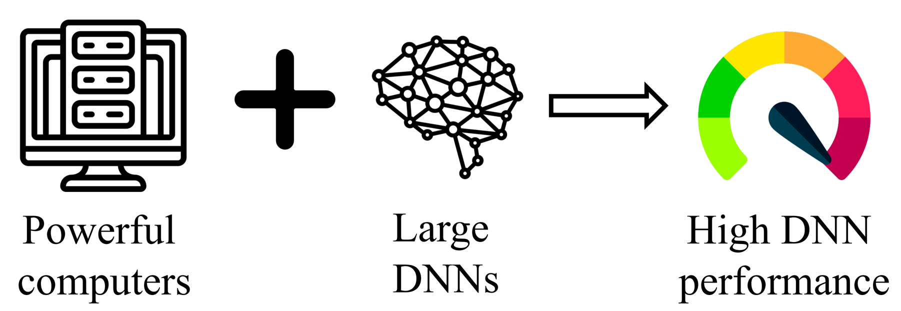
LDP: Learnable Dynamic Precision for Efficient Deep Neural Network Training and Inference

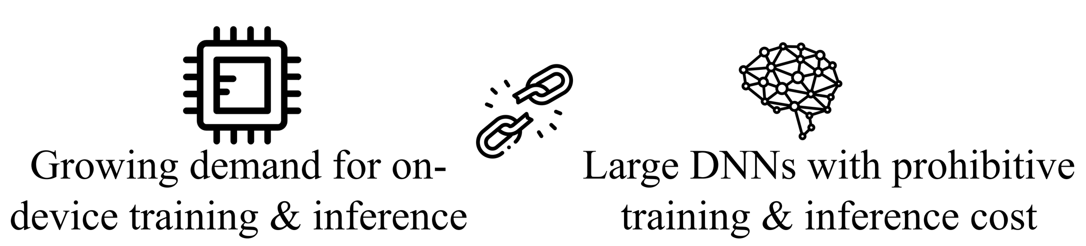
Zhongzhi Yu, Yonggan Fu, Shang Wu, Mengquan Li, Haoran You, Yingyan Lin
Rice University

Background and Motivation

- Deep neural networks' (DNNs) high performance comes with **large DNNs** and **powerful computers**



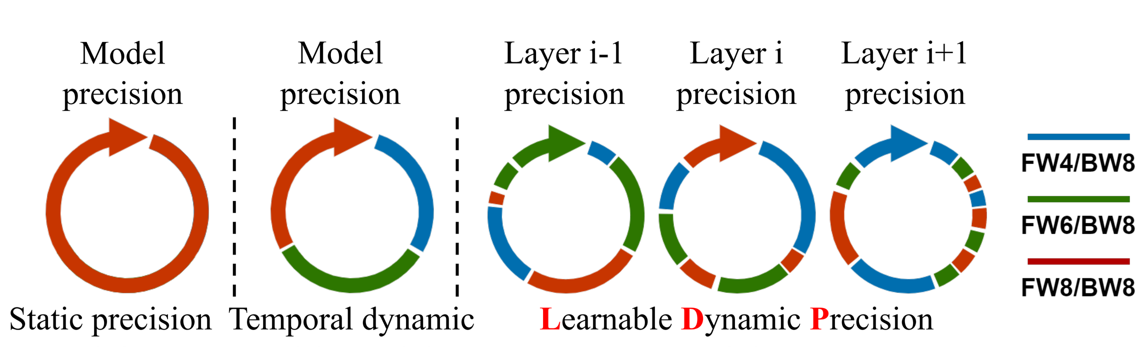
- Deep neural networks (DNNs) are **costly**:
  - Prohibitive **training** cost:
    - 10^{18} FLOPs for training ResNet-50@ImageNet
  - Excessive **inference** cost:
    - 10^{9} FLOPs for single-image inference with ResNet-50@ImageNet



> **Low-precision Method**: a promising direction to narrow the gap

Existing Low-precision Methods

- Static low-precision training: [S. Banner, NeurIPS'18]
  - Use same precision during training process
  - ☹ Large **accuracy gap** under low-precision
- Temporal dynamic low-precision training: A promising direction [Y. Fu, NeurIPS'20], [Y. Fu, ICLR'21]
  - Assign different precisions for different training stages for better accuracy-efficiency trade-off
  - ☹ Only consider **temporal** dynamic precision
  - ☹ Need extra efforts in **hyperparams finetuning**



Motivating Observations

❓ Is **only the temporal** dynamic precision enough?

- Inspirations from previous works:
  - Different layers have **different sensitivities** [C. Zhang, ICML'19] [K. Greff, ICLR'17]
  - Precision has **similar effect as learning rate** [Y. Fu, ICLR'20]

> **Spatial dynamic precision** allocation is also important

- Exploration on the importance of **spatial and temporal** precision allocation
  - Settings:
    - **Temporal**: Change precision at 30, 60, 90 epochs
    - **Spatial**: [a,b,c]: Assign a,b,c-bit to first three blocks, respectively
  - Insights:
    - Both **temporal and spatial** precision allocations impact the training accuracy-efficiency trade-off.
    - Different combination lead to **0.75%** accuracy gap.

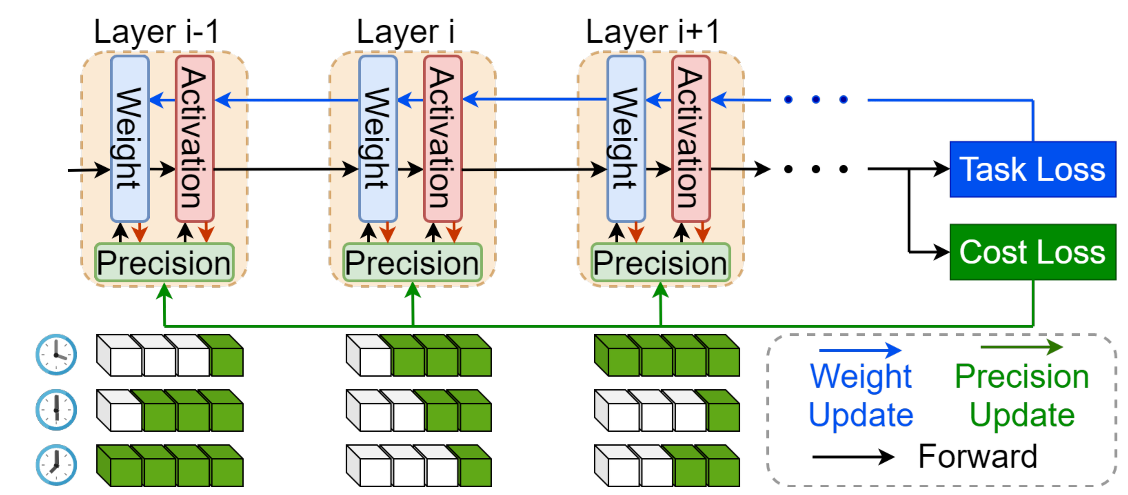
Training Stages [0-th,30-th]	[30-th,60-th]	[60-th,90-th]	[90-th,160-th]	Savings over static (%)	Accuracy/%
[4, 6, 8]	[6, 8, 4]	[8, 4, 6]	[8, 8, 8]	1.10×10^{8}	68.88 ± 0.21
[6, 8, 4]	[8, 4, 6]	[4, 6, 8]	[8, 8, 8]	1.10×10^{8}	69.63 ± 0.14
[8, 4, 6]	[4, 6, 8]	[6, 8, 4]	[8, 8, 8]	1.10×10^{8}	69.36 ± 0.16

❓ How to **automatically** generate the **spatial and temporal** precision allocation during training?

Contributions

- Learnable dynamic precision (LDP): a framework to **automatically** learn the **spatial** and **temporal** precision allocation during training
- Develop a **differentiable** method to enable **end-to-end** learnable dynamic precision DNN training
- Achieve the SOTA accuracy-efficiency trade-off on **seven DNNs**, **five datasets** and **three tasks** in both training and inference

LDP: Method



- **Automatically** learn the **spatial and temporal** precision allocation during training
- Enabler 1: Differentiable learnable precision
  - Challenge: How to achieve a differentiable precision learning on top of the **discrete precision**
  - Vanilla quantization process:

$$\text{Quantization Output} = \text{Round}\left(\frac{\text{Input} - \text{Zero Point}}{\text{Quantization Step}}\right) + \text{Zero Point}$$

$$\text{Quantization Step} = \frac{\text{Dynamic Range}}{2^{\text{Precision}} - 1}$$

  - Use a learnable quantization step with **a layer-wise learnable parameter β**

$$\text{Learnable Quantization Step} = \frac{\text{Dynamic Range}}{2^{\beta \times \text{Precision}} - 1}$$

- Enabler 2: Loss function design
  - Challenge: Balance accuracy and efficiency when **scales of L_{task} and L_{cost} vary** among different tasks and during training
  - Penalize training cost when exceeding threshold T

$$L_{cost} = \begin{cases} 0, & \text{if } C < T \\ C, & \text{if } C \geq T \end{cases}$$

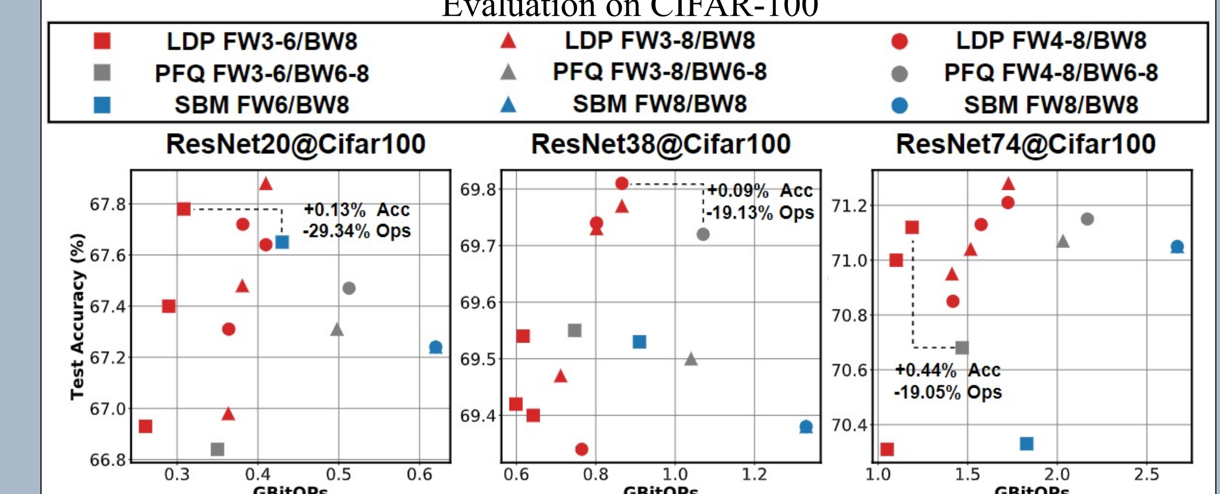
  - Balance each layer's **precision gradient** w.r.t. L_{task} and L_{cost}

$$\text{Precision Grad} = \text{Grad}(L_{Task}) + \alpha \times \text{Grad}(L_{Cost}) \times \frac{\text{Mean}(\text{Abs}(\text{Grad}(L_{Task})))}{\text{Mean}(\text{Abs}(\text{Grad}(L_{Cost})))}$$

LDP: Evaluation

- **Seven models** on **five datasets** from **three tasks**:
  - ResNet@CIFAR for image classification
  - ResNet18/DeiT-Tiny@ImageNet for image classification
  - PAN@Urban-100 for image super-resolution
  - Transformer@Wiki-101 for language modeling
- Three baselines:
  - Static low-precision training: SBM [S Banner, NeurIPS'18]
  - Dynamic low-precision training: PFQ [Y. Fu, NeurIPS'19] & CPT [Y. Fu, ICLR'20]

Evaluation on CIFAR-100



Datasets		CIFAR-100			
Model	Method	Precision	Acc(%)	Training Cost(GBitOps)	Inference Cost(GBitOps)
ResNet-38	SBM	FW8/BW8	69.38	1.33e8	2.69
	PFQ	FW3-8/BW8	69.50	1.04e8	2.69
	LDP	FW3-8/BW8	**69.77**	**0.87e8**	**1.35**
	Improv.		**+0.27**	**-16.3%**	**-49.8%**
	SBM	FW8/BW8	69.38	1.33e8	2.69
	PFQ	FW4-8/BW8	69.72	1.07e8	2.69
	LDP	FW4-8/BW8	**69.81**	**0.87e8**	**1.33**
	Improv.		**+0.09**	**-18.7%**	**-50.6%**
ResNet-74	SBM	FW8/BW8	71.05	2.67e8	5.42
	PFQ	FW3-8/BW8	71.07	2.03e8	5.42
	LDP	FW3-8/BW8	**71.28**	**1.72e8**	**2.83**
	Improv.		**+0.21**	**-15.3%**	**-47.8%**
	SBM	FW8/BW8	71.05	2.67e8	5.42
	PFQ	FW4-8/BW8	71.15	2.16e8	5.42
	LDP	FW4-8/BW8	**71.21**	**1.72e8**	**2.78**
	Improv.		**+0.06**	**-20.4%**	**-48.7%**

- CIFAR-100: **↑0.44%** accuracy, **↓29.34%** training cost, **↓50.6%** inference cost
- ImageNet: **↓30.8%** inference cost and **↓8.1%** training cost with comparable accuracy

Evaluation on ImageNet

Model	Method	Precision	Acc(%)	Training Cost (GBitOps)	Inference Cost (GBitOps)
ResNet-18	SBM	FW8/BW8	69.60	2.86e9	1.46e1
	CPT	FW4-8/BW8	69.64	1.99e9	1.46e1
	PFQ	FW4-8/BW6-8	69.12	2.47e9	1.46e1
	LDP	FW4-8/BW8	69.62	1.83e9	1.01e1
	Improv.		-0.02	-8.1%	-30.8%
DeiT-Tiny	SBM	FW8/BW8	71.71	4.74e9	0.96e1
	CPT	FW4-8/BW8	71.84	3.29e9	0.96e1
	PFQ	FW4-8/BW6-8	71.70	3.96e9	0.96e1
	LDP	FW4-8/BW8	71.92	3.08e9	0.67e1
	Improv.		+0.08	-6.4%	-30.2%

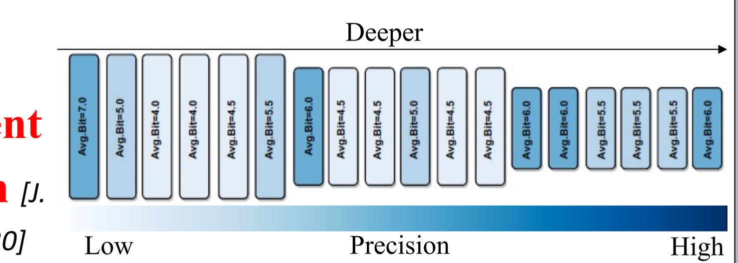
- WikiText-103: **↓0.96** perplexity (the lower, the better) with **↓25.9%** training cost

Evaluation of Transformer on WikiText-103

Method	Precision	Perplexity	Training Cost (GBitOps)
SBM	FW8/BW8	31.77	9.87e5
LDP	FW4-8/BW8	30.81	7.31e5
Improv.		-0.96	-25.9%

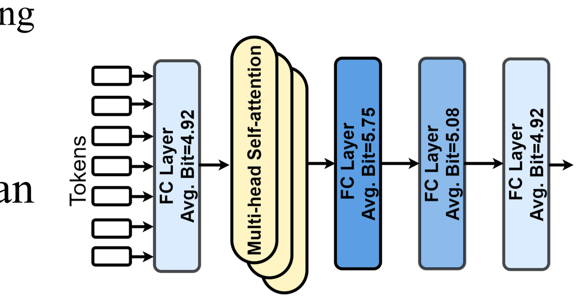
LDP: Visualization

- Vis. 1: Learned precision is **consistent with manual design** [J. Shen, AAAI'20] [Y. Wang, ISP'20]
  - Higher precision in
    - Blocks after downsampling
    - Deep blocks with lowest spatial resolution



Vis. 1: ResNet-38@CIFAR-100 block-wise average precision

- Vis. 2: Learned precision can **guide model design**
  - Decreased precision (higher redundancy) in the last two FC layers
  - Consistent with the work studying FC layers [J. Guo, arXiv'21]



Vis. 2: DeiT-Tiny@ImageNet layer-wise average precision