A. Optimal range equalization of two layers

Consider two fully-connected layers with weight matrices $\mathbf{W^{(1)}}$ and $\mathbf{W^{(2)}}$, that we scale as in 4.1. We investigate the problem of optimizing the quantization ranges by rescaling the weight matrices by $\mathbf{S} = diag(\mathbf{s})$, where $\mathbf{s} > 0$, such that $\widehat{\mathbf{W}}^{(\mathbf{1})} = \mathbf{S^{-1}W^{(1)}}$ and $\widehat{\mathbf{W}}^{(\mathbf{2})} = \mathbf{W^{(2)}S}$ the weight matrices after rescaling. We investigate the case of symmetric quantization, which also gives good results in practice for asymmetric quantization. We denote

$$\mathbf{r^{(1)}} = 2 \cdot \max_j |\mathbf{S^{-1}} \mathbf{W}^{(1)}_{ij}| = \mathbf{S^{-1}} \hat{\mathbf{r}}^{(1)} \tag{20}$$

$$\mathbf{r^{(2)}} = 2 \cdot \max_i |\mathbf{W}^{(2)}_{ij} \mathbf{S}| = \hat{\mathbf{r}}^{(2)} \mathbf{S} \tag{21}$$

$$R^{(k)} = max_i(\mathbf{r}^{(k)}_i) \tag{22}$$

where $\mathbf{r^{(k)}}$ is the per-channel weight quantization range that is scaled by $\mathbf{S}$, $\mathbf{R^{(k)}}$ the range for the full weight matrix $\mathbf{W^{(k)}}$ and $\hat{\mathbf{r}}^{(k)}$ are the original unscaled ranges.

Using this in our optimization goal of eq. 9 leads to

$$\max_{\mathbf{S}} \sum_i p_i^{(1)} p_i^{(2)} = \max_{\mathbf{S}} \sum_i \frac{r_i^{(1)} r_i^{(2)}}{R^{(1)} R^{(2)}} \tag{23}$$

$$= \max_{\mathbf{S}} \sum_i \frac{\frac{1}{s_i} \hat{r_i}^{(1)} \cdot s_i \hat{r_i}^{(2)}}{\max_j (\frac{1}{s_j} \hat{r}_j^{(1)}) \cdot \max_k (s_k \hat{r}_k^{(2)})} \tag{24}$$

$$= \sum_i \hat{r_i}^{(1)} \hat{r_i}^{(2)} \max_{\mathbf{S}} \frac{1}{\max_j (\frac{1}{s_j} \hat{r}_j^{(1)}) \cdot \max_k (s_k \hat{r}_k^{(2)})}. \tag{25}$$

We observe that the specific scaling s_i of each channel cancels out as long as they do not increase R, the range of the full weight matrix. We can reformulate the above to

$$\min_{\mathbf{S}} \left(\max_j \left(\frac{1}{s_j} \hat{r}_j^{(1)} \right) \cdot \max_k (s_k \hat{r}_k^{(2)}) \right) \tag{26}$$

which is minimal iff

$$\arg\max_j \frac{1}{s_j} r_j^{(1)} = \arg\max_k s_k r_k^{(2)}. \tag{27}$$

By contradiction, if $j \neq k$ there is a small positive ϵ such that $s'_k = s_k - \epsilon$ which will decrease $\max_k s_k \hat{r}_k^{(2)}$ by $\epsilon \hat{r}_k^{(2)}$ without effecting $\max_j \frac{1}{s_j} \hat{r}_j^{(1)}$. Therefore such a solution would not be optimal for eq. 26.

The condition from eq. 27 implies there is a limiting channel $i = \arg\max_i r_i^{(1)} r_i^{(2)}$ which defining the quantization range of both weight matrices $\widehat{\mathbf{W}}^{(\mathbf{1})}$ and $\widehat{\mathbf{W}}^{(\mathbf{2})}$. However, our optimization goal is not effected by the choice of the other s_j given the resulting $r_j^{(1)}$ and $r_j^{(2)}$ are smaller than $r_i^{(1)}$ and $r_i^{(2)}$, respectively. To break the ties of solutions we decide to set $\forall i : r_i^{(1)} = r_i^{(2)}$. Thus the channel's ranges between both tensors are matched as closely as possible and the introduced quantization error is spread equally among both weight tensors. This results in our final rescaling factor

$$s_i = \frac{1}{r_i^{(1)}} \sqrt{r_i^{(1)} r_i^{(2)}} \tag{28}$$

which satisfies our necessary condition from eq. 27 and ensures that $\forall i : r_i^{(1)} = r_i^{(2)}$.

B. Bias correction for convolutional layers

Similarly to fully connected layers we can compute $\boldsymbol{\epsilon}$ from $\mathbf{W}$ and $\widetilde{\mathbf{W}}$, it becomes a constant and we have that $\mathbb{E}[\boldsymbol{\epsilon} * \mathbf{x}] = \boldsymbol{\epsilon} * \mathbb{E}[\mathbf{x}]$. Expanding this yields:

$$[\boldsymbol{\epsilon} * \mathbb{E}[\mathbf{x}]]_{c_o ij} = \sum_{c_i mn} \mathbb{E}[\mathbf{x}_{c_i, i-m, j-n}] \boldsymbol{\epsilon}_{c_o c_i mn} \tag{29}$$

$$= \sum_{c_i} \left[\mathbb{E}[\mathbf{x}_{c_i}] \sum_{mn} \boldsymbol{\epsilon}_{c_o c_i mn} \right] \tag{30}$$

where we assume that the expected value of each input channel is the same for all spatial dimensions in the input channel. Since the value of $[\boldsymbol{\epsilon} * \mathbb{E}[\mathbf{x}]]_{c_o ij}$ does not depend on the spatial dimensions i and j, the expected error is the same for the full output channel and can be folded into the layer's bias parameter.

C. Clipped normal distribution

Given a normally distributed random variable X with mean μ and variance σ^2, and a clipped-linear function $f(\cdot)$ that clips its argument to the range $[a, b]$, s.t. $a < b$, the mean and variance of $f(X)$ can be determined using the standard rules of computing the mean and variance of a function:

$$\mu_{ab}^c = \int_{-\infty}^{\infty} f(x) p(x) dx \tag{31}$$

$$\sigma_{ab}^{c\,2} = \int_{-\infty}^{\infty} (f(x) - \mu_{ab}^c)^2 p(x) dx \tag{32}$$

where we define $p(x) = \mathcal{N}(x \mid \mu, \sigma)$, $\mu_{ab}^c = \mathbb{E}[f(X)]$ and $\sigma_{ab}^{c\,2} = Var[f(X)]$.

C.1. Mean of Clipped Normal Distribution

Using the fact that $f(x)$ is constant if $x \notin [a, b]$ we have that:

$$\mu_{ab}^c = \int_{-\infty}^{\infty} f(x) p(x) dx \tag{33}$$

$$= a \int_{-\infty}^{a} p(x) dx + \int_a^b x p(x) dx + b \int_b^{\infty} p(x) dx \tag{34}$$

The first and last term can be computed as $a\Phi(\alpha)$ and $b(1-\Phi(\beta))$ respectively, where we define $\alpha = \frac{a-\mu}{\sigma}$, $\beta = \frac{b-\mu}{\sigma}$, and $\Phi(x) = CDF(x \mid 0, 1)$, the normal CDF with zero mean and unit variance.

The integral over the linear part of $f(\cdot)$ can be computed as:

$$\int_a^b xp(x)dx = C\int_a^b xe^{-\frac{1}{2\sigma^2}(x-\mu)^2}dx \tag{35}$$

$$= -C\sigma^2 e^{-\frac{1}{2\sigma^2}(x-\mu)^2}\Big|_a^b + \mu(\Phi(\beta)-\Phi(\alpha)) \tag{36}$$

$$= \sigma\left(\phi(\alpha)-\phi(\beta)\right) + \mu(\Phi(\beta)-\Phi(\alpha)) \tag{37}$$

where we define $\phi(\cdot) = \mathcal{N}(\cdot \mid 0, 1)$, i.e. the standard normal pdf and $C = \frac{1}{\sigma\sqrt{2\pi}}$ is the normalization constant for a normal distribution with variance σ^2, thus

$$\begin{aligned}\mu_{ab}^c =& \sigma\left(\phi(\alpha)-\phi(\beta)\right) + \mu(\Phi(\beta)-\Phi(\alpha)) \\ &+ a\Phi(\alpha) + b(1-\Phi(\beta)).\end{aligned} \tag{38}$$

C.2. Variance of Clipped Normal Distribution

We again exploit the fact that $f(x)$ is constant if $x \notin [a, b]$:

$${\sigma_{ab}^c}^2 = \int_{-\infty}^{\infty}(f(x)-\mu_{ab}^c)^2 p(x)dx \tag{39}$$

$$\begin{aligned}&= \int_{-\infty}^{a}(a-\mu_{ab}^c)^2 p(x)dx + \\ &+ \int_a^b (x-\mu_{ab}^c)^2 p(x)dx + \\ &+ \int_b^{\infty}(b-\mu_{ab}^c)^2 p(x)dx\end{aligned} \tag{40}$$

The first and last term can be solved as $(a-\mu_{ab}^c)^2\Phi(\alpha)$ and $(b-\mu_{ab}^c)^2(1-\Phi(\beta))$ respectively.

The second term can be decomposed as follows:

$$\int_a^b (x-\mu_{ab}^c)^2 p(x)dx = \int_a^b (x^2 - 2x\mu_{ab}^c + {\mu_{ab}^c}^2)p(x)dx \tag{41}$$

$$\begin{aligned}&= \int_a^b x^2 p(x)dx \\ &+ Z({\mu_{ab}^c}^2 - 2\mu_{ab}^c\mu_{ab}^t)\end{aligned} \tag{42}$$

where we use the result from the previous subsection and define $Z = \Phi(\beta)-\Phi(\alpha)$, and where $\mu_{ab}^t = \frac{1}{Z}\int_a^b x\mathcal{N}(x \mid \mu, \sigma^2) = \mu + \sigma(\phi(\alpha)-\phi(\beta))/Z$ is the mean of the truncated normal distribution.

Model	CLE+BA	Clip@15
No BiasCorr	70.92%	2.55%
Analytic BiasCorr	71.19%	70.43%
Empirical BiasCorr	71.15%	69.85%

Table 6. Top1 ImageNet validation results for MobileNetV2 with weights and activations quantized to INT8. Comparing analytic and empirical bias correction combined with cross-layer equalization (CLE), bias absorption (BA) and clipping.

Evaluating the first term yields:

$$\begin{aligned}\int_a^b x^2 p(x)dx =& Z(\mu^2+\sigma^2) \\ &+ \sigma(a\phi(\alpha) - b\phi(\beta)) \\ &+ \sigma\mu(\phi(\alpha)-\phi(\beta))\end{aligned} \tag{43}$$

This results in:

$$\begin{aligned}Var[f(X)] =& Z(\mu^2+\sigma^2+{\mu_{ab}^c}^2 - 2\mu_{ab}^c\mu) \\ &+ \sigma(a\phi(\alpha) - b\phi(\beta)) \\ &+ \sigma(\mu - 2\mu_{ab}^c)(\phi(\alpha)-\phi(\beta)) \\ &+ (a-\mu_{ab}^c)^2\Phi(\alpha) \\ &+ (b-\mu_{ab}^c)^2(1-\Phi(\beta))\end{aligned} \tag{44}$$

D. Empirical quantization bias correction

If a network does not use batch normalization, or does not use batch normalization in all layers, a representative dataset can be used to compute the difference between pre-activation means before and after quantization. We then subtract this difference from the quantized model's pre-activations. This procedure can be run with unlabeled data. The procedure should be run after BatchNorm folding and cross-layer range equalization. Clipping should be applied in the quantized network, but not in the floating point network. Since the activation function and the quantization operation are fused, this procedure is run on a network with quantized weights only. However, after this procedure is applied activations can be quantized as well. We bias correct a layer only after all the layers feeding into it have been bias-corrected. The procedure is as follows:

1. Run N examples through the FP32 model and collect for each layer the per-channel pre-activation means $\mathbb{E}[\mathbf{y}]$.
2. For each layer L in the quantized model:
   - Collect the per-channel pre-activation means $\mathbb{E}[\tilde{\mathbf{y}}]$ of layer L for the same N examples as in step 1.
   - Compute the per-channel biased quantization error $\mathbb{E}[\boldsymbol{\epsilon}] = \mathbb{E}[\tilde{\mathbf{y}}] - \mathbb{E}[\mathbf{y}]$.

Model	Symmetric	Asymmetric
MobileNet V1	70.32%	70.51%
MobileNet V2	71.15%	71.19%
Resnet18	69.50%	69.62%

Table 7. Top1 ImageNet validation results for MobileNetV2 after applying DFQ. Weights and activations quantized using symmetric and asymmetric 8-bit integer quantization.

- Subtract $\mathbb{E}[\epsilon]$ from the layer's bias parameter.

In Table 6 we compare this empirical bias correction procedure with the analytic bias correction introduced in section 4.2. We observe that both approaches leads to similar results.

E. Additional experiments

Combination with fine-tuning The focus of our method is data-free quantization (level 1). However, our method can also be used as a pre-processing before quantization aware fine-tuning. To demonstrate this we used DFQ together with short-term quantization aware fine-tuning [18]. After just 1 epoch of quantization aware fine-tuning MobileNet V2, accuracy increases from 71.19% to 71.42%, almost recovering the FP32 performance (71.72%).

Symmetric vs asymmetric quantization In our experimental section all our experiments were performed with asymmetric quantization, since this is commonly used in literature. Here we also compare to symmetric quantization. Symmetric quantization does not use an offset, which eliminates several cross terms in the calculations done on hardware compared to asymmetric quantization. This makes symmetric quantization more efficient on some hardware at the expense of losing some expressive power.

In Table 7 we compare symmetric and asymmetric quantization in combination with DFQ. For all three models the advantage of asymmetric quantization is almost negligible. We noticed that cross-layer equalization is effective at removing outliers, resulting in weight distributions are often close to symmetric.

DFQ combined with per-channel quantization In our experiments we focused on per-tensor quantization since the more recent per-channel quantization is not efficiently supported on all hardware. For hardware that does support it, we analyze the effect of DFQ in combination with per-channel quantization.

In Table 8 we show the results of the different components of DFQ in combination with per-channel quantization. We notice that each individual component, cross-layer equalization, bias absorption and bias correction, incremental improve over per-channel quantization and reduce the total quantization error from 1.07% to only 0.39%.

Model	No BiasCorr	BiasCorr
Original model	70.65%	70.80%
CLE	70.93%	71.30%
CLE+BA	71.03%	**71.33%**

Table 8. Top1 ImageNet validation results for MobileNetV2 using 8-bit per-channel quantization. Activations are quantized per tensor. Showing the effect of cross-layer equalization (CLE) and bias absorption (BA) in combination with bias correction.