Mixed-Precision Neural Network Quantization via Learned Layer-wise Importance

Chen Tang*, Kai Ouyang*, Zhi Wang, Yifei Zhu, Yaowei Wang, Wen Ji, Wenwu Zhu

Overview

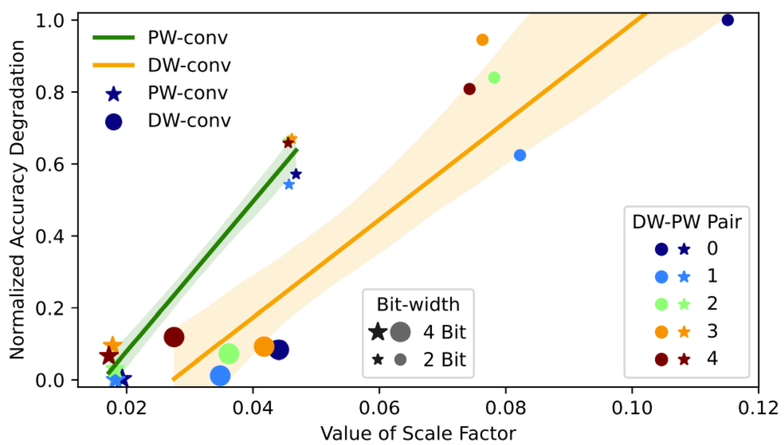
The exponentially large discrete search space in mixed-precision quantization makes it hard to determine the optimal bit-width for each layer.
Previous works usually resort to ***iterative search*** methods on the training set, which consume hundreds or even thousands of GPU-hours.

In our work, we find the ***scale-factor in the quantizer*** can be used to reflect the quantization-sensitivity of different layers.

Quantization function (quantizer) **Learnable scale-factor**

$$v^q = Q_b(v; s) = round(clip(\frac{v}{s}, min_b, max_b)) \times s,$$

We conduct a comparative experiment for MobileNetv1 to observe the numerical difference of the scale-factors.

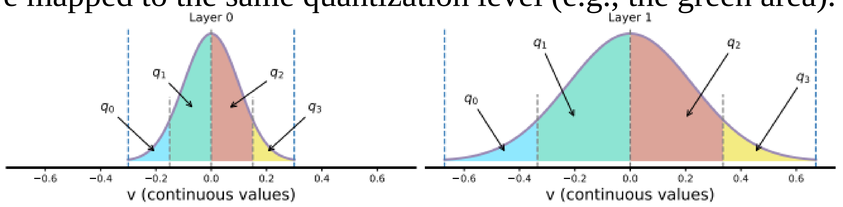


One can see that the scale factor values of DW-conv layer (more sensitive) are significantly higher than the PW-conv layer (less sensitive).

Motivation

Motivation - the scale factors are optimized to adjust the quantization mapping during training. After converging, they naturally capture the quantization information of layers.

We consider two example layers with **well-trained** ***s*** and weights under 2bits quantization. The learned scale-factor ***s*** controls the quantization intervals and their corresponding quantization levels (see the areas with different colors).
One can see we should give the layer on the right (with larger scale-factor value) more bit-width, since more continuous values are mapped to the same quantization level (e.g., the green area).



ILP based policy search

We design an integer linear programming-based one-time method to allocate bit-width automatically.

$$\arg\min_{\{x_{i,j}^{(l)}\}_{l=0}^{L}} \sum_{l=0}^{L} (s_{a,j}^{(l)} + \alpha \times s_{w,i}^{(l)}) \times x_{i,j}^{(l)}$$

$$\text{s.t.} \quad \sum_i \sum_j x_{i,j}^{(l)} = 1$$

$$\sum_l \sum_i \sum_j BitOps\left(l, x_{i,j}^{(l)}\right) \le C$$

$$\text{vars} \quad x_{i,j}^{(l)} \in \{0, 1\}$$

Results

Efficiency results

Based on our importance indicators, the MPQ policy search time for ResNet18 and ResNet50 is 0.06s and 0.35s, respectively.

Performance results

Table 4. Results for MobileNetv1 on ImageNet with BitOps constraints. "W-b" and "A-b" means weight and activation bit-widths. "Top-1" and "Top-5" represent top-1 and top-5 accuracy of quantized model respectively. "B (G)" means BitOps (G).

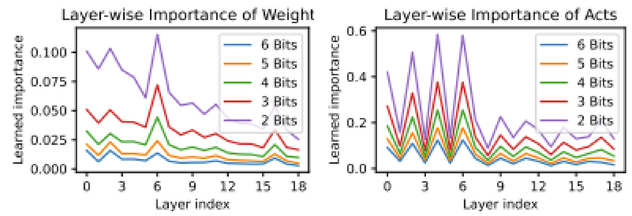
Method	W-b	A-b	Top-1	Top-5	B (G)
PROFIT	4	4	69.05	88.41	9.68
PACT	6	4	67.51	87.84	14.13
HMQ	3MP	4MP	69.30	-	-
HAQ	4MP	4MP	67.45	87.85	-
HAQ	6MP	4MP	70.40	89.69	-
Ours	3MP	3MP	**69.48**	**89.11**	**5.78**
Ours	4MP	4MP	**71.84**	**90.38**	**9.68**

Table 5. Weight only quantization results for MobileNetv1 on ImageNet. "W-b" means weight bit-widths. "S (M)" means quantized model size (MB).

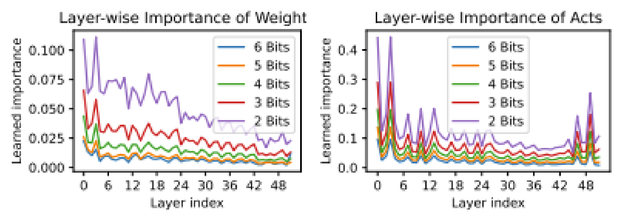
Method	W-b	Top-1	Top-5	S (M)
DeepComp	3MP	65.93	86.85	1.60
HAQ	3MP	67.66	88.21	1.58
HMQ	3MP	69.88	-	**1.51**
Ours	3MP	**71.57**	**90.30**	1.79
PACT	8	70.82	89.85	4.01
DeepComp	4MP	71.14	89.84	2.10
HAQ	4MP	71.74	90.36	**2.07**
HMQ	4MP	70.91	-	2.12
Ours	4MP	**72.60**	**90.83**	2.08

Visualization

The importance indicators for ResNet18 and ResNet50.



(a) ResNet18



(b) ResNet50