as well. ResNet56 is also more confident overall compared to ResNet50, although both models have a large proportion of their predictions with confidence near 1.0.

3.2 Model accuracy after quantization

Given knowledge of the calibration behavior of the networks (Figure 1), we can explain the behavior of accuracy as quantization noise increases. Figure 2 shows, going from the floating point model to a quantized one, the percentage of swapped/changed predictions, the change in error rate ($1-$accuracy), the ratio of the previous two values, and histograms of swapped predictions over confidence. These are tracked as the activation precision is held constant at 8 bits and the weight precision is decreased from 8 to 4 bits, allowing the quantization noise to be varied in a controlled manner. Note that the histograms are not normalized, so the area reflects the number of swapped predictions.

It can be seen for both models that as the quantization noise increases/precision decreases, at first, the predictions to be swapped are the lower confidence ones, supporting the previously presented intuition. Moreover, even though the proportion of swapped predictions is quite high, the increase in error rate is only a small proportion of this. This is reflected in the reliability curves in Figure 1, that show that low confidence predictions are also low accuracy, and supports the reasoning outlined in Section 2. Even though post-training quantization may have caused a large number of predictions to change, the predictive accuracy of the models remains robust, as the majority of predictions that do change do not lead to an increase in error rate. As the weight precision is decreased further (and the quantization noise increases), only then do higher confidence predictions

[3] `https://github.com/pytorch/pytorch/blob/master/torch/ao/quantization/fake_quantize.py`



(a) ResNet56, CIFAR-100

(b) ResNet50, ImageNet

Figure 2: Going from floating point to quantized: percentage of swapped predictions, percentage change in error rate, their ratio, and (unnormalized) histograms of swapped predictions over confidence. These are plotted as weight (w) precision is decreased whilst activation (a) precision is held constant. Initially, as quantization noise grows the ratio stays low, as mainly low confidence (and low accuracy) predictions are swapped. The ratio starts to grow more rapidly, alongside the error rate, when higher confidence (and higher accuracy) predictions start to be swapped at higher noise levels.

start to be swapped. The ratio of change in error rate to proportion swapped increases, and this again is reflected in Figure 1, where higher confidence predictions are shown to be more accurate.

It is not straightforward to directly compare the two models, as the distribution of quantization noise and how it relates to the number of bits used to represent the network will be different between them. However, we can still observe in Figure 2 that, as more predictions in the approximate interval $[0.5, 1]$ are swapped, the ratio of change in error rate to proportion swapped increases much more quickly for ResNet50 compared to ResNet56. This can be related to Figure 1, where ResNet56 is much less accurate than it is confident in this interval, whilst ResNet50 is better calibrated. This supports the idea that overconfidence improves the robustness of accuracy to quantization.

4 Discussion

We have shown the novel insight that model calibration can help explain the robustness of CNN accuracy to post-training quantization. Low confidence predictions are more easily swapped post-quantization. However, if these predictions are low accuracy as well then the overall accuracy will not decrease by much. High confidence predictions will be more accurate, but more difficult to change. Moreover, we reason that overconfidence may improve robustness, as higher accuracy predictions will be more confident, and lower confidence predictions will be less accurate. We hope that this work can lay the groundwork for further analysis on the understanding of compressed neural networks. For example, further investigation should be done into how quantization precision affects noise on the logit level, as this work only examines behavior after the softmax.

This result raises a potential dilemma, which is not considered in current literature. As research is increasingly moving towards producing deep learning approaches with better estimates of predictive uncertainties, better calibrated models may consequently have less robust accuracies to quantization. Interestingly, our findings, as they relate to the intrinsic calibration of a trained model, do not affect methods that improve calibration post-hoc, such as Temperature Scaling [4] or Deep Ensembles [3, 18]. However, methods applied during training, such as using a soft calibration objective [19] or label smoothing [20] may be affected. Thus, a natural extension of this work would be to investigate post-training quantization on models trained using these methods.

References

[1] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," 2017.

[2] B. Kompa, J. Snoek, and A. Beam, "Second opinion needed: communicating uncertainty in medical machine learning," *NPJ Digital Medicine*, vol. 4, 2021.

[3] X. Wu and M. J. F. Gales, "Should ensemble members be calibrated?," *CoRR*, vol. abs/2101.05397, 2021.

[4] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, PMLR, 06–11 Aug 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[6] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," 2017.

[7] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," *CoRR*, vol. abs/1806.08342, 2018.

[8] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, "A white paper on neural network quantization," 2021.

[9] M. Alizadeh, A. Behboodi, M. van Baalen, C. Louizos, T. Blankevoort, and M. Welling, "Gradient l1 regularization for quantization robustness," *ArXiv*, vol. abs/2002.07520, 2020.

[10] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *ArXiv*, vol. abs/2103.13630, 2021.

[11] S. Hooker, A. Courville, G. Clark, Y. Dauphin, and A. Frome, "What do compressed deep neural networks forget?," 2021.

[12] S. Yun and A. Wong, "Do all mobilenets quantize poorly? gaining insights into the effect of quantization on depthwise separable convolutional networks through the eyes of multi-scale distributional dynamics," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2447–2456, 2021.

[13] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research),"

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[15] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[17] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," *Proceedings of the 22nd international conference on Machine learning*, 2005.

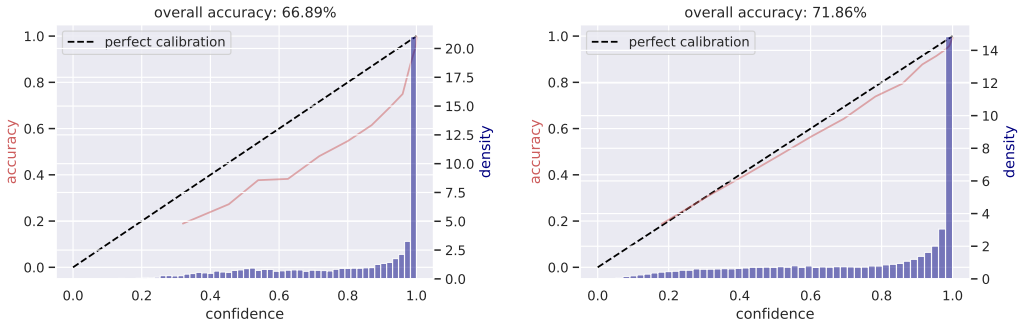
[18] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[19] A. Karandikar, N. Cain, D. Tran, B. Lakshminarayanan, J. Shlens, M. C. Mozer, and B. Roelofs, "Soft calibration objectives for neural networks," 2021.

[20] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?," in *NeurIPS*, 2019.

A Appendix

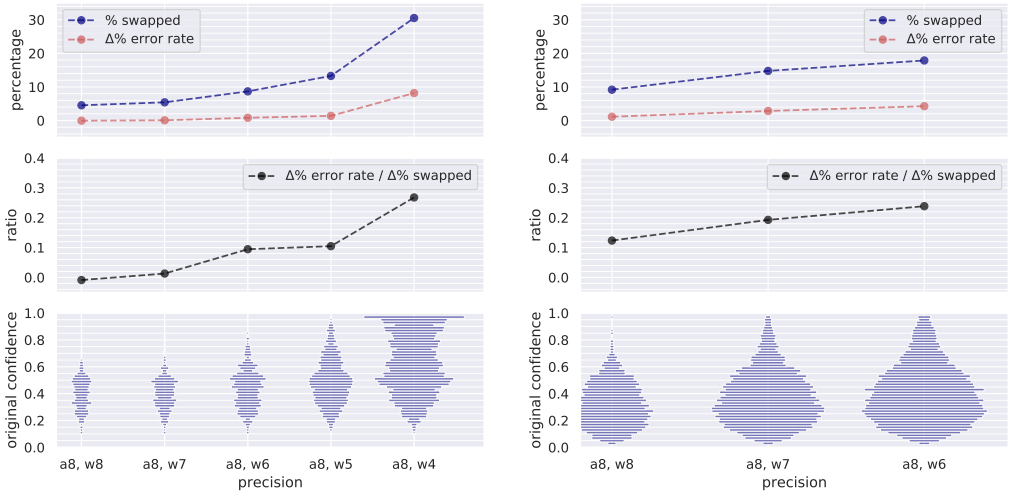
Additional experimental results are provided for ResNet20 on CIFAR-100 and MobileNetV2 on ImageNet (Figures 3 and 4). ResNet20 behaves quite similarly to ResNet56, which is to be expected as they share the same architecture. MobileNetV2 is similarly calibrated to ResNet50, but is more fragile to quantization, which is a common observation about this architecture [7, 8]. However, in Figure 4, its behavior is still consistent with the reasoning presented in this paper. There is just likely to be more quantization noise at the logit level for a given precision compared to ResNet50. Note that the figure is truncated for readability, as the change in values is significantly larger for lower precisions.



(a) ResNet20, CIFAR-100

(b) MobileNetV2, ImageNet

Figure 3: Reliability curves and confidence histograms for full precision (pre-quantization) networks on the corresponding test datasets.



(a) ResNet20, CIFAR-100

(b) MobileNetV2, ImageNet, truncated for readability

Figure 4: Going from floating point to quantized: percentage of swapped predictions, percentage change in error rate, their ratio, and (unnormalized) histograms of swapped predictions over confidence. These are plotted as weight (w) precision is decreased whilst activation (a) precision is held constant. MobileNetV2 is truncated for readability, as the values are much higher for lower precisions.