It’s All In the Teacher: Zero-Shot Quantization Brought Closer to the Teacher

Kanghyun Choi1, Hye Yoon Lee1, Deokki Hong1, Joonsang Yu2, Noseong Park1, Youngsok Kim1, and Jinho Lee1*

1College of Computing, Yonsei University 2CLOVA ImageVision, CLOVA AI Lab, NAVER
1{kanghyun.choi, hylee817, dk.hong, noseong, youngsok, leejinho}@yonsei.ac.kr
2joonsang.yu@navercorp.com

Abstract

Model quantization is considered as a promising method to greatly reduce the resource requirements of deep neural networks. To deal with the performance drop induced by quantization errors, a popular method is to use training data to fine-tune quantized networks. In real-world environments, however, such a method is frequently infeasible because training data is unavailable due to security, privacy, or confidentiality concerns. Zero-shot quantization addresses such problems, usually by taking information from the weights of a full-precision teacher network to compensate the performance drop of the quantized networks. In this paper, we first analyze the loss surface of state-of-the-art zero-shot quantization techniques and provide several findings. In contrast to usual knowledge distillation problems, zero-shot quantization often suffers from 1) the difficulty of optimizing multiple loss terms together, and 2) the poor generalization capability due to the use of synthetic samples. Furthermore, we observe that many weights fail to cross the rounding threshold during training the quantized networks even when it is necessary to do so for better performance. Based on the observations, we propose AIT, a simple yet powerful technique for zero-shot quantization, which addresses the aforementioned two problems in the following way: AIT i) uses a KL distance loss only without a cross-entropy loss, and ii) manipulates gradients to guarantee that a certain portion of weights are properly updated after crossing the rounding thresholds. Experiments show that AIT outperforms the performance of many existing methods by a great margin, taking over the overall state-of-the-art position in the field.

1. Introduction

Deep neural network quantization [14, 23, 37, 69] is a powerful tool for improving the computational efficiency of deep neural networks (DNNs). When being accompanied with the low-bitwidth hardware design [29, 50, 59], the latency and energy consumption of DNNs can be greatly reduced.

One problem of quantized models is, however, that they often suffer from the significant drop in accuracy, mainly due to quantization errors [37]. A popular way to address the problem is to further train or calibrate the model with training data [3, 9, 24, 53, 66, 69]. During the fine-tuning procedure, the forward pass is performed with quantized values whereas the backpropagation is done with floating-point values to recover the accuracy loss in the initial quantization.

Unfortunately, such fine-tuning methods, which assume the full availability of training data at the time of quantization, are often not feasible in reality. Many models are disclosed to public only with their trained weights, and the dataset may contain proprietary, confidential, or sensitive data that fundamentally prohibit sharing.

Zero-shot quantization (or data-free quantization) [4, 5, 10, 11, 45, 62, 67, 68, 70] is therefore a necessary technique for quantization. It assumes that only the architecture and the pre-trained weights are available at the time of quantization. Current successful approaches are mainly led by generative approaches [5, 10, 39, 62, 67, 70]. Using synthetic samples from generators, knowledge distillation [22] is applied against full-precision models. It is known that the state-of-the-art methodology achieves almost similar performance to that of the data-driven approaches (i.e., quantization with real samples) for 5-bit fixed-point quantization, and comparable performance on 4-bit fixed-point setting. [10].

However, the recipe of the fine-tuning in zero-shot quantization is mainly adopted from common knowledge distillation problems [8, 20, 22] that consider neither quantization nor synthetic samples. As in the knowledge distillation, the loss function of the zero-shot quantization is habitually built as a combination of the cross-entropy (CE) against the hard label and the Kullback–Leibler (KL) divergence against the full-precision network’s output.1

1In the remainder of this paper, we refer to CE as the cross-entropy against the hard label and KL as the KL divergence against the full-precision network unless otherwise stated.
in practice, but there are no detailed studies to introspect the appropriateness of the loss in the context of zero-shot quantization. Therefore, more analyses on those solutions are needed. Moreover, the distribution of synthetic samples can be different from that of the original data. In such a case, they can be considered a type of adversarial samples (also see Fig. 6 for examples) and thus, the quantized network produces a huge generalization gap.

To our knowledge, we for the first time perform in-depth analyses on the loss surface of the zero-shot quantization problem. Through the analyses, we find several key observations for better quantization. First, quantized models often have difficulty optimizing multiple loss terms, and the loss terms fail to cooperate — in other words, the angle between the gradients of CE and KL is quite large in many cases. Second, KL usually has a much flatter loss surface than that of CE, having a better potential for generalizability.

To this end, we propose a method to address such problems of the zero-shot quantization, called AIT (All In the Teacher). While pursuing a flatter surface of the loss curve, AIT lets the quantized student model get closer to the full-precision teacher model. To be more specific, we exclude CE from the loss, and apply our proposed gradient inundation with KL only. In addition, gradient inundation is designed to grow the gradients of KL in such a manner that a certain portion of weights are guaranteed to be updated in each layer. As a result, the quantized model approaches closer to the full-precision teacher, and our method takes over the state-of-the-art position for various datasets. Our contributions can be summarized as follows:

- We analyze the first and second-order loss surfaces, i.e., gradient and Hessian, of the zero-shot quantization problem. To the best of our knowledge, we are the first to closely investigate the loss function in the zero-shot quantization problem.
- We identify that the gradients from CE and KL form a large angle from the beginning to the end of fine-tuning. This implies that the quantized network is suffering from their trade-off instead of benefiting from them working in harmony.
- We analyze the local curvature of the loss surface and observe that the two losses of our interest exhibit a great amount of curvature difference.
- We observe that the quantized student suffers from infrequent updates, where only a few layers are changing their integer weights and the remaining layers are stuck below rounding thresholds.
- Based on these findings, we propose AIT which excludes the cross-entropy loss, and manipulates the gradients using our proposed gradient inundation method such that the quantized student model can faithfully resemble the full-precision teacher model.
- We perform a thorough evaluation of AIT. The results show that AIT outperforms the existing algorithms by a great margin, showing the state-of-the-art performance on the zero-shot quantization problem.

2. Background and Related Work

2.1. Quantization

Quantization of neural networks have been studied for a while, and there are numerous methods [6, 12, 17, 19, 30, 49]. In this work, we consider symmetric, uniform quantization that is known to be much easier to build hardware architectures for. With $n$ bits, a weight parameter $\theta$ is represented by one of $2^n$ ranges. We use $\theta'$ to denote the quantized weights as outputs of a quantization function $\text{Quant}()$. For $\text{Quant}()$, we use a simple yet efficient function as the following [24]:

\[
\begin{align*}
\theta' &= \text{Quant}(\theta) = \lfloor \theta \times S - z \rfloor, \\
S &= \frac{2^n - 1}{\theta_{\text{max}} - \theta_{\text{min}}}, \\
z &= S \times \theta_{\text{min}} + 2^{n-1},
\end{align*}
\]

where $S$ is the scaling factor to convert the range of $\theta$ to $n$ bit, and $z$ decides which quantized value zero is mapped to. After quantization, the quantized integer value represents $\theta' \in \mathbb{R}$ obtained by dequantization:

\[
\theta' = \frac{(\theta' + z)}{S}.
\]

The procedure is the same for activation values, except that the minimum and the maximum are obtained from observing activations from a few batches and taking a moving average.

2.2. Zero-shot Quantization

Even though quantization has been shown to be effective even for extremely low bits [14, 40, 53, 69], they usually require training data to fine-tuning or calibration. Zero-shot quantization is a method to relax the privacy or confidentiality problem of the training data.

Earlier methods for zero-shot quantization were focused on how to build a good quantization function $\text{Quant}()$, by using schemes such as weight equalization, bias correction, or range adjustments [4, 45, 68]. Among them, ZeroQ [5] was the first work to introduce the notion of distilled data that is designed to match the batch-norm stats of the original full-precision network. With this scheme, choosing the adequate mixed-precision quantization for each layer has been proposed together. On top of ZeroQ, DSG [67] added diverse sample generation to improve the performance.
and KL in CE cooperative quantization? In this work, we provide an in-depth analysis of the loss function where both utilizes cross-entropy (Sec. 3.1). Then, we investigate the difference of the local curvature with the lens of Hessian. Because the zero-shot quantization suffers from larger generalization gaps, finding the solution of the flatter minima is critical (Sec. 3.2).

3. Analyses on the Zero-shot Quantization

In this section, we provide in-depth analyses on the first/second-order loss surface of the state-of-the-art zero-shot quantization method. We studied the impact of CE and KL on the loss function of zero-shot quantization setting, and reveal that they are not cooperative but hinder each other (Sec. 3.1). Then, we investigate the difference of the local curvature with the lens of Hessian. Because the zero-shot quantization suffers from larger generalization gaps, finding the solution of the flatter minima is critical (Sec. 3.2).

3.1. Gradient Cosine Similarity

In this subsection, we attempt to find a partial answer to the question: are CE and KL cooperative in quantization? As discussed in Sec. 2.2, loss functions from current techniques for zero-shot quantization [10, 62, 70] are mainly composed of the CE against the hard labels, and KL divergence against the full-precision teacher.

However, it has been discussed that better models do not necessarily make good teachers when the student has limited capacity. In such cases, the student often has to make trade-off between KL and CE [8]. As quantized models have much lower representational capability [24, 46], it could be difficult for them to optimize both terms. Furthermore, with synthetic samples not exactly matching the distribution of the real samples, the labels associated to each sample via hard labels and teacher outputs can be distinct, contributing to the difficulty of addressing both losses.

In such regard, [16] suggested using cosine similarity of the two gradients as a metric for determining whether an auxiliary can contribute toward a single main task. The authors suggest that the two losses should be used together only in steps where their cosine similarity is larger than zero (when they form an acute angle).

Thus, inspired by the proposals of [16], we analyzed the training of the zero-shot quantization as the following. Using GDFQ [62] as a representative for zero-shot quantization, we measured the cosine similarity of the $g_{KL}$ and $g_{CE}$ while training a quantized ResNet-20 model with synthetic data. The full-precision teacher is pre-trained, and the generator is jointly trained with the quantized student using Eq. (5) and Eq. (6) with $\alpha = \delta = 0.5$, respectively. For comparison, we have also measured the same metric from a common real data (CIFAR-10) based knowledge distillation with pre-trained ResNet-20 as a teacher and random initialized ResNet-20 as a student (self-distillation) using the same loss function.

The results are presented in Fig. 1. Using the real samples denoted as ‘KD’, the KL and CE terms induce gradients of the similar direction, supporting the common wisdom that the combination of them works well in practice. However, with the zero-shot quantization setting using synthetic samples denoted as ‘ZQ’, the cosine distance between them take negative values. The bottom two plots visualize the angle of the gradients. In both plots, the $g_{CE}$ is set to (1,0), and $g_{KL}$ is plotted to preserve the relative angle to the $g_{CE}$.

The trend persists throughout the training, as shown in the plot. Right after the training begins, the cosine distance of ZQ becomes negative and it is maintained until the end of training, while that of the KD is positive. This implies that combinations of the two losses do not cooperate well with each other, and using them together could potentially harm the model performance. Although we display only one case for clarity, the same trend was observed across many models and datasets. Refer to the Appendix for further results.
3.2. Generalizability

The observation in Sec. 3.1 suggest that using only one of the two losses — KL divergence against the full-precision teacher or CE against the hard label — could be better for the problem. Some work [8] suggest modifying the teacher for distillation, but such method is unavailable in a zero-shot setting because we have no access to the training data.

In such regard, we examine the generalizability of the loss terms. Including zero-shot quantization, diverse applications relying on synthetic samples [13, 36, 38, 51, 58, 61, 64] usually suffer from huge generalization gap, coming from the discrepancy in the data distributions. One can easily infer that quantization requires stronger generalization when performed under a zero-shot environment.

To evaluate the generalizability, we measure local curvature of the loss surface. Popularly measured with the Hessian matrix $H \in \mathbb{R}^{n \times n}$, where $\theta$ is a vector of $n$ weight parameters), local curvature of the loss surface is a metric that is drawing much attention from the field, and it is thought to hold a key to better generalization [2, 7, 25–28, 31]. As illustrated in Fig. 2a, if the optimizer settles at a sharp minima, the performance at test time is likely to incur a larger degradation compared that of a flat minima. Such gap would be much larger with synthetic data incorrectly modeling the validation data distribution.

Fig. 2b plots $Tr(H)$, the trace of the Hessian matrix, approximated by PyHessian [63] implementing Lanczos algorithm [35]. We separate Hessian calculation for each of CE and KL. The trace values are significantly different, where that of the KL is much smaller than the CE. The gap is notably larger in zero-shot quantization (right, ZQ) than on real-data knowledge distillation (left, KD). In addition, the distribution of the eigenvalues displayed in Fig. 2d also show a huge difference in the local curvature of the loss terms. While CE has longer tail for high eigenvalues, those of KL has more concentration to lower eigenvalues.

This could potentially lead to two conclusions: the loss surface of the KL divergence is much flatter, or the model has converged to the minima in the loss surface of the KL divergence. However, in our case, we believe it advocates for the former, based on an auxiliary experiment. It is commonly observed that near the minima, the disparity between gradients start to arise [18, 43]. Following the same regard, we measure the cosine distance of the gradient averaged within an epoch, compared to that of the previous epoch (inter-epoch cosine similarity). As displayed in Fig. 3a, the gradients of KL in the zero-shot quantization setting (ZQ) points to a consistent direction (large cosine similarity) compared to that of the real data distillation (KD), indicating that it has not reached the minima yet.

Fig. 2c illustrates a more direct visualization of the loss surface. From the Hessian matrix, we took the largest eigenvector $e$ at each epoch, we plotted the value of CE and KL on the left plot by calculating $L(\theta(t) + k \cdot e \cdot \hat{g}(t))$ for $k \in [-0.5, 0.5]$, where $\hat{g}$ is the average gradient along $e$. The left plot presents the CE in red color scheme and KL in blue color scheme. It is clearly shown that the surface is flatter in the KL surface especially near the end of training.
3.3. Summary

Summarizing the studies in this section, we first observed that the CE and KL form a large angle in the gradient space, and the quantized model has difficulty optimizing both directions. Furthermore, by measuring the stats from Hessian matrices, we conclude that KL has a much flatter loss surface for potentially better generalization, which is an important issue for generative zero-shot quantization methods.

4. AIT Method

In this section, we describe our AIT (All In the Teacher) method in detail. From the observations in Sec. 3, we first drop CE term from the loss, and apply a novel gradient inundation to bring the quantized model as close as possible to the full-precision teacher.

4.1. KL-only Zero-shot Quantization

Motivated by the experiments from Sec. 3.2, we ran GDQ [62] with KL-only loss (i.e., $\delta = 1$). However, as will be shown later in Tab. 2, the performance severely degrades in all settings. We find an explanation from the experiments of Fig. 3a. Even toward the end of the training, the direction of the $g_{KL}$ remains consistent, and training for more epochs did not solve the problem. This indicates that the model did not converge at the minima of the KL surface.

Another set of experiments shown in Figs. 3b and 3c gives a closer look at the phenomenon. We count the average number of weight parameters that cross the rounding threshold (parameters whose quantized values have changed from the previous step). We make two observations: First, the portion of quantized values crossing the rounding threshold is extremely small. Even when training has not stabilized (epoch 60), only 0.0011% of weights are being updated each step. At a later epoch (350), the phenomenon becomes worse that only four values are updated in a whole epoch, which is only 1.8e-7% of weight updates per step during the epoch. In addition, the changes are extremely unbalanced, where all the updates are only occurring in just three layers.

We posit that this is from the quantized training process that constrains integer value updates. During the training, the quantized network internally stores its full-precision values. The parameters are quantized for the forward pass of the backpropagation, and the gradients are applied to the internal full precision values. As the gradient values become smaller after a few epochs of training, the change in the parameters are usually not large enough to cross the threshold, and only a few layers are continuously making changes, stopping the model from moving towards a lower point in the loss surface.

4.2. Gradient Inundation

To address the problem of KL-only method, we propose gradient inundation (GI). Overall, we attempt to dynamically manipulate the gradients $g_l$ of each layer $l$, such that certain number of parameters are guaranteed to update in its integer value. With stochastic gradient descent, consider the update rule of parameter $\theta_{l,k}$ at step $k$ with learning rate $\eta$:

$$\theta_{l,k+1} = \theta_{l,k} - \eta \cdot g_{l,k}.$$

with gradient inundation, the modified rule is as the following: For the parameters $\theta_{l,k}$, quantized parameters $\theta_{l,k}^q$ and the corresponding gradients $g_{l,k}$ from layer $l$,

$$\theta_{l,k+1} = \theta_{l,k} - \eta \cdot g_{l,k}^q,$$

$$g_{l,k}^q = \kappa_l \cdot g_{l,k},$$

$$\kappa_l = \arg \min_{\kappa_l} ||\Delta \theta_{l,k}^q - T||,$$

$$\Delta \theta_{l,k}^q = \sum \mathbb{I}(\theta_{l,k}^q \neq \theta_{l,k+1}^q),$$

$$T = \rho \cdot \dim(\theta_l),$$

where $\rho \in [0, 1]$ is a predetermined proportion that exceeds the quantization threshold, $\mathbb{I}()$ is the indicator function, and $\dim(\theta_l)$ is the number of elements in $\theta_l$. Our goal is to find $\kappa_l$ that guarantees the number of parameter updates on a quantized layer $\Delta \theta_{l,k}^q$ exceeds a certain ratio $T$.

To quickly find an approximate solution, we applied a simple two-step heuristic. Firstly, starting from 1.0, $\kappa_l$ is doubled until $\Delta \theta_{l,k}^q > T$. Then, to satisfy Eq. 10, $\kappa_l$ is adjusted between $\kappa_l/2$ and $\kappa_l$ by binary search. For the sake of
computation efficiency, the total number of search steps is limited to five. In addition, to assure early phase stability of the training, we added a warm-up phase for the GI method. In the warm-up phase, the maximum of $\kappa$ is limited to 128 for more accurate solutions. When the generator requires a separate warm-up, the GI warm-up phase starts after the generator warm-up ends. Similar to learning rate exponential decay scheduling, we apply the exponential decay to $\rho$. We discuss the sensitivity to this in Sec. 5.4.

5. Experimental Results

5.1. Experimental Environments

We evaluate AIT on three datasets, CIFAR-10, CIFAR-100 [33], and ImageNet (ILSVRC2012 [34]). CIFAR-10 and CIFAR-100 contain 10 and 100 classes of images, respectively, and represent small-scale datasets. ImageNet has 1000 classes of images with 1.2M training samples and 50K validation samples, which represent large-scale dataset.

For CIFAR-10/100, we use commonly used ResNet-20 [21] model. For ImageNet, we use ResNet-18 and ResNet-50 to represent popular medium- and large-sized models, and MobileNetV2 [57] to represent a lightweight model. All pretrained models are from pytorchcv library [1]. For more results on various models, please refer to the Appendix.

For baselines, we use the official code provided by the authors of ZeroQ [5], GDFQ [62], ARC [70] and Qimera [10] with the identical settings. AIT is implemented using PyTorch [52] version 1.10.0. All experiments are conducted using NVIDIA RTX3090 and A6000 GPUs.

The generator is trained with the loss function Eq. (5) with $\alpha = 0.5$ using Adam optimizer [32] with learning rate of 0.001. For training the quantized student model, SGD with Nesterov [47] was used with momentum 0.9. For AIT, the hyperparameter $\rho$ was set to 0.001 and 0.0001 for CIFAR and ImageNet respectively, which both were decayed by 0.1 every 100 epochs. Experiments on CIFAR and ImageNet were run for 400 epochs on learning rate $\eta = 1 \times 10^{-4}$, with batchsize of 200 and 16, respectively.

5.2. Performance Comparison

AIT can be applied to most generative zero-shot quantization methods. In this section, we apply our method to three: GDFQ [62], the first method to suggest such approach, ARC [70], which improves the generator, and Qimera [10], the SOTA technique in the same family. We also include ZeroQ [5] for comparison. We report top-1 accuracies.

Overall, AIT achieves significant performance improvements in most settings tested, whether implemented on top of GDFQ, ARC or Qimera. Notably large improvements have been observed on ImageNet datasets, especially in 4w4a settings, because there still exists a large gap towards the full-precision (32bit) model. The largest gain was found for 4w4a ResNet-50 on top of GDFQ, with the gain of 12.12%p that seems to mainly come from the huge gap (25.61%p) GDFQ originally had between the full-precision model. For the results of lower-bit settings, refer to Appendix.

An interesting trend is that for the other two methods with better generators (Qimera, ARC), the performance gain on 4w4a setting is larger for smaller models (ResNet-50 → ResNet-18 → MobileNetV2). The improvements are (+1.38%p, +2.99%p, +5.19%p) for Qimera and (+3.90%p, +4.41%p, +6.34%p) for ARC in a descending order of model size. This indirectly supports our claim that quantized networks with smaller capacity have difficulties optimizing for multiple loss terms, and AIT can alleviate such effect.

In addition, the performance of ARC+AIT is better than Qimera+AIT for all ImageNet settings except one, even though Qimera outperforms ARC in their default settings. We find the reason from generator model size of ARC. While Qimera uses the exact same generator from GDFQ, ARC uses a larger generator model found by neural architecture search. The result demonstrate that AIT is making better use of the potential of the generator network. Small performance degradations were observed for CIFAR-10 on Qimera by 0.03%. Since CIFAR-10 is a small dataset and the performance is already close to the fp32 model, we believe this is because there is not much room left to improve.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model (FP32 Acc.)</th>
<th>Bits</th>
<th>ZeroQ</th>
<th>GDFQ</th>
<th>GDFQ + AIT</th>
<th>Qimera</th>
<th>Qimera + AIT</th>
<th>ARC</th>
<th>ARC + AIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>ResNet-20 93.89</td>
<td>5w5a</td>
<td>91.34</td>
<td>93.38</td>
<td>93.41</td>
<td>93.46</td>
<td>93.43</td>
<td>92.88</td>
<td>92.89</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>ResNet-20 93.89</td>
<td>5w5a</td>
<td>65.61</td>
<td>66.12</td>
<td>69.26</td>
<td>69.02</td>
<td>69.26</td>
<td>68.40</td>
<td>68.40</td>
</tr>
<tr>
<td>ImageNet</td>
<td>ResNet-18 73.03</td>
<td>5w5a</td>
<td>59.88</td>
<td>68.11</td>
<td>71.70</td>
<td>70.45</td>
<td>71.68</td>
<td>68.40</td>
<td>71.96</td>
</tr>
</tbody>
</table>

Table 1. Comparison on AIT with data-free quantization schemes.
5.3. Ablation Study

Tab. 2 shows an ablation study performed over GDFQ. The ResNet family and MobileNet are denoted as ‘RN’ and ‘MB’ respectively. ‘KL-only’ drops CE from the original loss function of GDFQ, and let the quantized model optimize only on the KL divergence against the full-precision teacher. However, this results in a huge degradation in all settings. As analyzed in Sec. 4.1, this is due to the scarce, unbalanced quantized weight updates. By applying gradient inundation, the lost performance is more than recovered and the superior gain over the baseline is obtained (KL-only+GI).

‘Baseline+GI’ represents the gradient inundation applied on top of GDFQ without dropping the CE term, and ‘CE only+GI’ represents the same with KL term dropped from the loss. Unfortunately, they only result in performance degradation, because the baseline GDFQ with CE+KL loss or CE loss does not suffer from the aforementioned scarce weight update problems. Therefore, gradient inundation only makes detrimental changes to the quantized weights.

Seeing the effect of ‘KL-only + GI’, one might wonder if the weight update problem can be solved by simply increasing the learning rate. ‘KL-only (high lr)’ row shows the results of such experiments conducted with ×100 learning rate. In addition, Fig. 4 shows the distribution of the updates in each layer, in comparison to Fig. 3c. ‘KL-only (high lr)’ achieves a small gain but does not entirely solve the problem. First, increasing the learning rate incurs too frequent updates in a few layer which was already getting enough updates as shown in Figs. 4a and 4b, and many layers still not being updated. Moreover, further increasing the learning rate results in divergence of the model. Figs. 4c and 4d presents the number of updates with gradient inundation, where ρ is depicted in dotted red lines. Gradient inundation tunes the gradients to the right level, leading to a better performance.

For a comprehensive comparison, we also tested adaptive optimizers such as Adam [32], RMSProp [60], and especially LARS [65], which adjusts learning rates per layer. The results of these optimizers for the quantized model are shown in Tab. 3. ‘Baseline-’ denotes the existing methods, and ‘GI-’ denotes modified optimizer with GI. The results show that ‘Baseline-Adam’ and ‘Baseline-RMSProp’ suffer from noticeable accuracy degradation, especially on the large models. ‘Baseline-LARS’ survived from such trend, it does not make significant differences compared to ‘Baseline-SGD’.

We further expand our research by applying GI to Adam and RMSProp, written as ‘GI-Adam’ and ‘GI-RMSProp’. The results verify that our method has solid performance on various optimizers and still outperforms existing methods.

5.4. Sensitivity Analysis

The ρ value controls the portion of quantized weights guaranteed to get updates in each layer. In Tab. 4, a sensitivity study of ρ on top of GDFQ+AIT has been performed. The results show that AIT is not very sensitive to ρ, although there is some effect. Refer to the Appendix for more results.

Tab. 5 shows the learning rate sensitivity of our method with comparison to the GDFQ. The results show that the GI method is robust to the learning rate changes on both datasets while steadily outperforms the baseline method.

5.5. Further Analysis

In this section, we present more details of AIT. Fig. 5a shows the KL divergence over the training in the baseline GDFQ, ‘KL-only’ and AIT. AIT is able to reach a lower KL distance. This supports our analysis from Sec. 4.1 that there is still room for KL to be optimized further.

Another observation can be found from Fig. 5b, where we have measured the angle between the CE and KL. For ‘KL-only’ and AIT, CE is calculated only for measurements, and did not affect the SGD updates. It is now shown that ‘KL-only’ and AIT, both have positive cosine similarities
6. Discussion

Removing Cross-Entropy against the hard label from the loss function as done in AIT could be a penalty, because some methods \cite{10} rely on the sample labels. However, as we demonstrated in Sec. 5, AIT applied to Qimera was able to obtain significantly better performance despite the exclusion of the mixed labels. In addition, our method does not depend on per-image hard label, so it can be widely used for segmentation \cite{41, 56} or object detection \cite{54, 55}.

Privacy Leak is a societal concern of zero-shot quantization because the generator creates synthetic samples that follow the distribution of real data. As several input reconstruction techniques point out \cite{15, 42}, it could be that the synthetic samples can reconstruct the private training data. However, to the extent of our observation, there is no sign that AIT reconstructs the real data as shown in Fig. 6. Training method for the generator in AIT is no different from its baselines, since it does not alter the generator loss in Eq. (5), thus does not contribute any further to the privacy leak.

7. Conclusion

In this work, we analyzed the SOTA family of solutions for zero-shot quantization. Through a series of experiments and analyses, we found that current solutions can be improved through pursuing a flatter minima and guaranteeing weight updates during fine-tuning. We achieve the goal by bringing the quantized model closer to the full-precision model in terms of the KL divergence and designing AIT—before our method, people habitually have used a combination of CE and KL as the main loss of the zero-shot quantization. Experimental results show that AIT is effective and can be easily applied to existing algorithms.

Acknowledgements

This work has been supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2022R1C1C1008131, 2022R1C1C1011307), Year 2022 Copyright Technology R&D Program by Ministry of Culture, Sports and Tourism and Korea Creative Content Agency(Project Name: Development of artificial intelligence-based copyright infringement suspicious element detection and alternative material content recommendation technology for educational content, Project Number: CR202104003, Contribution Rate: 30%), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2020-0-01361, Artificial Intelligence Graduate School Program (Yonsei University)).
References

[1] Computer vision models on PyTorch. 6
[26] Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in SGD. In International Conference of Artificial Neural Networks, 2018. 4


