OPQ: Compressing Deep Neural Networks with One-shot Pruning-Quantization

AAAI'21
Presented by,
Pavana Prakash

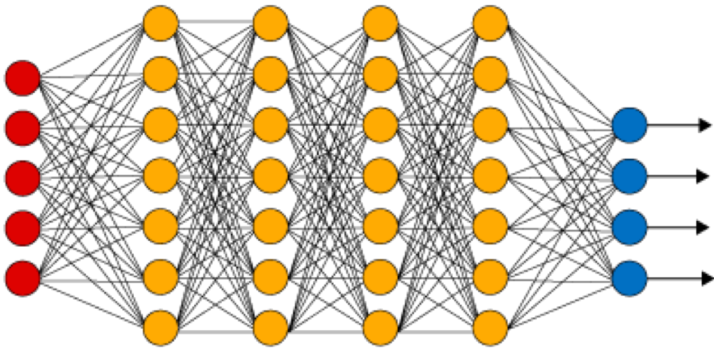
Peng Hu[1,2], Xi Peng[2] , Hongyuan Zhu[1] , Mohamed M. Sabry Aly[3] , Jie Lin[1]
[1]Institute for Infocomm Research, Singapore
[2]Sichuan University, Chengdu
[3]Nanyang Technological University, Singapore

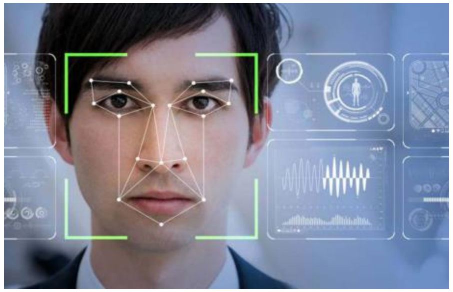
Outline

- Motivation
- Model Compression preliminary
- Unified Layer-wise Weight Pruning
- Unified Channel-wise Weight Quantization
- Experiments
- Conclusion

Introduction

- The success of modern deep neural networks (DNNs) is mainly dependent on the availability of advanced computing power and large data





Facial recognition

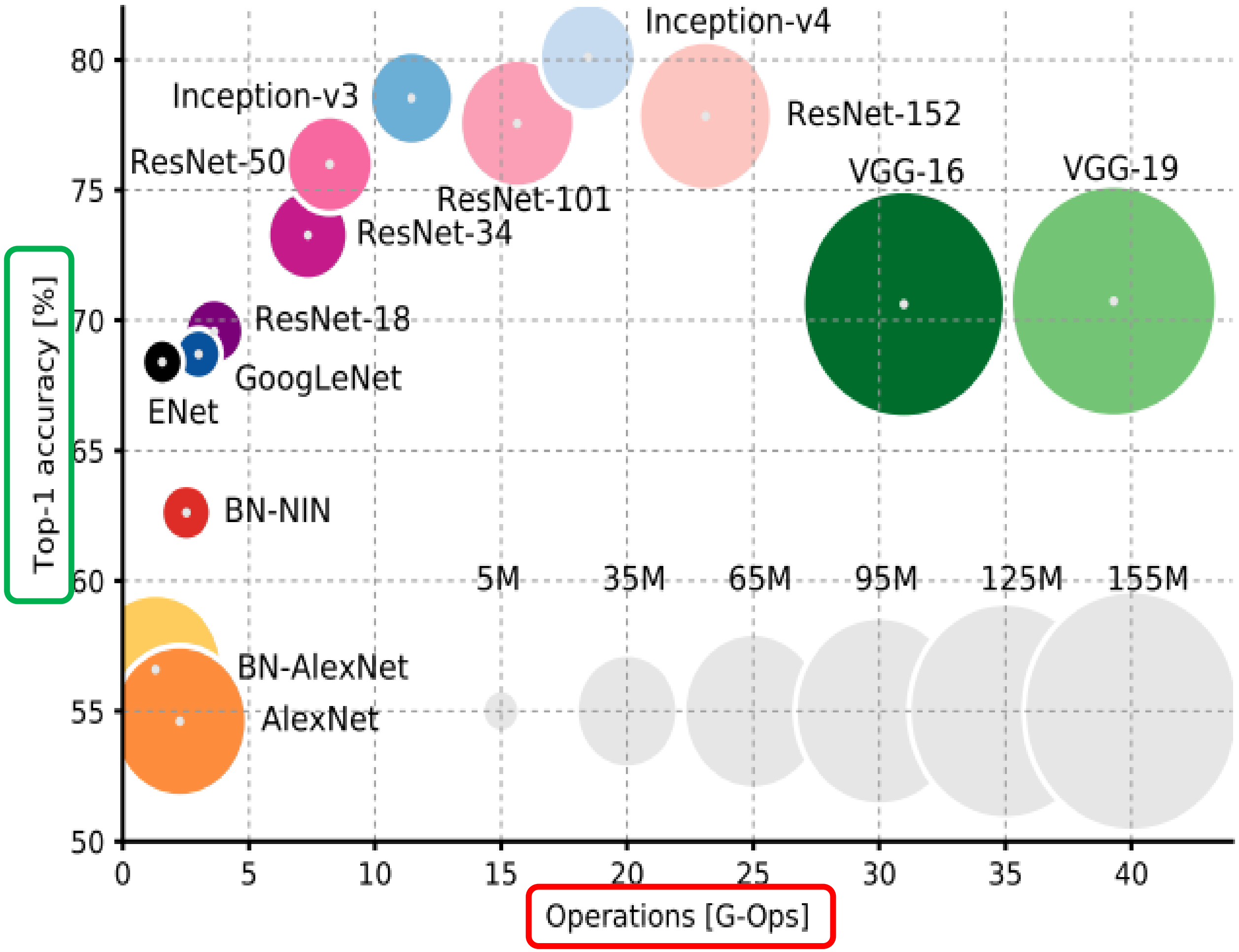
Smart healthcare

- However, its large model size and high computational operations, have impeded DNNs popularity, especially on mobile devices

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Motivation



Scaling up DNN size improves model accuracy.

Large model impedes training on resource-constrained devices!

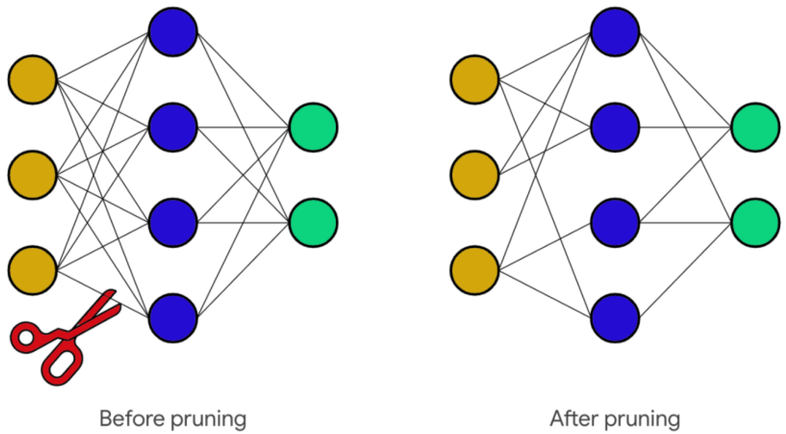
- ***Low memory resources***
- ***Expensive computation***
- ***Limited storage space***
- ***Increased latency at inference***

Source : A. Canziani, A. Paszke, and E. Culurciello. An analysis of deep neural network models for practical applications. In IEEE International Symposium on Circuits & Systems, 2016.

Preliminary - Model Compression

- Model compression methods: reduce the model size significantly

- Pruning:
  - Removes unimportant connections
  - Weights less than a salient threshold are set to zero



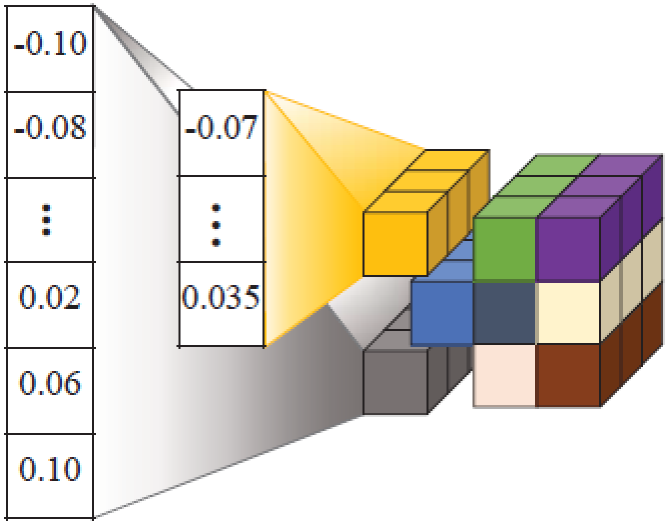
- Quantization:
  - Typically, weights : 32-bit floating point : precision not necessary
  - Reduces the number of bits required to represent a number
  - Reduces memory storage and computation costs

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Challenges

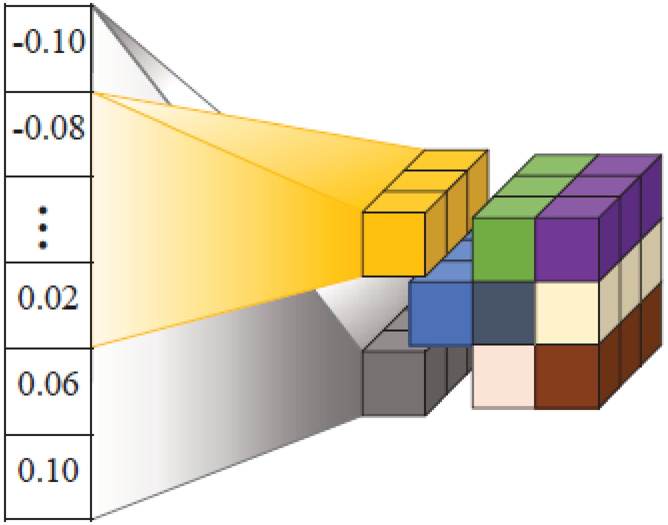
- Existing solutions are inefficient: iterative/manual fashion compression allocation + accuracy loss
- Each channel of a layer has its own channel-specific quantizer
- Expense of extra overhead introduced by channel-wise codebooks



- Pruning + quantization: almost impossible to manually tune the pruning ratios and quantization codebooks at fine-grained levels

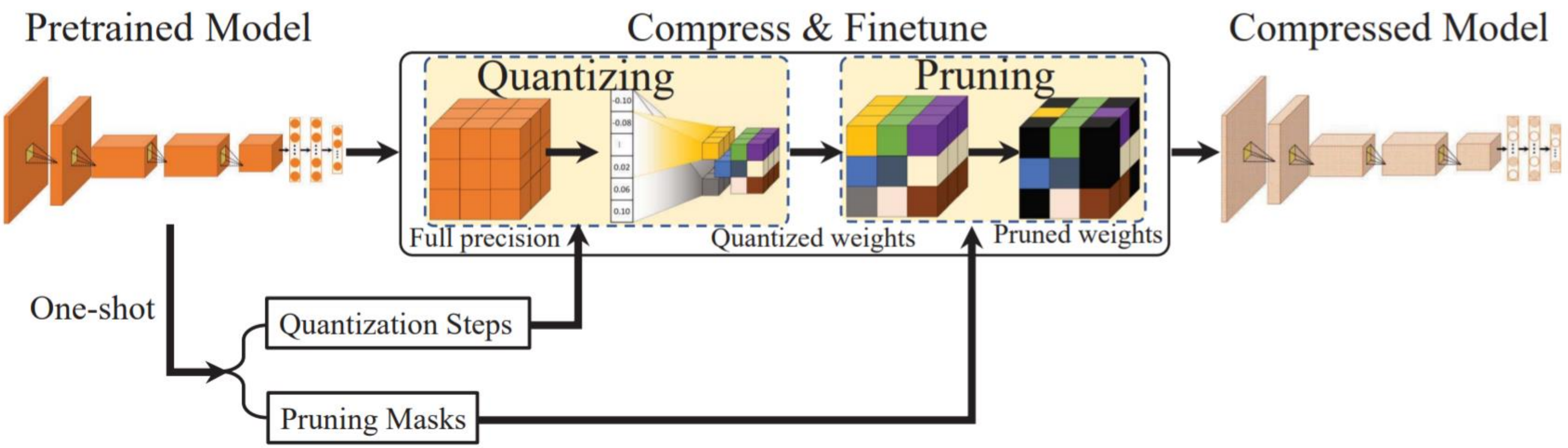
Strategy

- Proposed methods:
  - One-shot Pruning-Quantization (OPQ)
  - Unified channel-wise quantization



- Claim: Pre-trained model is sufficient for solving pruning and quantization simultaneously
- During finetuning, the compression module is fixed and only weight parameters are updated

Method pipeline



- Channels of each layer share the same quantizer (i.e., the same codebook)
- No additional computation burdens
- Given a pre-trained model, the pruning masks and quantization steps are analytically derived in one-shot
- Fixed while finetuning the compressed model.

Proposed method

- On a pre-trained model, the pruning-quantization is conducted on its weight parameters,

$$\boxed{\hat{\mathbf{W}} = \mathbf{M} \circ (\Delta \lfloor \frac{\mathbf{W}}{\Delta} \rceil)}$$

Pruning masks $\{\mathbf{M}_i\}_{i=1}^{L}$
Quantization steps $\{\Delta_i\}_i^L$

1: Compute the pruning masks $\{\mathbf{M}_i\}_{i=1}^{L}$ for all layers (see Section 3.2).
2: Calculate the qunatization steps $\{\Delta_i\}_i^L$ for all layer (see Section 3.3).
3: **for** $1, 2, \cdots, N_e$ **do**
4:     **repeat**
5:         Randomly sample a minbatch from the training set.
6:         Compress the weights using $\{\Delta_i\}_i^L$ and $\{\mathbf{M}_i\}_{i=1}^{L}$ for the model.
7:         Forward propagate with the pruned and quantized weights, and compute the cross entropy loss.
8:         Update the model weights with descending their stochastic gradient.
9:     **until** all samples selected
10: **end for**

Unified Layer-wise Weight Pruning

- General unified formulation to prune the weights W of all layers
- Find the pruning ratios of all layers $\{p_i\}_{i=1}^{L}$: percentage of weights with small magnitude,

$$p = \frac{1}{N}\sum_{i=1}^{L}\int_{-\beta_i}^{\beta_i} N_i f_i(x)dx = \frac{2}{N}\sum_{i=1}^{L}\int_{0}^{\beta_i} N_i f_i(x)dx. \quad (1)$$

symmetric range $[-\bar{\beta}_i, \beta_i]$: positive scalar value

- Pruning error,

$$\mathcal{L}_i^{\beta} = \sum_{j=1}^{N_i} (W_{ij})^2 \Big|_{|W_{ij}|\leqslant\beta_i} = 2\int_{0}^{\beta_i} N_i x^2 f_i(x)dx. \quad (2)$$

Unified Layer-wise Weight Pruning

- Pruning objective function,

$$\beta_1^*, \cdots, \beta_L^* = \underset{\beta_1, \beta_2, \cdots, \beta_L}{\arg\min} \frac{1}{N} \sum_{i=1}^{L} \mathcal{L}_i^{\beta} = \underset{\beta_1, \beta_2, \cdots, \beta_L}{\arg\min} \frac{2}{N} \sum_{i=1}^{L} \int_0^{\beta_i} N_i x^2 f_i(x) dx.$$

- For a given objective pruning rate p*,

$$\mathcal{L}(\beta_1, \cdots, \beta_L, \lambda) = \sum_{i=1}^{L} \frac{2}{N} \int_0^{\beta_i} N_i x^2 f_i(x) dx - \lambda \left(\frac{2}{N} \sum_{i=1}^{L} \int_0^{\beta_i} N_i f_i(x) dx - p^* \right)$$

- Since $f_i(x)$ could be a Laplace probability density function

$$\boxed{\frac{\partial \mathcal{L}(\beta_1, \cdots, \beta_L, \lambda)}{\partial \lambda} \approx \frac{1}{N} \sum_{i=1}^{L} N_i \left(1 - e^{-\frac{\sqrt{\lambda}}{\tau_i}} \right) - p^*}$$

- Levenberg-Marquardt algorithm, Newton-Raphson methods are used
- Finally derive binary mask

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Unified Channel-wise Weight Quantization

- Number of quantization bins required to store the unpruned weights of the channel,

$$\boxed{K_{ij} = \left\lfloor \frac{2\alpha_{ij}}{\Delta_i} \right\rceil}$$

Range $[-\alpha_{ij}, \alpha_{ij}]$: positive real value

- Per layer and the whole model,

$$\boxed{K_i = \frac{1}{\bar{N}_i} \sum_{j=1}^{C_i} \bar{N}_{ij} K_{ij},} \qquad \boxed{\frac{1}{\bar{N}} \sum_{i=1}^{L} K_i N_i \int_{\beta_i}^{+\infty} f_i(x) dx = \frac{1}{\bar{N}} \sum_{i=1}^{L} \sum_{j=1}^{C_i} \bar{N}_{ij} K_{ij} = 2^B}$$

- Common quantization function,

$$\boxed{Q_i(x) = \mathrm{sgn}(x) \Delta_i \left\lfloor \frac{|x|}{\Delta_i} \right\rceil}$$

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Unified Channel-wise Weight Quantization

- Mean-square errors caused by quantization,

$$\boxed{\mathcal{L}_q = 2\sum_{i=1}^{L}\frac{1}{\bar{N}_i}\int_{\beta_i}^{+\infty} N_i f_i(x)(x - Q_i(x))^2 dx \approx \sum_{i=1}^{L}\frac{\Delta_i^2}{12}}$$

Solve using lagrangian,

$$\boxed{\Delta_i = \frac{1}{2^{B-1}\bar{N}}\sum_{i=1}^{L}\sum_{j=1}^{C_i}\bar{N}_{ij}\alpha_{ij}}$$

$$\boxed{\lambda = \left(\frac{1}{2^{B-1}\bar{N}}\sum_{i=1}^{L}\sqrt[3]{\frac{\bar{N}}{12}}\frac{\sum_{j=1}^{C_i}\bar{N}_{ij}\alpha_{ij}}{\sqrt[3]{\sum_{j=1}^{C_i}\bar{N}_{ij}\alpha_{ij}}}\right)^3}$$

- Combining the above equations, we obtain the quantization quotas for all layers, which can be used to quantize the given DNN model

Experiments

- Models
  - AlexNet, VGG-16, ResNet-50, MobileNet-V1
- Dataset – ImageNet
- Baselines
  - **Pruning** : Data-Free Pruning (Srinivas and Babu 2015),
  - Adaptive Fastfood 32 (Yang et al. 2015),
  - Less Is More (Zhou, Alvarez, and Porikli 2016),
  - Dynamic Network Surgery (Guo, Yao, and Chen 2016),
  - Circulant CNN (Cheng et al. 2015),
  - and Constraint-Aware (Chen et al. 2018))
  - **Quantization** : Q-CNN (Wu et al. 2016),
  - Binary-Weight- Networks (Rastegari et al. 2016),
  - and ReNorm (He and Cheng 2018))
  - **pruning-quantization methods:** Deep Compression (Han, Mao, and Dally 2016), CLIP-Q (Tung and Mori 2020), and ANNC (Yang et al. 2020a)

Experiments

Method	Top-1 (%)	Top-5 (%)	Prune Rate (%)	Bit	Rate
Data-Free Pruning (Srinivas and Babu 2015)	55.60 (2.24↓)	-	36.24	32	1.57×
Adaptive Fastfood 32 (Yang et al. 2015)	**58.10** (0.69↑)	-	44.12	32	1.79×
Less Is More (Zhou, Alvarez, and Porikli 2016)	53.86 (0.57↓)	-	76.76	32	4.30×
Dynamic Network Surgery (Guo, Yao, and Chen 2016)	56.91 (0.3↑)	80.01 (-)	94.3	32	17.54×
Circulant CNN (Cheng et al. 2015)	56.8 (0.4↓)	**82.2** (0.7↓)	95.45	32	18.38×
Constraint-Aware (Chen et al. 2018)	54.84 (2.57↓)	-	**95.13**	32	20.53×
Q-CNN (Wu et al. 2016)	56.31 (0.99↓)	79.70 (0.60↓)	-	1.57	20.26×
Binary-Weight-Networks (Rastegari et al. 2016)	56.8 (0.2↑)	79.4 (0.8↓)	-	**1**	32×
Deep Compression (Han, Mao, and Dally 2016)	57.22 (0.00↑)	80.30 (0.03↑)	89	5.4	6.66×
CLIP-Q (Tung and Mori 2020)	57.9 (0.7↑)	-	91.96	3.34	119.09×
ANNC (Yang et al. 2020a)	57.52 (**1.00**↑)	80.22 (0.03↑)	92.6	3.7	118×
Ours	57.09 (0.46↑)	80.25 (**1.20**↑)	92.30	2.99	**138.96**×

Table 1: AlexNet on ImageNet.

Method	Top-1 (%)	Top-5 (%)	Prune Rate (%)	Bit	Rate
ThiNet-GAP (Luo, Wu, and Lin 2017)	67.34 (1.0↓)	87.92 (0.52↓)	94.00	32	16.63×
Q-CNN (Wu et al. 2016)	68.11 (3.04↓)	88.89 (1.06↓)	-	**1.35**	23.68×
Deep Compression (Han, Mao, and Dally 2016)	68.83 (0.33↑)	89.09 (**0.41**↑)	92.5	6.4	66.67×
CLIP-Q (Tung and Mori 2020)	69.2 (**0.7**↑)	-	94.20	3.06	180.47×
Ours	**71.39** (0.24↓)	**90.28** (0.09↓)	**94.41**	2.92	**195.87**×

Table 2: VGG-16 on ImageNet.

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Experiments

Method	Top-1 (%)	Top-5 (%)	Prune Rate (%)	Bit	Rate
To Prune or Not To Prune (Zhu and Gupta 2017)	69.5 (1.1↓)	89.5 (0.0↑)	50	32	2×
Deep Compression (Han, Mao, and Dally 2016)	65.93 (4.97↓)	86.85 (3.05↓)	-	3	10.67×
ReNorm (He and Cheng 2018)	65.93 (9.75↓)	83.48 (6.37↓)	-	4	8×
HAQ (Wang et al. 2019)	67.66 (3.24↓)	88.21 (1.69↓)	-	3	10.67×
CLIP-Q (Tung and Mori 2020)	70.3 (0.0↑)	-	47.36	4.61	13.19×
ANNC (Yang et al. 2020a)	69.71 (1.19↓)	89.14 (0.76↓)	-	3	10.67×
ANNC (Yang et al. 2020a)	66.49 (4.41↓)	87.29 (2.61↓)	58	**2.8**	26.7×
Ours	**70.83 (0.55↑)**	**89.70 (0.27↑)**	57.78	3.26	23.26×
Ours	70.24 (0.04↓)	89.30 (0.13↓)	**67.66**	3.08	**32.15**×

Table 3: MobileNet-V1 on ImageNet.

Method	Top-1 (%)	Top-5 (%)	Prune Rate (%)	Bit	Rate
ThiNet (Luo, Wu, and Lin 2017)	71.01 (1.87↓)	90.02 (1.12↓)	51.76	32	2.07×
Deep Compression (Han, Mao, and Dally 2016)	76.15 (0.00↑)	92.88 (0.02↑)	-	4	8×
HAQ (Wang et al. 2019)	76.14 (0.01↓)	92.89 (0.03↑)	-	4	8×
ACIQ (Banner, Nahshan, and Soudry 2019)	75.3 (0.8↓)	-	-	4	8×
CLIP-Q (Tung and Mori 2020)	73.7 (**0.6**↑)	-	69.38	3.28	31.81×
Ours	**76.41** (0.40↑)	**93.04 (0.11↑)**	**74.14**	**3.25**	**38.03**×

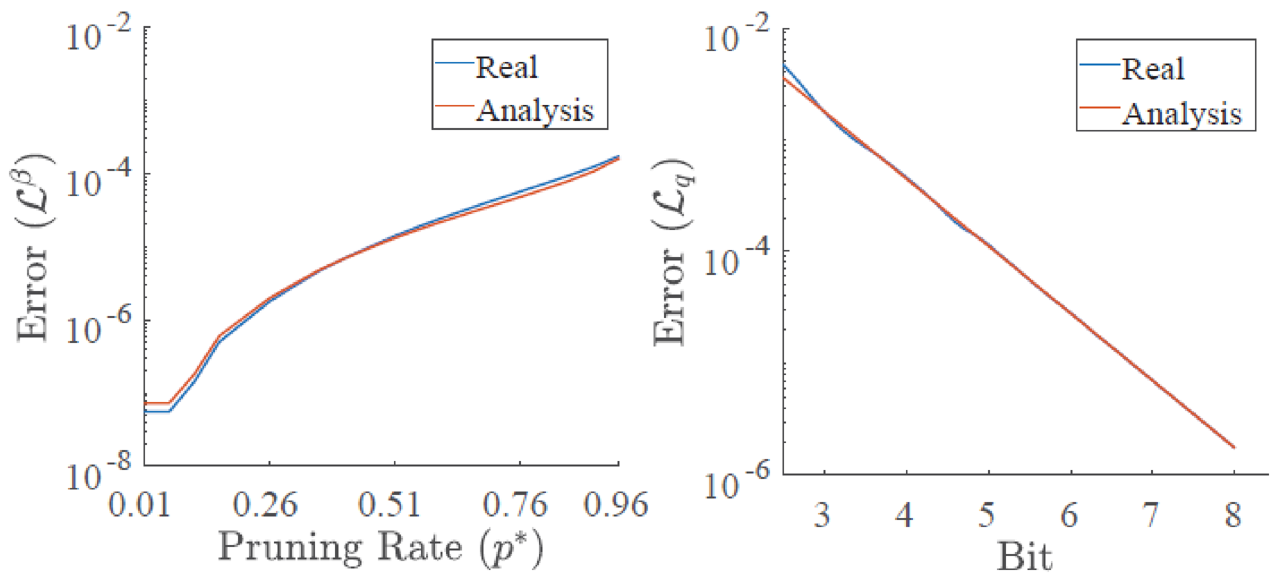
Table 4: ResNet-50 on ImageNet.

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Error Analysis

- Pruning: Laplace probability density function approximates the real pruning error well, demonstrating the validity of the equation
- Quantization: approximation error is in a good agreement with the real quantization error



(a) Pruning Error Analysis. (b) Quantization Error Analysis.

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Conclusion

- The paper proposed a novel One-shot Pruning-Quantization method (OPQ) to compress DNN models
- Their method has addressed two challenging problems in network compression.
  - First, different from the prior art, OPQ is a one-shot compression method without manual tuning or iterative optimization of the compression strategy
  - Second, their unified channel-wise pruning to enforce all channels of each layer to share a common codebook, avoids the overheads brought by the traditional channel-wise quantization
- Experiments show that their method achieves superior results comparing to the state-of-the-art.

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Discussion

- For future work, we will explore how to further compress DNN model
- Implement their method with custom hardware architecture
- Validate the inference efficiency of the compressed models on practical hardware platforms.
- Potential negative impact: the compression bias caused by OPQ because of unusual weight distribution, too lower objective compression rate, etc

THANK YOU

Pavana Prakash

Department of Electrical and Computer Engineering
University of Houston
Houston, TX