# Differentiable Soft Min-Max Loss to Restrict Weight Range for Model Quantization

**Arnav Kundu** [*1] **Chungkuk Yoo** [*1] **Minsik Cho** [1] **Saurabh Adya** [1]

## Abstract

The range of weights in a model disrupts effective lower bit quantization. Penalizing the range of weights improve quantization accuracy, but calculation of range (max-min) is not differentiable. In this work, we propose Differentiable Soft Min-Max Loss (DSMM) to restrict weight ranges so that we can get a quantization-friendly model which has narrow weight ranges. We apply DSMM with a learnable parameter which can adjust hardness of DSMM without requiring a special hyper-parameter. DSMM improves lower bit quantization accuracy with state-of-the-art post-training quantization (PTQ), quantization-aware training (QAT), and weight clustering across various domains and model sizes.

## 1. Introduction

Quantization bit-resolution is inversely proportional to the range of weights and affects accuracy of the quantized models. Since outliers tend to increase range, outliers are detrimental for quantization friendly models.

As an example, lets assume we want to quantize the weight distributions shown in Figure 1 (left) into 4 bins. For the original distribution in red most of the weights will be quantized to the central 2 bins and the model accuracy would drop significantly. This problem gets worse for low bit quantization such as 1 or 2 bit quantization. To this end, we introduce Differentiable Soft Min-Max Loss (DSMM), a simple yet powerful method that helps to reduce weight range during training without severely affecting full precision accuracy and provides a quantization friendly checkpoint. Using DSMM loss we intend to trim the edges of the **black** distribution and convert it to the red distribution as shown in Figure 1 (left). Such a restriction might regress the full-precision model's accuracy slightly as the model has to operate under new constraints, but it would have a quantization friendly weight distribution removing outlier weights. Therefore, lower bit quantization accuracy can be improved as shown in Figure 1 (right). In case of higher bit quantization such as 4bit or 8bit, a model might already have enough bits to properly represent wide ranges of weights. Therefore the benefit of DSMM loss could be limited.

Soft max operation has been proposed in previous works [1], but we bring it to the quantization domain for the first time by modifying the function. The key innovation of our work is DSMM loss for reducing a range of weights by operating on outliers more preferably to quantization. To this end, we make the degree of smoothness (temperature $\alpha$ in Equation (1)) in the function as a learnable parameter so that it automatically fits to an optimal value without further hyper-parameter search for the temperature factor. We then add a new term $e^{-\alpha}$ term into the function to encourage the temperature to increase. Without the additional term, the temperature would move towards negative infinite direction to minimize the DSMM always and it disrupts the purpose of DSMM, making quantization-friendly pre-trained model.

We show that DSMM loss works well with state of the art post-training quantization (PTQ), quantization aware training (QAT) and weight clustering algorithms. We also show that our method is applicable to multiple domains like computer vision and natural language processing.

## 2. Related Works

In this paper we have applied our DSMM loss with various training time quantization (quantization-aware training, QAT) algorithms like LSQ [2] and DoReFa [3] used in PACT [4]. PACT clips activation values with a trainable parameter for activation quantization and uses DoReFa for weight quantization. LSQ quantizes weights and activations with learnable step size (scale or bin size).

Also, we have compared our DSMM loss with a state-of-the-art post-training quantization (PTQ) methods. DFQ [5] equalizes per-channel weight ranges by applying per-channel scaling factors. It resolves the wide weight range problem across channels, but still the weight range would

[*]Equal contribution [1]AIML, Apple. Correspondence to: Saurabh Adya <sadya@apple.com>.

*Published at the $2^{nd}$ Differentiable Almost Everything Workshop at the $41^{st}$ International Conference on Machine Learning*, Vienna, Austria. July 2024. 

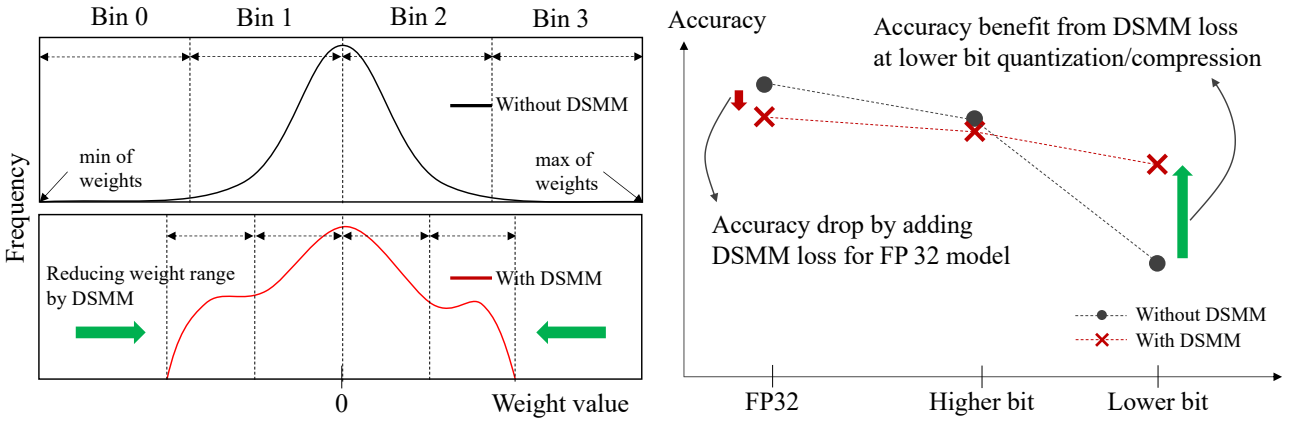


*Figure 1.* (Left) Example weight distribution for quantization using 4 bins. (Right) Benefit of DSMM loss for low bit quantization.

remain wide for lower bit quantization like 4bit as DFQ does not target outliers within a channel. AdaRound [6] proposed adaptive rounding for quantization bin assignment instead of nearest rounding. SQuant [7] decomposes a layer by the Hessian-based optimization objective into sub-items, then composes them in a quantized domain. PD-Quant [8] quantizes weights by comparing model prediction result before and after quantization of each layer. Additionally, we show DSMM loss is effective not only for integer quantization but also for weight clustering. DKM [9] introduces differentiable k-means clustering for weights to represent them in n-bit centroids which have arbitrary float values.

In our extensible experiments, we show our DSMM loss improves accuracies with cutting-edge QAT, PTQ and weight clustering for lower bit quantization like 1bit, 2bit and 4bit.

## 3. Differentiable Soft Min-Max Loss

We introduce Differentiable Soft Min-Max Loss as an auxiliary loss to reduce the range of weights for every layer to get better pre-trained models for further quantization or compression. Just like $L_1$ and $L_2$ regularization our approach is invariant to the quantization or compression technique used. But as opposed to $L_1$ or $L_2$ regularization, Differentiable Soft Min-Max Loss only affects the range of the distribution and not the absolute magnitude of it. In our experiments, we demonstrate that $L_2$ regularization (1x(baseline) and 10x(heavy L2)) only affects the magnitude of the weights but does not remove outliers from the distribution. As evident from Figure 1, heavier L2 regularization just reduces the scale of weights but does affect the shape of the distribution whereas DSMM loss reshapes the distribution. To capture the range we intend to calculate the difference between the maximum and minimum value of the weights in a easily differentiable form as illustrated below. The loss for a given weight W is described in Equation (1).

Here temperature $\alpha$ is a learnable parameter per layer. $e^{-\alpha}$ term in the auxiliary loss $L_{dsmm}$, encourages temperature $\alpha$ to increase during training time optimization process to approach hard-min-max loss towards the end of training. This loss smoothly penalizes not only outliers but also near-outlier weights together. We allow $\alpha$ to be learnable because in our experiments we found that fixing it introduces a new hyper-parameter to tune while worsening the accuracy of the model. A trainable $\alpha$ also allows us to control the smoothness of the loss per layer therefore, introducing more degrees of freedom to the loss.

$$
\begin{aligned}
s_{max} &= \frac{\Sigma(W \odot e^{\alpha \times (W - W_{max})})}{\Sigma e^{\alpha \times (W - W_{max})}} \\
s_{min} &= \frac{\Sigma(W \odot e^{-\alpha \times (W - W_{min})})}{\Sigma e^{-\alpha \times (W - W_{min})}} \\
L_{DSMM} &= (s_{max} - s_{min}) + e^{-\alpha}
\end{aligned}
\tag{1}
$$

DSMM loss was employed during training time of the base model itself and not during quantization. This was done because the purpose of DSMM loss is to provide effective initial weights for quantization. This ensures extensibility of DSMM loss to any quantization technique.

## 4. Experiment

### 4.1. Post-Training Quantization with DSMM loss

We compare models trained using DSMM loss and other weight regularization, L2 and heavy L2, using PTQ methods such as DFQ [5], AdaRound [6], SQuant [7], and PD-Quant [8]. As shown in Table 1, models trained with DSMM loss are more quantization friendly than other regularization. DSMM loss shows the best accuracy as it reduces outliers as well as weight range. On the other hand, heavy L2 regularization makes weight ranges smaller, but it does not remove outliers, therefore prove to be ineffective here.

Compared to FP32 accuracy of baseline models, models trained with DSMM loss, have slight accuracy regression in full-precision inference as we expected in Figure 1 (right). However, after quantizing, the models trained with DSMM loss shows better accuracies.

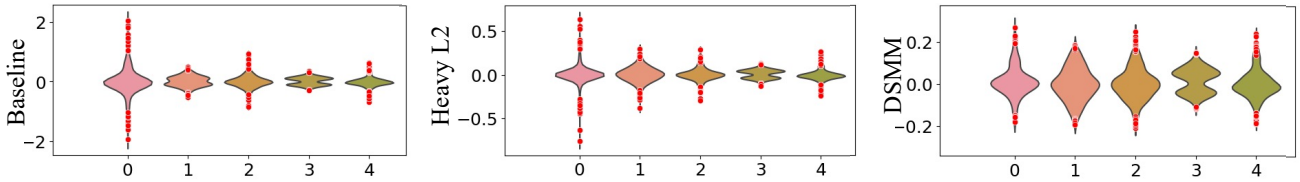


*Figure 2.* Weight distribution of the first five 3x3 convolution layers of MobileNet-V2 using L2 norm (baseline), heavy L2 norm (10x heavy L2 norm than baseline) and the proposed Differentiable Soft Min-Max Loss (the red dots correspond outliers).

*Table 1.* Top-1 accuracies (%) of MobileNet-V1 and V2 on ImageNet-1K using PTQ methods with 4bit weight and 8bit activation quantization. Heavy L2: applied 10x heavy L2 regularization than baseline. Naïve: quantizing without any advanced PTQ techniques. DFQ [5], AR [6], SQ [7], PD-Q [8]. DSMM$^{\alpha}$: fixed $\alpha$ (10.0). DSMM: trainable $\alpha$.

| METHOD | MOBILENET-V1 | | | | | | MOBILENET-V2 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FP32 | NAÏVE | DFQ | AR | SQ | PD-Q | FP32 | NAÏVE | DFQ | AR | SQ | PD-Q |
| BASELINE | 74.12 | 2.67 | 54.06 | 70.42 | 63.85 | 71.87 | 73.08 | 2.57 | **56.56** | 71.29 | 59.30 | 71.54 |
| HEAVY L2 | 72.67 | 13.41 | 57.68 | 69.23 | 66.51 | 69.86 | 71.00 | 4.17 | 0.09 | 68.06 | 57.30 | 68.92 |
| DSMM$^{\alpha=10}$ | 74.21 | 8.52 | 42.48 | 69.71 | 67.23 | 71.92 | 72.80 | 13.77 | 56.19 | 70.79 | 62.16 | 71.27 |
| DSMM(OURS) | 73.95 | **44.24** | **59.21** | **71.35** | **67.86** | **72.39** | 72.81 | **36.69** | 51.46 | **71.77** | **67.16** | **71.98** |

*Table 2.* BLEU score for machine translation task on Transformer Base model [10] with 8bit weight quantization.

| METHOD | FP32 | NAÏVE |
|---|---|---|
| BASELINE | 28.2 | 2.9 |
| DSMM | 27.9 | **27.8** |

*Table 3.* Top-1 accuracies (%) of 2bit weight and 8bit activation PTQ using PD-Quant [8]. MNV1: MobileNet-V1, MNV2: MobileNet-V2, RN50: ResNet-50, RN101: ResNet-101. FP32 accuracy is in Table 1 and Table 4. DSMM$^1$: applied DSMM from scratch. DSMM$^2$: applied DSMM during fine-tuning

| METHOD | MNV1 | MNV2 | RN50 | RN101 |
|---|---|---|---|---|
| BASELINE | 47.62 | 50.66 | 62.92 | 66.50 |
| DSMM$^1$ | **54.20** | **57.53** | **69.31** | **71.24** |
| DSMM$^2$ | **53.10** | **56.31** | **66.85** | **69.66** |

Even without advanced PTQ approaches, models trained with DSMM loss can be reasonably quantized without any further fine-tuning (See Naïve in Table 1). This proves that models with DSMM loss have good weight distribution so they can be quantized with fairly high quantization accuracies. This approach scales to other applications like machine translation as illustrated in Table 2. It is also applicable to larger models like ResNet{50,101} as shown in Table 4.

In Table 3, we can clearly see the benefit of DSMM lossfor lower bit PTQ. The accuracy of ResNet101 trained with DSMM loss only regressed by 8.39% (79.63% $\rightarrow$ 71.24%), while the baseline model trained without DSMM loss shows higher absolute regression, 12.95% (79.45% $\rightarrow$ 66.50%).

### 4.2. PTQ for models fine-tuned with DSMM loss

Training models from scratch might not always be feasible especially given the cost and time taken. Therefore, we finetuned pre-trained models with a low learning rate while applying DSMM as an additional loss to the existing task loss. Our experimental results are shown in Table 3 where we applied DSMM on pre-trained models, followed by PD-Quant (PD-Q) [8] for PTQ. It can be observed that on quantization, finetuning a pre-trained model does not show the same performance gains as training from scratch but it is still significantly better than the baseline. This shows that our method is also applicable to pre-trained models if finetuned with the additional loss.

*Table 4.* Top-1 accuracies (%) of ResNet50 and ResNet101 on ImageNet-1K with 4bit weight and 8bit activation PTQ.

| METHOD | RESNET50 | | | |
|---|---|---|---|---|
| | FP32 | AR | SQ | PD-Q |
| BASELINE | 78.04 | 74.73 | 74.68 | 76.60 |
| DSMM | 78.22 | **75.61** | **75.75** | **77.21** |
| METHOD | RESNET101 | | | |
| | FP32 | AR | SQ | PD-Q |
| BASELINE | 79.45 | 75.18 | 75.23 | 78.16 |
| DSMM | 79.63 | **76.71** | **77.20** | **78.49** |

### 4.3. Quantization-Aware Training with DSMM loss

We apply state-of-the-art quantization techniques like PACT [4] while training the models from scratch using DSMM loss. For LSQ [2] we initialize the model to pre-trained ResNet-18 (RN) [11] with DSMM loss.

As shown in Table 5, DSMM loss helps the quantization techniques in improving their accuracy, especially for extremely low bit quantization such as at 2 bit while it shows similar accuracies with 4 bit. For example, DSMM loss improves 2 bit quantization accuracy with LSQ to over than

*Table 5.* Top-1 accuracies (%) of ResNet18 with various QAT methods. Weights and activations are quantized with the same bit (2W2A: 2bit, 4W4A: 4bit).

| METHOD | 2W2A QAT | | |
|---|---|---|---|
| | FP32 | PACT | LSQ |
| BASELINE | 69.76 | 51.97 | 58.33 |
| DSMM | 69.84 | **55.64** | **62.47** |
| METHOD | 4W4A QAT | | |
| | FP32 | PACT | LSQ |
| BASELINE | 69.76 | 66.90 | **69.90** |
| DSMM | 69.84 | **68.36** | 69.45 |

62% from 58%, but there is no noticeable difference in 4 bit LSQ accuracies with and without DSMM loss. The reason why DSMM loss would not help much for higher bit like 4 bit quantization is that QAT can effectively represent outliers using many bits as we expected in Figure 1 (right).

### 4.4. Weight Clustering with DSMM loss

We evaluate the effectiveness of DSMM loss with the state-of-the-art weight clustering technique, DKM [9], for ResNet-18 and MobileNet-V1. The bit-dim ratio, $\frac{b}{d}$ is an important factor in the DKM algorithm which effectively defines the kind of compression a DKM palettized model would see. We ran these experiments for both scalar and vector palettization. For scalar palettization($dim = 1$) we ran 1 bit, 2 bit and 4 bit compression. Figure 3 shows that DSMM loss significantly improves accuracy from DKM 1 bit and 2 bit models. As we discussed, there is no significant difference for higher bit like 4 bit because many bit compression can also cover outliers even without DSMM.

We also expand the application of DSMM loss to vector palettization(2-bit/2-dim and 4-bit/4-dim) DKM [9] as demonstrated in Figure 3. For these experiments, we kept the effective bit-dim ratio, $\frac{b}{d}$ equivalent to 1 so as to see variation across the models almost compressed to 32x. Since a vector palettized model will require range constraining for all dimensions, we applied multi-dimensional DSMM loss for all layers. For vector palettized ResNet-18 there is an average absolute improvement of $> 1\%$ using models trained with DSMM loss, and for vector palettized MobileNet-V1, the gain ranges from 2% to 5%.

Finally we also validated that DSMM loss for weight clustering scales to other domains as well by applying it in compressing MobileBERT [12]. For Question Answering (QNLI) [13] using MobileBERT, DSMM loss slightly improved the performance of the model as demonstrated in Table 6. Note that we applied DSMM loss to a QNLI fine-tuning task based on a pre-trained MobileBERT [14]. It might be necessary to apply DSMM loss to the entire

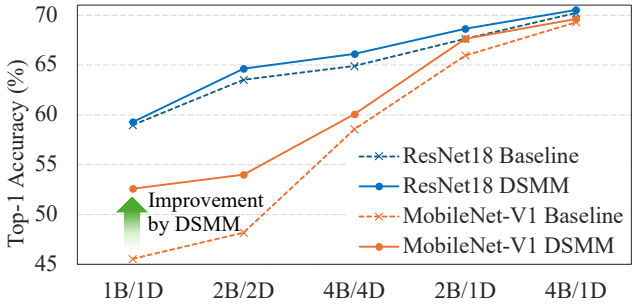


*Figure 3.* Top-1 accuracy for various compression ratios and models using weight clustering. nB/mD: n-bit/m-dim weight clustering.

training task of MobileBERT from scratch so that DSMM loss would have more chances to get effective weight distribution for model compression.

*Table 6.* Question-answering NLI (QNLI) accuracies of MobileBERT using single dimension DKM

| METHOD | PRE-TRAIN | 1-BIT | 2-BIT |
|---|---|---|---|
| DKM BASELINE | 90.41 | 61.34 | 80.12 |
| DKM + DSMM | 90.83 | **61.49** | **80.87** |

## 5. Discussion

As we discussed earlier, DSMM loss would be effective for lower bit quantization, because higher bit quantization can represent outlier weights and wide weight range. Also most of state-of-the-art quantization techniques already achieved reasonable accuracy with higher bit comparing its full-precision models, so there would not be a room for improvement further in higher bit quantization.

We are planning to conduct more studies with DSMM, such as expending to large language model quantization and providing theoretical grounding from current empirical studies.

## 6. Conclusion

In this paper, we introduced Differentiable Soft Min-Max Loss as an effective technique to reduce the weight range for quantization for multiple tasks across domains. This serves as a good initialization for PTQ, QAT and weight clustering methods, and hence can be coupled with any of them. This helps to augment the accuracy gained from such techniques and is invariant to the quantization algorithm. We demonstrated how DSMM loss converts a wide weight range distribution to a more densely-packed distribution for model quantization. While full-precision accuracy with DSMM loss can be slightly regressed as it penalizes outlier weights, it significantly improves quantization accuracy, especially for lower bit.

## References


[1] K. Asadi and M. L. Littman, "An alternative softmax operator for reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, PMLR, Jun. 2017, pp. 243–252. [Online]. Available: https://proceedings.mlr.press/v70/asadi17a.html.

[2] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," in *International Conference on Learning Representations*, 2019.

[3] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv preprint arXiv:1606.06160*, 2016.

[4] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "Pact: Parameterized clipping activation for quantized neural networks," *arXiv preprint arXiv:1805.06085*, 2018.

[5] M. Nagel, M. v. Baalen, T. Blankevoort, and M. Welling, "Data-free quantization through weight equalization and bias correction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1325–1334.

[6] M. Nagel, R. A. Amjad, M. Van Baalen, C. Louizos, and T. Blankevoort, "Up or down? adaptive rounding for post-training quantization," in *International Conference on Machine Learning*, PMLR, 2020, pp. 7197–7206.

[7] C. Guo, Y. Qiu, J. Leng, *et al.*, "SQuant: On-the-fly data-free quantization via diagonal hessian approximation," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=JXhROKNZzOc.

[8] J. Liu, L. Niu, Z. Yuan, D. Yang, X. Wang, and W. Liu, "Pd-quant: Post-training quantization based on prediction difference metric," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 427–24 437.

[9] M. Cho, K. Alizadeh-Vahid, S. Adya, and M. Rastegari, "Dkm: Differentiable k-means clustering layer for neural network compression," in *International Conference on Learning Representations*, 2021.

[10] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[12] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "Mobilebert: A compact task-agnostic bert for resource-limited devices," *arXiv preprint arXiv:2004.02984*, 2020.

[13] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[14] T. Wolf, L. Debut, V. Sanh, *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.

[15] A. G. Howard, M. Zhu, B. Chen, *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[17] 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*, Ieee, 2009, pp. 248–255.

# A. Experiment settings

## A.1. Pre-training from scratch with and without DSMM loss

We train ResNet-18 [11], MobileNet-V1 [15] and MobileNet-V2 [16] on ImageNet 1K [17] with proposed Differentiable Soft Min-Max Loss on a x86 Linux machine with eight GPUs to get pre-trained models before model compression and quantization-aware training. We set initial learning rates to 1.0, 0.4 and 0.4 for ResNet-18, MobileNet-V1 and MobileNet-V2 respectively. We use SGD with 0.9 of momentum with Nesterov. We apply 1e-4 of weight decay (L2 norm weight regularization) for ResNet-18 and 4e-5 for MobileNet-V1 and V2. For heavy L2-regularization, in Figure 2, we use 4e-4 of weight decay (10x heavier than baseline) for MobileNet-V2 to see whether heavy L2-regularization helps quantization or not as a naive solution for range restriction. Strength of DSMM loss is set to 0.01. The learnable parameter $\alpha$ is initially set to 0.1. For comparison, we use pre-trained models of Resnet-18 from Torchvision. As we are using modified version of ResNet-50, and ResNet-101, MobileNet-V1 and V2 for better FP32 performance, we trained those models from scratch without DSMM loss using the same settings above.

## A.2. Quantization and weight clustering

DSMM loss is not a model compression nor quantization method. It penalizes outlier weights during training of the base model from scratch. To evaluate the effectiveness of DSMM loss with model compression and quantization, we apply state-of-the-art compression/quantization techniques, DKM [9], LSQ [2], DFQ [5], AdaRound [6], SQuant [7], and PD-Quant [8] to the pre-trained model with and without DSMM loss. Except SQuant [1], PD-Quant [2], DFQ and AdaRound [3], since other works do not provide official implementation, we implement those techniques ourselves.

We follow the same hyper-parameters used in the works, but we apply compression and quantization for all layers including the first and last layers. It is important to compress/quantize all layers including first and last layers considering computation burden at the first layer with a large convolutional filter size such as 7x7 convolutions in the first layer of ResNet and the large number of weights in the last linear layer, e.g., 1.2M of weights in the last layer of MobileNet-V1 which has 4.2M of weights in total.

[1] https://github.com/clevercool/SQuant
[2] https://github.com/hustvl/PD-Quant
[3] https://quic.github.io/aimet-pages/