MetaAug: Meta-Data Augmentation for Post-Training Quantization

Cuong Pham[1], Hoang-Anh Dung[1], Cuong C. Nguyen[2], Trung Le[1], Dinh Phung[1,3], Gustavo Carneiro[2], Thanh-Toan Do[1]

[1] Department of Data Science and AI, Monash University, Australia
[2] Centre for Vision, Speech and Signal Processing, University of Surrey, UK
[3] VinAI, Vietnam

EUROPEAN CONFERENCE ON COMPUTER VISION
MILANO 2024

Motivation

Post-Training Quantization (PTQ)

- PTQ is an effective approach for quantizing pre-trained models using a small calibration dataset.
- Most previous works (e.g., QDrop, PD-Quant, and Genie) rely only on the original calibration data for training and lack a validation set to validate the quantized models. This could make the quantized models prone to overfitting.

Contributions

- A novel meta-learning-based approach to mitigate overfitting in PTQ
  - ✓ A transformation network and a quantized model are jointly optimized through bi-level optimization.
  - ✓ The outputs of the transformation network are used to train the quantized model, while the original data is used to validate it.
- We investigate various losses for training the transformation network to preserve information from the original data, and introduce a margin loss to prevent it from becoming an identity mapping.

Proposed Method

Meta-learning formulation for PTQ

Bi-level optimization

$$T^* = \arg\min_T \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{val}(\hat{\theta}_Q, x_i)$$
$$s.t.: \hat{\theta}_Q = \arg\min_{\theta_Q} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_Q(\theta_Q, T(x_i))$$

Regarding $\mathcal{L}_Q$

$$\mathcal{L}_Q(\theta_Q, T(S) = \frac{1}{N}\sum_{i=1}^{N} \left\| A_{FP}^l(T(x_i) - A_Q^l(T(x_i)) \right\|^2$$

Regarding $\mathcal{L}_{val}$

$$\mathcal{L}_{val}(\hat{\theta}_Q, S) = \frac{1}{N}\sum_{i=1}^{N} KL\left[\sigma(f_{\theta_{FP}}(x_i) \parallel \sigma(f_{\hat{\theta}_Q}(x_i)\right]$$

Proposed Method

Properties of transformation network *T*

- Encouraging generated images $S^g = \{T(x_i)\}_{i=1}^{N}$ retain information of the original images $S = \{x_i\}_{i=1}^{N}$.
- Prevent the transformation network from becoming an identity mapping.

Update transformation network *T*

Information preservation

$$\mathcal{L}_{DP}(T, S) = \frac{1}{N}\sum_{i=1}^{N} KL\left[\mathcal{P}_i \parallel \mathcal{P}_i^{(g)}\right]$$

$$\mathcal{P}_{i|j} = \frac{K(f_{\theta_{FP}}(x_i), f_{\theta_{FP}}(x_j))}{\sum_{k=1, k\neq j} K(f_{\theta_{FP}}(x_k), f_{\theta_{FP}}(x_j))}$$

Identity prevention

$$\mathcal{L}_{margin}(T, S) = \frac{1}{N}\sum_{i=1}^{N} \max\left(0, \epsilon - \frac{1}{M}\|x_i - T(x_i)\|^2\right)$$

Overall loss for training *T*

$$\mathcal{L}_T(T, S) = \alpha\mathcal{L}_{val}(\hat{\theta}_Q, S) + \beta\mathcal{L}_{margin}(T, S) + \gamma\mathcal{L}_{DP}(T, S)$$

Algorithms

Algorithm 1 Data modification for post-training quantization.

procedure TRAIN(θ_{FP},S)
  ▷ θ_{FP}: weight of the full-precision model. ◁
  ▷ L: Number of blocks in the full-precision model. ◁
  ▷ S: Calibration data. ◁
  ▷ N_T: Number of iterations to update T. ◁
  ▷ N_Q: Number of iterations to quantize model. ◁
  ▷ T: Transformation network to modify calibration dataset S. ◁
  Initialize the quantized model θ_Q from θ_{FP} using LAPQ
  Warm up the transformation network T.
  for $l = 1$ to L **do**
    for $t = 1$ to N_T **do**
      Sample a mini-batch: $\mathbb{B} = \{x_i : x_i \sim \mathcal{S}\}$
      Modify $\mathbb{B}$ with the transformation network T to get $T(\mathbb{B}) = \{T(x_i)\}_{i=1}^{|\mathbb{B}|}$
      ▷ Forward pass and update the quantized model using modified data. ◁
      Compute: $\mathcal{L}_Q(\theta_Q, T(\mathbb{B})) = \frac{1}{|\mathbb{B}|}\sum_{i=1}^{|\mathbb{B}|} ||A_{FP}^l(T(x_i)) - A_Q^l(T(x_i))||^2$
      Update $\widehat{\theta}_Q$: $\widehat{\theta}_Q \leftarrow \text{Adam}(\mathcal{L}_Q(\theta_Q, T(\mathbb{B})))$
      ▷ Validate $\widehat{\theta}_Q$ on the original calibration data. ◁
      Sample a mini-batch data: $\mathbb{B}^v = \{x_i^v : x_i^v \sim \mathcal{S}\}$
      Compute: $\mathcal{L}_T(T, \mathbb{B}^v)$
      Update T: $T \leftarrow \text{Adam}(\mathcal{L}_T(T, \mathbb{B}^v))$
    ▷ Quantize l^{th} block of θ_Q using the original calibration data $\mathcal{S}$ and modified data with the learned T. ◁
    for $t = 1$ to N_Q **do**
      Sample a mini-batch: $\mathbb{B}_q = \{x_{qi} : x_{qi} \sim \mathcal{S}_q = T(\mathcal{S}) \cup \mathcal{S}\}$
      Compute: $\mathcal{L}_Q(\theta_Q, \mathbb{B}_q) = \frac{1}{|\mathbb{B}_q|}\sum_{i=1}^{|\mathbb{B}_q|} ||A_{FP}^l(x_{qi})) - A_Q^l(x_{qi})||^2$
      Update: $\theta_Q \leftarrow \text{Adam}(\mathcal{L}_Q(\theta_Q, \mathbb{B}_q))$
  return quantized model θ_Q and T.

Experimental Results

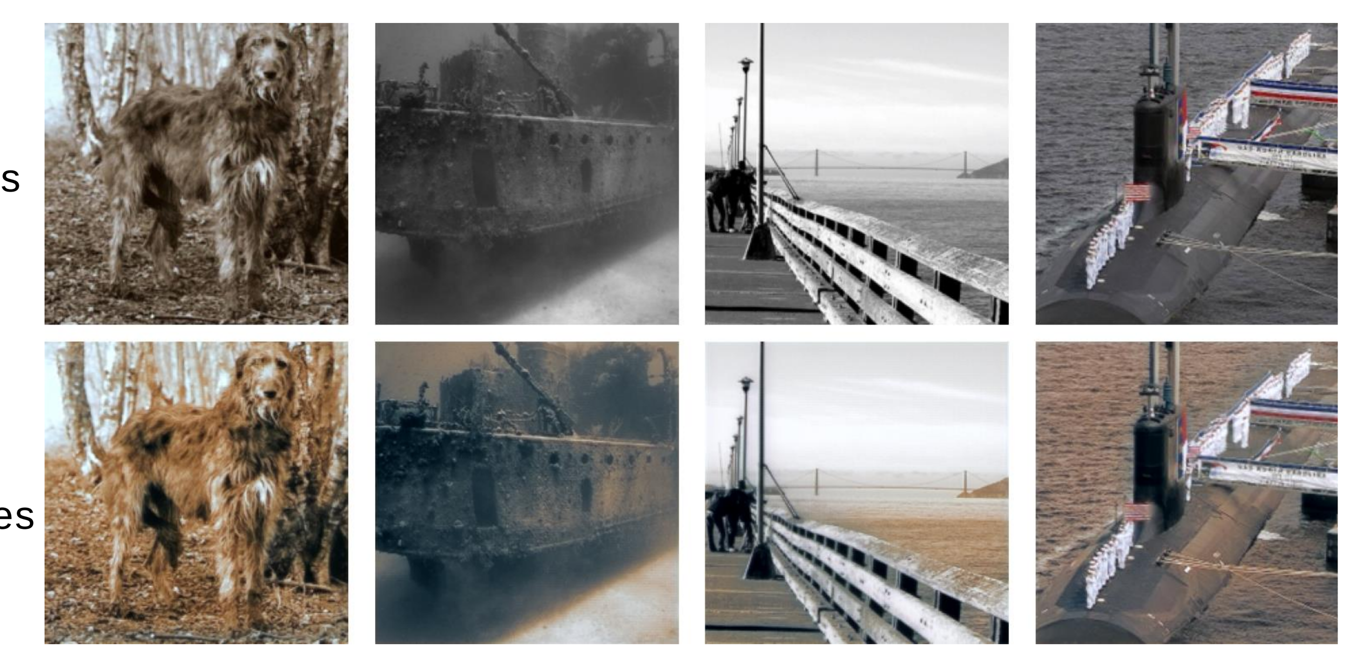
Method	Bit-width (W/A)	Accuracy on the test set	Accuracy on the calibration set	Train-test accuracy gap
FP	32/32	71.01	85.16	14.15
QDrop [46]	2/2	51.14	77.53	26.39
PD-Quant [24]		53.14	83.30	30.16
Genie-M [16]		53.77	80.18	27.01
MetaAug (Ours)		54.22	77.64	**23.42**
QDrop [46]	2/4	64.66	81.64	16.98
PD-Quant [24]		65.17	84.38	19.21
Genie-M [16]		65.77	84.18	18.41
MetaAug (Ours)		66.01	82.91	**16.90**

Method	Bit-width (W/A)	ResNet-18	ResNet-50	MobileNetV2
FP	32/32	71.01	76.63	72.20
AdaRound [28]	4/4	67.96	73.88	61.52
BRECQ* [21]		69.60	75.05	66.57
QDrop [46]		69.10	75.03	67.89
QDrop* [46]		69.62	75.45	68.84
PD-Quant [24]		69.23	75.16	68.19
Genie-M [16]		69.35	75.21	68.65
Bit-Shrinking* [23]		69.94	**76.04**	69.02
MetaAug (Ours)		69.48	75.29	68.76
MetaAug* (Ours)		**69.97**	75.78	**69.22**
AdaRound [28]	3/3	64.14	68.40	41.52
BRECQ* [21]		64.80	70.29	53.34
QDrop [46]		65.56	71.07	54.27
QDrop* [46]		66.75	72.38	57.98
Genie [16]		66.16	71.61	57.54
Bit-Shrinking* [23]		67.12	72.91	58.66
MetaAug (Ours)		66.37	71.73	57.77
MetaAug* (Ours)		**67.66**	**73.04**	**59.87**
BRECQ* [21]	2/4	64.80	70.29	53.34
QDrop [46]		64.66	70.08	52.92
QDrop* [46]		65.25	70.65	54.22
PD-Quant [24]		65.17	70.77	55.17
Genie-M [16]		65.77	70.51	56.38
Bit-Shrinking* [23]		65.77	71.11	54.88
MetaAug (Ours)		66.01	70.76	56.45
MetaAug* (Ours)		**66.48**	**71.48**	**56.65**
BRECQ* [21]	2/2	42.54	29.01	0.24
QDrop [46]		51.14	54.74	8.46
QDrop* [46]		54.72	58.67	13.05
PD-Quant [24]		53.14	57.16	13.76
Genie-M [16]		53.71	56.71	$16.25^{\ddagger}$
Bit-Shrinking* [23]		57.33	59.03	18.23
MetaAug (Ours)		54.22	57.30	16.97
MetaAug* (Ours)		**57.89**	**60.50**	**19.61**

Visualization

Original images

Modified images



References

Wei, Xiuying, et al. "QDrop: Randomly dropping quantization for extremely low-bit post-training quantization." ICLR 2022.
Liu, Jiawei, et al. "PD-Quant: Post-training quantization based on prediction difference metric." CVPR. 2023.
Jeon, Yongkweon, Chungman Lee, and Ho-young Kim. "Genie: show me the data for quantization." CVPR 2023.