Reclaiming Residual Knowledge: A Novel Paradigm to Low-Bit Quantization

Róisín Luo (Jiaolin Luo)[1,2]
j.luo2@universityofgalway.ie
Alexandru Drimbarean[3]

James McDermott[1,2]

Colm O'Riordan[1,2]

[1] SFI Centre for Research Training in Artificial Intelligence
Dublin, D02 FX65, Ireland

[2] University of Galway
Galway, H91 TK33, Ireland

[3] Tobii Corporation
Galway, H91 V0TX, Ireland

Abstract

This paper explores a novel paradigm in low-bit (*i.e.* 4-bits or lower) quantization, differing from existing state-of-the-art methods, by framing optimal quantization as an architecture search problem within convolutional neural networks (ConvNets). Our framework, dubbed **CoRa** (Optimal Quantization Residual **Co**nvolutional Operator Low-**Ra**nk Adaptation), is motivated by two key aspects. Firstly, quantization residual knowledge, *i.e.* the lost information between floating-point weights and quantized weights, has long been neglected by the research community. Reclaiming the critical residual knowledge, with an infinitesimal extra parameter cost, can reverse performance degradation without training. Secondly, state-of-the-art quantization frameworks search for optimal quantized weights to address the performance degradation. Yet, the vast search spaces in weight optimization pose a challenge for the efficient optimization in large models. For example, state-of-the-art BRECQ necessitates 2×10^4 iterations to quantize models. Fundamentally differing from existing methods, **CoRa** searches for the optimal architectures of low-rank adapters, reclaiming critical quantization residual knowledge, within the search spaces smaller compared to the weight spaces, by many orders of magnitude. The low-rank adapters approximate the quantization residual weights, discarded in previous methods. We evaluate our approach over multiple pre-trained ConvNets on ImageNet. **CoRa** achieves comparable performance against both state-of-the-art quantization-aware training and post-training quantization baselines, in 4-bit and 3-bit quantization, by using less than 250 iterations on a small calibration set with 1600 images. Thus, **CoRa** establishes a new state-of-the-art in terms of the optimization efficiency in low-bit quantization. Implementation can be found on https://github.com/aoibhinncrtai/cora_torch.

1 Introduction

ConvNets [1, 23] are favored as vision foundation models, offering distinct advantages such as the inductive bias in modeling visual patterns [6, 16, 35], efficient training, and hardware-

© 2024. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

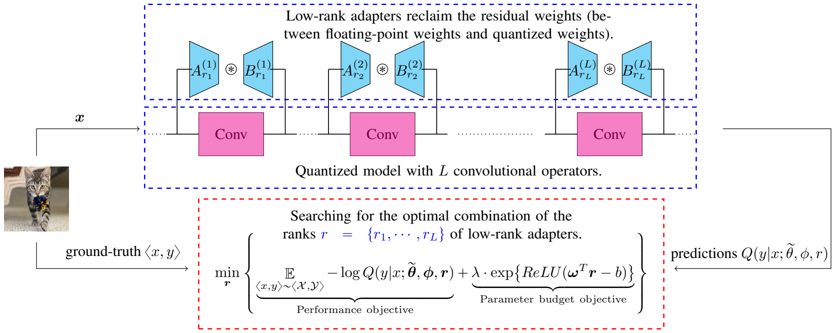

Low-rank adapters reclaim the residual weights (between floating-point weights and quantized weights).
$A^{(1)}_{r_1}$ ⊛ $B^{(1)}_{r_1}$
$A^{(2)}_{r_2}$ ⊛ $B^{(2)}_{r_2}$
$A^{(L)}_{r_L}$ ⊛ $B^{(L)}_{r_L}$
x
Conv
Conv
Conv
Quantized model with L convolutional operators.
Searching for the optimal combination of the ranks $r = \{r_1, \cdots, r_L\}$ of low-rank adapters.
ground-truth $\langle x, y\rangle$
predictions $Q(y|x;\widetilde{\theta},\phi,r)$
$\min_{\boldsymbol{r}} \left\{ \underbrace{\mathbb{E}_{\langle x,y\rangle\sim\langle\mathcal{X},\mathcal{Y}\rangle} - \log Q(y|x;\widetilde{\boldsymbol{\theta}},\boldsymbol{\phi},\boldsymbol{r})}_{\text{Performance objective}} + \underbrace{\lambda\cdot\exp\{ReLU(\boldsymbol{\omega}^T\boldsymbol{r}-b)\}}_{\text{Parameter budget objective}} \right\}$

Figure 1: **CoRa** framework: Searching for the optimal adapters, reclaiming the quantization residual knowledge, instead for the optimal quantized weights. The low-rank convolutional adapter at the l-th layer $B^{(l)}_{r_l} \circledast A^{(l)}_{r_l}$ is determined by a discrete integer r_l.

friendliness [27]. Network quantization is indispensable in enabling efficient inference when deploying large models on devices with limited resources [14, 25, 29, 33]. Representing floating-point tensors as integers significantly reduces the computational requirements and memory footprint.

Yet, low-bit quantization often leads to severe performance degradation [5, 11, 24]. For example, the standard accuracy of a *resnet18*, pretrained on ImageNet [7], plummets to a mere 1.91% from 67.32%, with 4-bit weight-only quantization (WOQ), using *min-max* clipping [31]. To tackle this issue, two research lines are undertaken: quantization-aware training (QAT) and post-training quantization (PTQ) [11, 13, 30].

QAT methods seek the optimal quantized weights during the training process to minimize performance degradation. Despite their promising performance, the substantial computational and data requirements pose major challenges in deployment efficiency. For instance, the state-of-the-art PACT [4] entails a minimum of $10^{8.12}$ iterations with $1.2M$ training samples to converge on ImageNet. Additionally, empirical evidence shows that QAT methods often yield very limited performance at low-bit quantization due to optimization difficulty [10, 26, 32]. PTQ methods, *e.g*. AdaRound [30] and BRECQ [22], overcome these limitations by reconstructing the optimal quantized weights of pre-trained models, with optimization on small calibration sets; these potentially reduce computational and data requirements.

Notably, both state-of-the-art QAT and PTQ methods quantize models by optimizing within weight spaces. Their optimization efficiencies are substantially hindered by the vast dimensions of search spaces. For example, a *resnet50* contains over 2.5M trainable parameters [15], suggesting a search space of the dimension of $\mathbb{R}^{25,000,000}$. The state-of-the-art PTQ method BRECQ needs at least 2×10^4 iterations to converge for a *resnet50* pre-trained on ImageNet.

This research delves into a question: "**Beyond the quantization methods with weight space optimization, does an alternative paradigm exist**?". Intuitively, the quantization residual knowledge – namely, the *quantization residual weights* between floating-point weights and quantized weights – retains vital information lost during the quantization process. This quantization residual knowledge, which has long been overlooked by the research community, holds the potential value that reverses performance degradation without training. Motivated by this perspective, our approach, **CoRa**, as shown in Figure 1, explores a novel paradigm, differing from state-of-the-art QAT and PTQ methods: **by seeking the optimal low-rank adapters [17], reclaiming the residual knowledge; thus reversing the perfor-**

mance degradation, and establishing a new state-of-the-art in terms of the optimization efficiency.

A low-rank adapter consists of two cascaded convolutional filters (*e.g*. A and B) with significantly lower sizes, which are directly converted from high-rank quantization residual weights. As shown in Figure 1, the l-th layer adapter $B_{r_l}^{(l)} \circledast A_{r_l}^{(l)}$, with a low rank r_l, is attached to the l-th layer convolutional filter, and approximates the quantization residual weights. **CoRa** seeks the optimal ranks $r = \{r_1, \cdots, r_L\}$ for all adapters. Surprisingly, earlier works [8, 17, 19, 34, 37, 38, 39] do not address the problem of converting the existing weights of convolutional operators into the weights of the adapters without training. To tackle this problem, we prove a result, as stated in Residual Convolutional Representation Theorem 1.

The search space of the low-rank adapters in a model is significantly smaller by many orders of magnitude compared to the space of weights. For instance, a *resnet50* has 53 convolutional filters. In this case, the structure of the low-rank adapters is only controlled by 53 parameters (*i.e*. 53 ranks). This suggests that the search space is of dimension $\mathbb{R}^{53}$, smaller by 6 orders of magnitude than the weight space. Thanks to the smaller search space, **CoRa** converges within less than 250 iterations for pre-trained models on ImageNet, yet achieves comparable performance against both state-of-the-art QAT and PTQ baselines.

This research is in the scope of low-bit WOQ and ConvNets. Our contributions are summarized as:

1 **CoRa method**. We present an efficient, low-bit, and PTQ framework for ConvNets, by framing optimal quantization as an architecture search problem, to re-capture quantization residual knowledge with low-rank adapters;

2 **Neural combinatorial optimization**. We introduce a differentiable neural combinatorial optimization approach, searching for the optimal low-rank adapters, by using a smooth high-order normalized Butterworth kernel;

3 **Training-free residual operator conversion**. We show a result, converting the weights of existing high-rank quantization residual convolutional operators to low-rank adapters without training, as stated in Theorem 1.

2 Preliminaries

Dataset and classifier. Let $\langle \mathcal{X}, \mathcal{Y} \rangle$ be an image dataset where $\mathcal{X}$ denotes images and $\mathcal{Y}$ denotes labels. We use $Q(y|x;\theta)$ to represent a classifier, where θ denotes parameters. Q predicts the probability of a discrete class y given image x.

Quantization. We use $[\![W]\!]_n$ to denote the n-bit quantization of tensor W. The *clipping range* refers to the value range in quantization [11]. We use two clipping schemes: (1) *min-max clipping* chooses the minimum and maximum values. (2) *normal clipping* chooses $[\mu - k \cdot \sigma, \mu + k \cdot \sigma]$, where μ denotes the mean of the tensor, σ denotes the standard deviation of the tensor and k determines the range. Details are in Appendix A.

Kolda mode-*n* matricization and tensorization. Let $Z \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ be a N-order tensor [20]. The Kolda mode-n *matricization* of Z [20], denoted as $Z_{(n)}$, refers to *unfolding* Z so that: the n-th dimension becomes the first dimension, and the remaining dimensions are squeezed as the second dimension by $\Pi_{i \neq n} I_i$. Let $Y \in \mathbb{R}^{I_n \times J}$ ($J = \Pi_{i \neq n} I_i$) be a matrix. The mode-n *tensorization* of Y, denoted as $Y_{[n, I_1 \times I_{n-1} \times I_{n+1} \times \cdots \times I_N]}$, refers to *folding* Y into the shape

$I_1 \times \cdots \times I_N$. Readers can further refer to the literature [20, 21, 40]. Details are also provided in Appendix B.

Residual convolutional operator. Let $W \in \mathbb{R}^{m \times n \times k_1 \times k_2}$ be the weights of a convolution operator, where m denotes output channels, n denotes input channels and $k_1 \times k_2$ denotes filter kernel size. We refer to W as *convolutional operator* for brevity. We use $W \circledast x$ to denote the convolution operation. Convolutional operators are linear operators. We refer to $\Delta[\![W]\!]_n := W - [\![W]\!]_n$ as *quantization residual operator*, or *residual operator* if without ambiguity.

Theorem 1 (Residual Convolutional Representation)**.** *Suppose a singular value decomposition given by:* $(\Delta[\![W]\!]_n)_{(1)} = US_rV^T$ *(*$r = \mathrm{rank}(S_r)$*). Then the factorization holds true:*

$$W \circledast x = [\![W]\!]_n \circledast x + \underbrace{B \circledast A}_{\text{residual operator}} \circledast x \tag{1}$$

where $A = (S_r^{\frac{1}{2}} V^T)_{[1, r \times 1 \times 1]}$ *and* $B = (US_r^{\frac{1}{2}})_{[1, n \times k_1 \times k_2]}$. *The* $B \circledast A$ *is referred as r-rank residual operator. The proof is provided in Appendix C.*

3 Method

We frame the optimal quantization as an architecture search problem. Suppose a L-layer floating-point ConvNet Q:

$$Q(y|x;\theta) := Q(y|x;W^{(1)}, \cdots, W^{(L)}) \tag{2}$$

in which $W^{(l)}$ denotes the parameters of the l-th layer and $\theta := \{W^{(1)}, \cdots, W^{(L)}\}$. The quantized Q with bit-width n is:

$$Q(y|x;\widetilde{\theta}) := Q(y|x;[\![W^{(1)}]\!]_n, \cdots, [\![W^{(L)}]\!]_n) \tag{3}$$

where $\widetilde{\theta} := \{[\![W^{(1)}]\!]_n, \cdots, [\![W^{(L)}]\!]_n\}$.

Approximating residual knowledge. According to Theorem 1, in the l-th layer, the residual operator $\Delta[\![W^{(l)}]\!]_n$ is approximated by a r_l-rank residual operator:

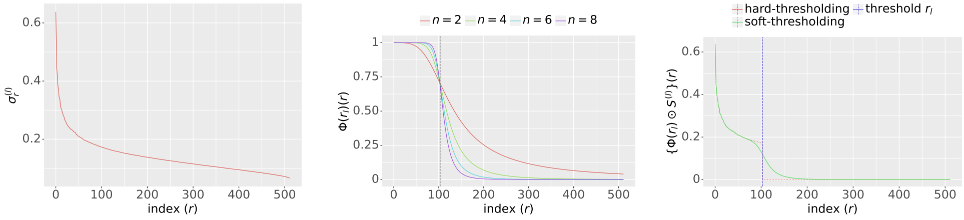
$$W^{(l)} \circledast x - [\![W^{(l)}]\!]_n \circledast x = \Delta[\![W^{(l)}]\!]_n \circledast x \approx B_{r_l}^{(l)} \circledast A_{r_l}^{(l)} \circledast x. \tag{4}$$

Notably, the $A_{r_l}^{(l)}$ and $B_{r_l}^{(l)}$ are directly converted from $\Delta[\![W^{(l)}]\!]_n$ without training, which is guaranteed by Theorem 1. By approximating the residual operators via Equation (4), the quantized model is written as:

$$Q(y|x;\widetilde{\theta}, \phi, r) := Q(y|x;[\![W^{(1)}]\!]_n + B_{r_1}^{(1)} \circledast A_{r_1}^{(1)}, \cdots, [\![W^{(L)}]\!]_n + B_{r_L}^{(L)} \circledast A_{r_L}^{(L)}) \tag{5}$$

where $\phi = \{B_{r_1}^{(1)} \circledast A_{r_1}^{(1)}, \cdots, B_{r_L}^{(L)} \circledast A_{r_L}^{(L)}\}$ are the low-rank residual operators, and $r = \{r_1, \cdots, r_L\}$ ($0 \leq r_l \leq R_l$, $r_l \in \mathbb{N}$) are the parameters controlling the ranks of these operators. The implementation is as shown in Figure 1.

Discrete combinatorial optimization. Suppose $r = \{r_1, \cdots, r_L\}$ are a set with L discrete ranks, controlling the structure of the low-rank adapters in Figure 1. Suppose R_l is the l-th layer maximum rank of r_l. Formally, the optimization objective is to seek a set of



(a) singular values $S^{(l)}$ (b) $\Phi(r_l)$ w/ various orders (c) thresholding by $\Phi(r_l) \odot S^{(l)}$

Figure 2: Differentiable thresholding with a high-order normalized Butterworth kernel $\Phi(r_l)$. The $S^{(l)}$ is in the 14-th layer of a pre-trained *resnet18* on *ImageNet*. The cut-off rank r_l is 103.

optimal discrete r, by maximizing the performance on a calibration set $\langle \mathcal{X}, \mathcal{Y} \rangle$, subject to an adaptation parameter budget constraint:

$$\begin{aligned} r^* = \arg\min_{r} \quad & \left\{ \mathop{\mathbb{E}}_{\langle x,y \rangle \sim \langle \mathcal{X}, \mathcal{Y} \rangle} - \log Q(y|x; \widetilde{\theta}, \phi, r) \right\} \\ \text{subject to} \quad & \omega^T r \leq b, 0 \leq r_i \leq R_i, \; r_i \in \mathbb{N}, \; 0 \leq b \leq 1 \end{aligned} \tag{6}$$

where r^* denotes the optimal ranks, b denotes normalized maximum adaptation parameter budget (*i.e.* target budget), and $\omega := \{\omega_1, \cdots, \omega_L\}$ denotes the *rank normalization coefficients* used to compute the normalized parameter size. The ω_l is given by: $\omega_l := \frac{1}{R_l} \cdot \frac{\Theta_l}{\sum_{i=1}^{L} \Theta_i}$ where Θ_i is the i-th layer parameter size, as proved in Appendix D. The optimization search space size is less than dimension of $\mathbb{R}^L$. The only learnable parameters are r.

Adapter parameter budget constraint. We limit the amount of the parameters of the adapters by:

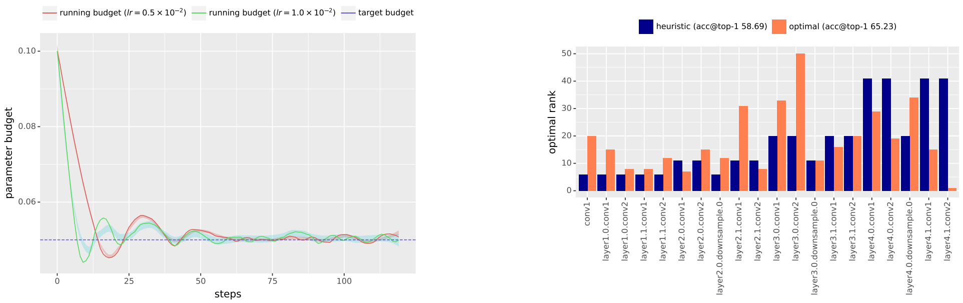
$$\min_{r} \left\{ \lambda \cdot \exp\left\{ ReLU(\omega^T r - b) \right\} \right\} \tag{7}$$

where $\lambda \in \mathbb{R}$ is a penalty coefficient. The motivation of using $\exp(\cdot)$ is to obtain non-linear gradients favoring gradient-based optimization, by assigning smaller gradients to smaller ranks, instead of the constant gradients ω. The $ReLU(\cdot)$ is used to stop gradients if the running budget $\omega^T r$ is already below the target budget b.

3.1 Differentiable relaxation

Equation (6) is not differentiable with respect to r. Solving the discrete combinatorial optimization problem in Equation (6) often entails iterative algorithms, *e.g.* evolutionary algorithms (EA) and integer programming (IP) [2, 28, 36]. Nevertheless, the huge discrete search spaces remain a significant hurdle. For instance, the number of possible combinations of low-rank adapter sizes in a *resnet50* is above 10^{18}.

To enable efficient optimization, firstly, we relax the r in Equation (6) from discrete integers to continuous values. Secondly, we differentiably parameterize the operations of choosing r, by using a high-order normalized Butterworth kernel. With these endeavors, Equation (6) is differentiable with respect to r. We are able to use standard gradient descent algorithms to efficiently optimize (*e.g.* SGD and Adam).



(a) running budget ($\omega^T r$)

(b) solution

Figure 3: Optimization iterations and solution. The experiments are with *resnet18* pre-trained on ImageNet.

3.2 Parameterized differentiable thresholding

Hard thresholding. Suppose $S^{(l)}$ is the singular value matrix of $[\![\Delta W^{(l)}]\!]_n$. Suppose $S^{(l)}_{r_l}$ chooses the r_l largest values of $S^{(l)}$:

$$S^{(l)} = diag(\sigma_1^{(1)}, \cdots, \sigma_{R_l}^{(l)}) \quad S^{(l)}_{r_l} = diag(\sigma_1^{(1)}, \cdots, \sigma_{r_l}^{(l)}, 0, \cdots, 0) \quad 0 \leq r_l \leq R_l. \tag{8}$$

Formally, choosing the r_l can be formulated as the Hadamard product (*i.e.* element-wise product) of a thresholding mask matrix $\Phi^*(r_l)$ and $S^{(l)}$ in that:

$$S^{(l)}_{r_l} = \Phi^*(r_l) \odot S^{(l)} \qquad \Phi^*(r_l) = diag(\underbrace{1, \cdots, 1}_{r_l \text{ ones}}, \underbrace{0, \cdots, 0}_{R_l - r_l \text{ zeros}}). \tag{9}$$

We refer to Equation (9) as *hard-thresholding*, which is not differentiable with respect to r.

Soft thresholding. We differentiably approximate the *hard-thresholding* with a high-order normalized Butterworth kernel (NBK) [3]. An k-order NBK with a cut-off rank r_l is a vector map $\Phi(r_l) : r_l \mapsto [0,1]^{R_l}$ defined by:

$$\Phi(r_l) := \left(\frac{1}{\sqrt{1 + (\frac{r}{r_l})^{2k}}} \right)_{r=1}^{R_l} \qquad r_l \ll R_l \tag{10}$$

where n is the order. Figure 2(a) shows an example of $S^{(l)}$. Figure 2(b) shows an example of NBK. Figure 2(c) shows the results of the differentiable thresholding with NBK.

Converting residual operator to low-rank operator. In the l-th layer, we differentiably convert a high-rank residual operator $\Delta[\![W^{(l)}]\!]_n$ into a low-rank operator $B^{(l)}_{r_l} \circledast A^{(l)}_{r_l}$ with rank r_l, by using Equation (10):

$$\Delta[\![W^{(l)}]\!]_n \approx B^{(l)}_{r_l} \circledast A^{(l)}_{r_l} \tag{11}$$

$$\approx (U^{(l)}[\Phi(r_l) \odot S^{(l)\frac{1}{2}}])_{[1, n \times k_1 \times k_2]} \circledast ([\Phi(r_l) \odot S^{(l)\frac{1}{2}}]V^{(l)^T})_{[1, r_l \times 1 \times 1]}. \tag{12}$$

which is differentiable with respect to r_l.

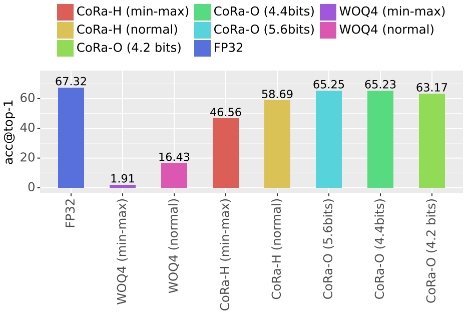


Figure 4: Ablation study on *ImageNet* with *resnet18*. **CoRa-H**: Heuristic ranks. **CoRa-O**: Optimal ranks.

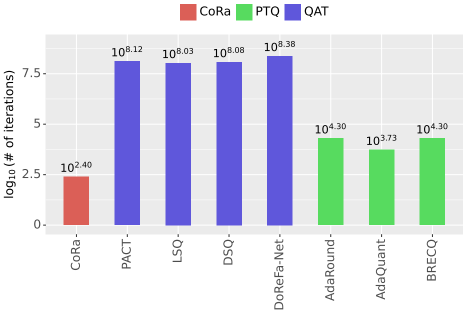


Figure 5: Optimization efficiency on *ImageNet*. Results are in a logarithmic scale.

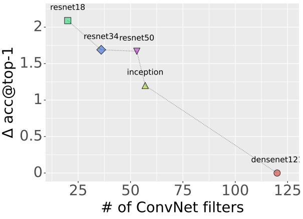


Figure 6: Performance scalability with respect to ConvNet sizes.

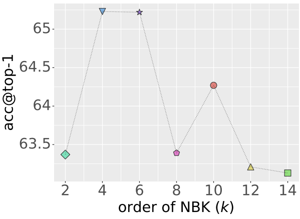


Figure 7: Sensitivity of the order *k*.

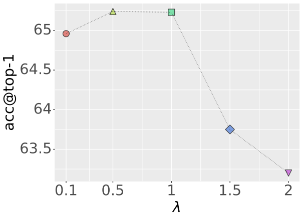


Figure 8: Sensitivity of λ.

3.3 Neural combinatorial optimization

By combining Equation (6) and Equation (7), the optimization loss is:

$$\mathcal{L}(r) := \mathop{\mathbb{E}}_{\langle x,y\rangle\sim\langle\mathcal{X},\mathcal{Y}\rangle} \Big\{ -\log Q(y|x;\widetilde{\theta},\phi,r) + \lambda\cdot\exp\{ReLU(\omega^T r - b)\} \Big\}. \tag{13}$$

The optimal $r = \{r_1,\cdots,r_L\}$ are found using gradient descent optimizers, *e.g.* SGD and Adam.

Heuristic choice of ranks. This method serves as a baseline. The r_l is heuristically chosen as $r_l = \lfloor b\cdot R_l \rfloor$, proportionally assigning the l-th layer rank according to the budget b. For example, suppose the maximum rank at the l-th layer is 512 and budget b is set to 5%, the heuristic r_l is chosen as $\lfloor 512\times 0.05 \rfloor = 25$.

Intriguing observation. Figure 3(a) shows the running budget $\omega^T r$ during the optimization. Figure 3(b) shows an example of the solution on a *resnet18* on ImageNet. Our analysis in terms of the solutions from a variety of ConvNets suggests that: **The heuristic choices often overstate the importance of middle to last layers; conversely, optimal solutions underscore the importance of beginning to middle layers**. The full solutions are provided in Appendix F.

Table 1: Top-1 accuracy comparison of low-bit quantization. We use marker × to indicate that the results are not available.

Model	bits	FP32	CoRa	QAT Baselines				PTQ Baselines		
				PACT	LSQ	DSQ	DoReFa-Net	AdaRound	AdaQuant	BRECQ
resnet18	4	67.32	**65.23**	66.08	67.89	66.99	65.99	65.08	65.18	66.96
resnet50		74.52	**72.85**	×	73.84	73.39	×	72.81	72.80	73.88
resnet18	3		**64.50**	65.31	66.13	×	×	64.47	55.04	66.12
resnet50			**72.81**	×	×	×	×	71.06	65.43	70.87
# of iterations		$10^{8.38}$	$\leq 10^{2.40}$	$10^{8.12}$	$10^{8.03}$	$10^{8.08}$	$10^{8.38}$	$10^{4.30}$	$10^{3.73}$	$10^{4.30}$
training size		$1.2M$	1600	$1.2M$	$1.2M$	$1.2M$	$1.2M$	2048	1000	1024

3.4 Tricks for stable optimization

Stable optimization for the proposed neural combinatorial optimization in Section 3.3 is challenging. We adopt several tricks to numerically stabilize the optimization process.

- **Gradient clipping**. To stabilize the optimization, the solver clips the gradients into the range of $[-0.2, 0.2]$.
- **Adaptive gradient**. Equation (7) gives non-linear gradients with respect to ranks. Smaller ranks have smaller gradients towards zero, while larger ranks have larger gradients. We believe this design favors the stabilization of optimization.
- **Solution clamping**. The solver clamps the ranges of the solutions after every gradient update, guaranteeing that the rank is not less than 1 and not greater than the limit R_l.
- **Anomaly reassignment**. The solver detects numerical anomalies. If a *NaN* rank value is detected, it is replaced with rank 1.

4 Experiments

We conduct experiments from five aspects: (1) ablation study (Section 4.1), (2) comparing with state-of-the-art QAT and PTQ baselines (Section 4.2), (3) extensive evaluation (Section 4.3), (4) performance scalability with respect to model sizes (Section 4.4) and (5) hyper-parameter sensitivity (Section 4.5).

Reproducibility. We sample 1600 images from the *ImageNet* validation set as our calibration set while using the remainder as our validation set. We use *normal clipping* with $k = 4.0$ to quantize the main network and *min-max clipping* to quantize adapters. The order n of NBK is set to 4.0. The penalty coefficient λ is set to 1. The batch size is 32. The target budget b is set to 5%, which results in a 1.25% increase in memory footprint with 8-bit quantization for low-rank adapters. The optimizer is Adam without weight decay. The learning rate is set to 0.01. We use a maximum 250 iterations for all experiments.

Testbed. All experimental results, including the measured results of floating-point reference accuracy, are conducted on the M2 chip of a MacBook Air, equipped with a GPU of size 24 GiB. Due to the choice of the validation set, in tandem with the random seed and the hardware acceleration implementation in the testbed, the results of reference accuracy are slightly lower compared to the results from pytorch. However, this does not affect the results, we obtain using pytorch, for a fair comparisons with baselines. We report the results that we measured on our own testbed rather than using the results from the literature.

Equivalent quantization bit-width. Let n and m be the quantization bit-widths of the main network and adapters. The equivalent quantization bit-width is given as: $n + m \cdot b$. For

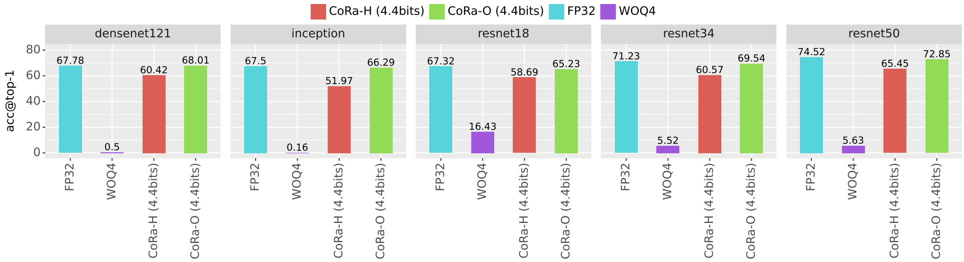


Figure 9: Top-1 accuracy of multiple vision architectures on *ImageNet* with 4-bit quantization.

example, suppose $n = 4$, $m = 8$ and $b = 5\%$, the equivalent quantization bit-width is 4.4-bits. The proof is provided in Appendix E.

4.1 Ablation study

We conduct ablation experiments to show the design considerations in: (1) *normal* clipping is better than *min-max* clipping, (2) the results with optimal ranks outperform the results with heuristic choices, and (3) quantizing residual adapters with 8-bits does not affect performance. The results are shown in Figure 4.

Intriguingly, we can quantize adapters while the performance remains almost unchanged. This can significantly reduce the amount of extra parameters which are used to retain residual knowledge.

4.2 Comparing with baselines

We compare our method against both state-of-the-art QAT and PTQ baselines. We choose four QAT baselines: PACT [4], LSQ [10], DSQ [12], and DoReFa-Net [41]. We choose three PTQ baselines: AdaRound [30], AdaQuant [18], and BRECQ [22]. We quantize the *resnet18* and *resnet50* (pre-trained on *ImageNet*) with 4-bits and 3-bits quantization.

Top-1 accuracy. Our results achieve comparable performance against the baselines. The results are shown in Table 1. **Optimization efficiency**. Our method is more efficient by many orders of magnitude than state-of-the-art baselines. The results are shown in Figure 5 and Table 1. Notably, our method uses only 250 iterations with very minimum extra parameter cost. We have established a new state-of-the-art in terms of optimization efficiency.

4.3 Extensive evaluation

We show the results of extensive performance evaluation over multiple image classifiers pre-trained on ImageNet in Figure 9. Our results achieve comparable performance against the floating-point reference models with the differences within 2.5%. The full solutions are provided in Appendix F.

4.4 Performance scalability

Figure 6 shows the performance scalability of **CoRa** with respect to model sizes, which are assessed by the numbers of filters. The result shows that the top-1 accuracy difference to floating-point models decreases with respect to the number of filters in models.

4.5 Hyper-parameter sensitivity

There are two hyper-parameters: the order n of the NBK in Equation (10) and the λ in the loss function. We empirically investigate how their choices affect performance. The experiments are on a *resnet18* pre-trained on ImageNet. Figure 7 shows that the NBK order $k = 4$ achieves best performance. Figure 8 shows that λ achieves best performance between 0.5 and 1. It is notable that k and λ are model-dependent.

5 Related work

Low-rank convolutional operator approximation. Low-rank approximation of convolutional operators is promising in accelerating the computations [23]. However, convolution operations are not matrix multiplications. Conventional low-rank approximation, *e.g.* LoRa [17] and QLoRa [9], fails to approximate convolutional operators. Relatively few works in the literature have explored this problem. Denton et al. decompose filters into the outer product of three rank-1 filters by optimization [8]. Rigamonti et al. use rank-1 filters to approximate convolutional filters by learning [34]. Jaderberg et al. reconstruct low-rank filters with optimization, by exploiting the significant redundancy across multiple channels and filters [19]. A recent work, Conv-LoRA, approximates filters with the composed convolutions of two filters for low-rank fine-tuning on ConvNets [39]. However, Conv-LoRA does not solve the problem of converting existing operators to low-rank operators without training. Previous works need to reconstruct low-rank filters by learning, thus they do not satisfy our needs. **CoRa** uses Theorem 1 to convert existing residual operators into low-rank operators without training.

6 Future work

CoRa introduces a novel paradigm in low-bit quantization and demonstrates significant optimization efficiency, with a new state-of-the-art result, as shown by our experiments compared to baselines. This paper exclusively investigates this paradigm on ConvNets. Future research will aim to explore this paradigm further from three aspects: (1) enhancing the performance of existing quantization methods (*e.g.* QAT and PQT) by reclaiming the residual knowledge using **CoRa**; (2) extending this paradigm to architectures beyond ConvNets, such as transformers; and (3) broadening the scope to more diverse tasks, including large vision models (LVMs) and large language models (LLMs).

7 Conclusions

We explore a novel paradigm, in optimal low-bit quantization, differing from existing state-of-the-art methods, by re-framing the problem as an architecture search problem, of optimally reclaiming quantization residual knowledge. Thanks to significantly smaller search spaces of adapters, our method is more efficient yet achieves comparable performance against state-of-the-art baselines. **CoRa** has established a new state-of-the-art in terms of the optimization efficiency.

Acknowledgements

This publication has emanated the grant support from research conducted with the financial support of Science Foundation Ireland under grant number 18/CRT/6223. Financial support has also been provided by XPeri Corporation (Galway) and Tobii Corporation (Galway). For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. We thank researchers Fatemeh Amerehi and Laura O'Mahony, from the University of Limerick (Ireland), for their proofreading. Specially, we extend our gratitude to the reviewers for their constructive comments, which have significantly enhanced the quality of our research.

References

[1] Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundational models defining a new era in vision: A survey and outlook. *arXiv preprint arXiv:2307.13721*, 2023.

[2] Thomas Bartz-Beielstein, Jürgen Branke, Jörn Mehnen, and Olaf Mersmann. Evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(3):178–195, 2014.

[3] Stephen Butterworth et al. On the theory of filter amplifiers. *Wireless Engineer*, 7(6): 536–541, 1930.

[4] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.

[5] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3009–3018. IEEE, 2019.

[6] Stéphane d'Ascoli, Levent Sagun, Giulio Biroli, and Joan Bruna. Finding the needle in the haystack with convolutions: on the benefits of architectural bias. *Advances in Neural Information Processing Systems*, 32, 2019.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in neural information processing systems*, 27, 2014.

[9] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

[10] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

[11] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC, 2022.

[12] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4852–4861, 2019.

[13] Yunhui Guo. A survey on methods and theories of quantized neural networks. *arXiv preprint arXiv:1808.04752*, 2018.

[14] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Katherine Hermann, Ting Chen, and Simon Kornblith. The origins and prevalence of texture bias in convolutional neural networks. *Advances in Neural Information Processing Systems*, 33:19000–19015, 2020.

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[18] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Improving post training neural quantization: Layer-wise calibration and integer programming. *arXiv preprint arXiv:2006.10518*, 2020.

[19] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.

[20] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[21] James Li, Jacob Bien, and Martin T Wells. rtensor: An r package for multidimensional array (tensor) unfolding, multiplication, and decomposition. *Journal of Statistical Software*, 87:1–31, 2018.

[22] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021.

[23] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 2021.

[24] Zhuo Li, Hengyi Li, and Lin Meng. Model compression for deep neural networks: A survey. *Computers*, 12(3):60, 2023.

[25] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461: 370–403, 2021.

[26] Zhi-Gang Liu and Matthew Mattina. Learning low-precision neural networks without straight-through estimator (ste). *arXiv preprint arXiv:1903.01061*, 2019.

[27] José Maurício, Inês Domingues, and Jorge Bernardino. Comparing vision transformers and convolutional neural networks for image classification: A literature review. *Applied Sciences*, 13(9):5521, 2023.

[28] Nina Mazyavkina, Sergey Sviridov, Sergei Ivanov, and Evgeny Burnaev. Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research*, 134:105400, 2021.

[29] Rahul Mishra, Hari Prabhat Gupta, and Tanima Dutta. A survey on deep neural network compression: Challenges, overview, and solutions. *arXiv preprint arXiv:2010.03954*, 2020.

[30] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206. PMLR, 2020.

[31] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

[32] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *International Conference on Machine Learning*, pages 16318–16330. PMLR, 2022.

[33] James O' Neill. An overview of neural network compression. *arXiv preprint arXiv:2006.03669*, 2020.

[34] Roberto Rigamonti, Amos Sironi, Vincent Lepetit, and Pascal Fua. Learning separable filters. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2754–2761, 2013.

[35] Zihao Wang and Lei Wu. Theoretical analysis of the inductive biases in deep convolutional networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[36] Laurence A Wolsey. *Integer programming*. John Wiley & Sons, 2020.

[37] Huanrui Yang, Minxue Tang, Wei Wen, Feng Yan, Daniel Hu, Ang Li, Hai Li, and Yiran Chen. Learning low-rank deep neural networks via singular vector orthogonality regularization and singular value sparsification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 678–679, 2020.

[38] Xiangyu Zhang, Jianhua Zou, Kaiming He, and Jian Sun. Accelerating very deep convolutional networks for classification and detection. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):1943–1955, 2015.

[39] Zihan Zhong, Zhiqiang Tang, Tong He, Haoyang Fang, and Chun Yuan. Convolution meets lora: Parameter efficient finetuning for segment anything model. *arXiv preprint arXiv:2401.17868*, 2024.

[40] Jingkai Zhou, Varun Jampani, Zhixiong Pi, Qiong Liu, and Ming-Hsuan Yang. Decoupled dynamic filter networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6647–6656, 2021.

[41] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefanet: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.