[ICML-2024 (Oral)] DoRA: Weight-Decomposed Low-Rank Adaptation

组会汇报

王雷

南开大学计算机学院

2024 年 12 月 19 日

DoRA: Weight-Decomposed Low-Rank Adaptation

- Institution： NVIDIA, HKUST
- Paper: `https://arxiv.org/pdf/2402.09353`
- Code: `https://github.com/NVlabs/DoRA`
- Website: `https://nbasyl.github.io/DoRA-project-page/`
- Huggingface: `https://github.com/huggingface/peft/releases/tag/v0.10.0`
- Citations: 193

PEFT v.s. FT

- 为了使大规模通用模型适应下游任务，通常采用全微调(FT)

PEFT v.s. FT

- 为了使大规模通用模型适应下游任务，通常采用全微调 (FT)
- 然而 FT 成本太高

PEFT v.s. FT

- 为了使大规模通用模型适应下游任务，通常采用全微调 (FT)
- 然而 FT 成本太高
- 因此，参数高效微调（PEFT）应运而生

PEFT v.s. FT

- 为了使大规模通用模型适应下游任务，通常采用全微调 (FT)
- 然而 FT 成本太高
- 因此，参数高效微调（PEFT）应运而生
- 其中，LoRA [1] 最受欢迎，归因于简单有效这两个特性

PEFT v.s. FT

- 为了使大规模通用模型适应下游任务，通常采用全微调 (FT)
- 然而 FT 成本太高
- 因此，参数高效微调（PEFT）应运而生
- 其中，LoRA [1] 最受欢迎，归因于简单有效这两个特性
- 然而，LoRA 与 FT 仍然有着性能差距，一般都归咎于可训练参数有限，却没有探究其本质原因

PEFT v.s. FT

- 为了使大规模通用模型适应下游任务，通常采用全微调 (FT)
- 然而 FT 成本太高
- 因此，参数高效微调（PEFT）应运而生
- 其中，LoRA [1] 最受欢迎，归因于简单有效这两个特性
- 然而，LoRA 与 FT 仍然有着性能差距，一般都归咎于可训练参数有限，却没有探究其本质原因
- 受权重归一化的启发 [2]，DoRA 将模型预训练权重分解为幅值和方向分量，然后对比 LoRA 和 FT 引入的前两者的变化

权重分解分析

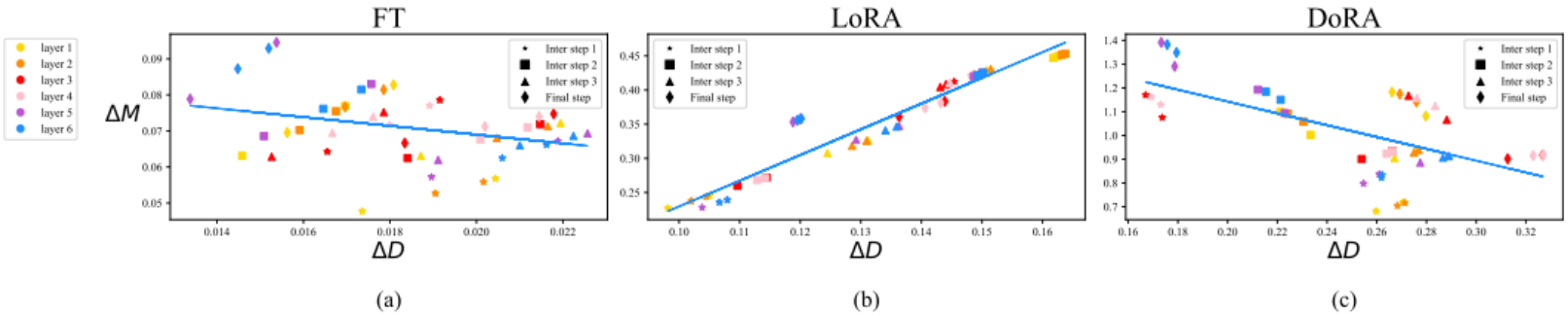


图 1: 幅值和方向的更新

- $$\Delta M_{\mathrm{FT}}^{t} = \frac{\sum_{n=1}^{k} \left| m_{\mathrm{FT}}^{n,t} - m_{0}^{n} \right|}{k}$$
$$\Delta D_{\mathrm{FT}}^{t} = \frac{\sum_{n=1}^{k} \left(1 - \cos\left(V_{\mathrm{FT}}^{n,t}, W_{0}^{n}\right)\right)}{k} \quad (1)$$

- FT：这两者有一个更新量较大即可
- LoRA：LoRA 明显是正相关
- 因此，DoRA 解耦这两者去单独优化

DoRA

- $$W = m\frac{V}{\|V\|_c} = \|W\|_c\frac{W}{\|W\|_c} \tag{2}$$
- $$\begin{aligned} W' &= \underline{m}\frac{V+\Delta V}{\|V+\Delta V\|_c} \\ &= \underline{m}\frac{W_0+\underline{BA}}{\|W_0+\underline{BA}\|_c}. \end{aligned} \tag{3}$$
- 动态归一化范数

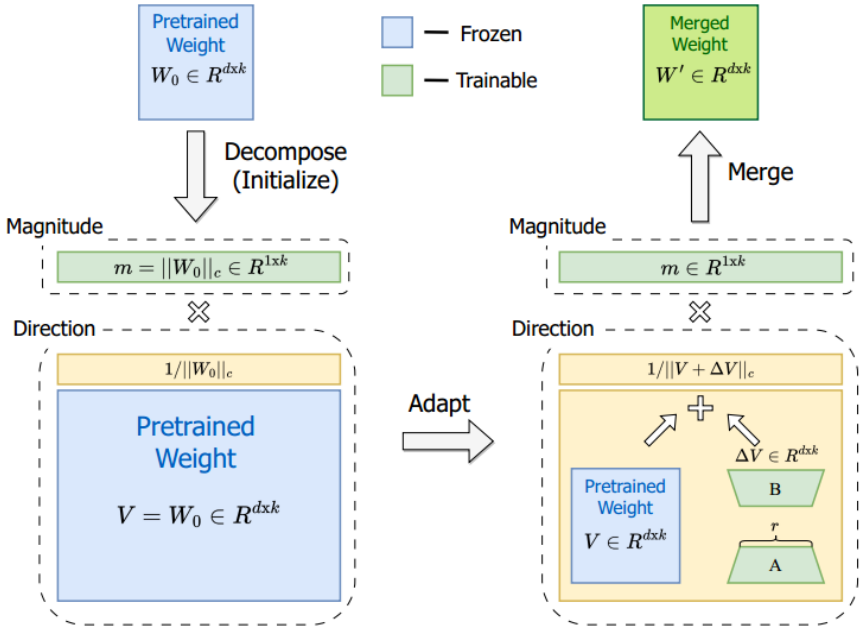


图 2: DoRA 概述。

DoRA 梯度分析

DoRA 表示为：

$$W' = \underline{m}\frac{V + \Delta V}{\|V + \Delta V\|_c} = \underline{m}\frac{W_0 + \underline{BA}}{\|W_0 + \underline{BA}\|_c}, \tag{4}$$

令 $V' = V + \Delta V$，公式 (4) 可以写为：

$$W' = m\frac{V'}{\|V'\|_c}, \tag{5}$$

首先计算 W' 关于 V' 的梯度：

$$\frac{\partial W'}{\partial V'} = m\frac{\partial}{\partial V'}\left(\frac{V'}{\|V'\|_c}\right), \tag{6}$$

注意到 $\frac{V'}{\|V'\|_c}$ 是单位化操作，可以分为两部分：

$$\frac{\partial}{\partial V'}\left(\frac{V'}{\|V'\|_c}\right) = \frac{1}{\|V'\|_c}\left(I - \frac{V'V'^T}{\|V'\|_c^2}\right), \tag{7}$$

DoRA 梯度分析

根据公式 (6) 和公式 (7)，结合 $\mathcal{L}$ 关于 W' 的梯度 $\nabla_{W'}\mathcal{L}$，可得：

$$\nabla_{V'}\mathcal{L} = \frac{m}{\|V'\|_c}\left(I - \frac{V'V'^T}{\|V'\|_c^2}\right)\nabla_{W'}\mathcal{L}. \tag{8}$$

正交投影矩阵剔除 V' 方向梯度分量，使得在正交空间内，梯度分布更加均匀，同时与梯度缩放协同，使得 $\nabla_{W'}\mathcal{L}$ 的协方差矩阵接近单位矩阵。

由于 $V' = V + \Delta V$ ，将 V' 的变化完全归因于 ΔV 的变化：

$$\frac{\partial \mathcal{L}}{\partial \Delta V} = \frac{\partial \mathcal{L}}{\partial V'}. \tag{9}$$

这表示当 V 是固定不变的时，对 ΔV 的梯度等价于对 V' 的梯度。

因此，分解带来了训练稳定性的提升。

DoRA 梯度分析

对于幅值 m，权重矩阵 W' 关于 m 的偏导数为：

$$\frac{\partial W'}{\partial m} = \frac{V'}{\|V'\|_c} \tag{10}$$

结合链式法则，得到损失函数关于 m 的梯度：

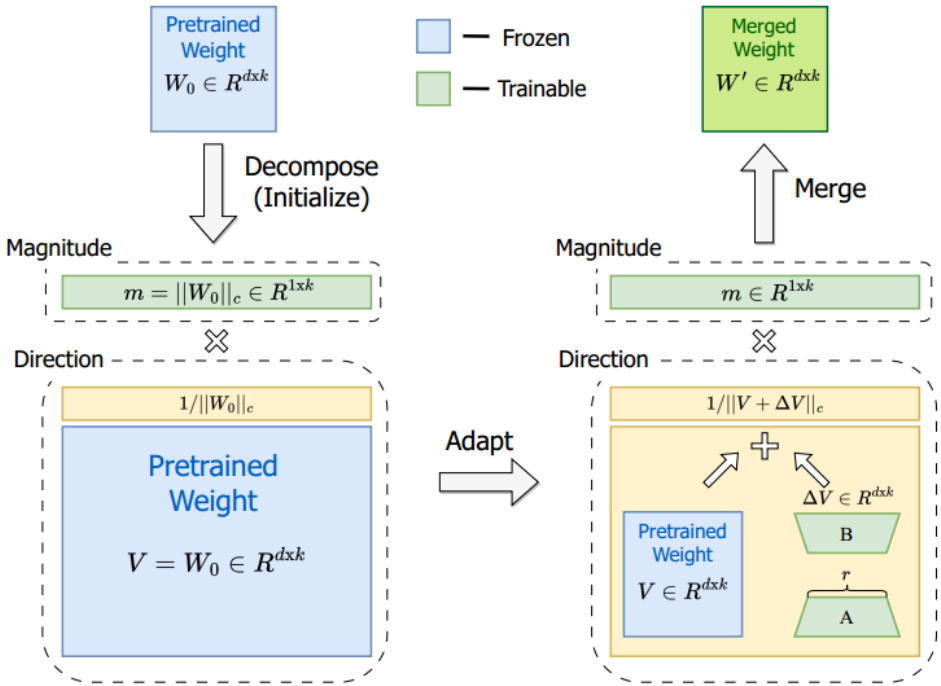
$$\nabla_m \mathcal{L} = \nabla_{W'} \mathcal{L} \cdot \frac{V'}{\|V'\|_c} \tag{11}$$

可以观察到幅值梯度 $\nabla_m \mathcal{L}$ 直接取决于梯度 $\nabla_{W'} \mathcal{L}$ 在 V' 方向上的投影，方向梯度会使 V' 偏离 V,V' 偏离后，$\nabla_{W'} \mathcal{L}$ 与 V' 之间夹角增大，从而导致 $\cos(\nabla_{W'} \mathcal{L}, V')$ 减小，进而导致投影量 $\nabla_{W'} \mathcal{L} \cdot V'$ 减小，因此，幅值梯度 $\nabla_m \mathcal{L}$ 也就变小。

减少训练开销

公式 (4) 中的 $\|V + \Delta V\|_c$ 在反向传播时更新需要额外显存，因此，我们可以选择不更新，将其视为常数。因此，$\nabla_{V'}\mathcal{L}$ 可以重写为:

$$\nabla_{V'}\mathcal{L} = \frac{m}{C}\nabla_{W'}\mathcal{L}, \text{ 其中 } C = \|V'\|_c \tag{12}$$



Commonsense Reasoning

Model	PEFT Method	# Params (%)	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
ChatGPT	-	-	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA-7B	Prefix	0.11	64.3	76.8	73.9	42.1	72.1	72.9	54.0	60.6	64.6
	Series	0.99	63.0	79.2	76.3	67.9	75.7	74.5	57.1	72.4	70.8
	Parallel	3.54	67.9	76.4	78.8	69.8	78.9	73.7	57.3	75.2	72.2
	LoRA	0.83	68.9	80.7	77.4	78.1	78.8	77.8	61.3	74.8	74.7
	DoRA† (Ours)	0.43	70.0	82.6	79.7	83.2	80.6	80.6	65.4	77.6	**77.5**
	DoRA (Ours)	0.84	69.7	83.4	78.6	87.2	81.0	81.9	66.2	79.2	**78.4**
LLaMA-13B	Prefix	0.03	65.3	75.4	72.1	55.2	68.6	79.5	62.9	68.0	68.4
	Series	0.80	71.8	83	79.2	88.1	82.4	82.5	67.3	81.8	79.5
	Parallel	2.89	72.5	84.9	79.8	92.1	84.7	84.2	71.2	82.4	81.4
	LoRA	0.67	72.1	83.5	80.5	90.5	83.7	82.8	68.3	82.4	80.5
	DoRA† (Ours)	0.35	72.5	85.3	79.9	90.1	82.9	82.7	69.7	83.6	**80.8**
	DoRA (Ours)	0.68	72.4	84.9	81.5	92.4	84.2	84.2	69.6	82.8	**81.5**
LLaMA2-7B	LoRA	0.83	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
	DoRA† (Ours)	0.43	72.0	83.1	79.9	89.1	83.0	84.5	71.0	81.2	**80.5**
	DoRA (Ours)	0.84	71.8	83.7	76.0	89.1	82.6	83.7	68.2	82.4	**79.7**
LLaMA3-8B	LoRA	0.70	70.8	85.2	79.9	91.7	84.3	84.2	71.2	79.0	80.8
	DoRA† (Ours)	0.35	74.5	88.8	80.3	95.5	84.7	90.1	79.1	87.2	**85.0**
	DoRA (Ours)	0.71	74.6	89.3	79.9	95.5	85.6	90.5	80.4	85.8	**85.2**

图 3: LLaMA 7B/13B、LLaMA2 7B 和 LLaMA3 8B 与各种 PEFT 方法在八个常识推理数据集上的准确度比较。

Image/Video-Text Understanding

Table 2. The multi-task evaluation results on VQA, GQA, NVLR^2 and COCO Caption with the VL-BART backbone.

Method	# Params (%)	VQA^{v2}	GQA	NVLR^2	COCO Cap	Avg.
FT	100	66.9	56.7	73.7	112.0	77.3
LoRA	5.93	65.2	53.6	71.9	115.3	76.5
DoRA (Ours)	5.96	65.8	54.7	73.1	115.9	**77.4**

Table 3. The multi-task evaluation results on TVQA, How2QA, TVC, and YC2C with the VL-BART backbone.

Method	# Params (%)	TVQA	How2QA	TVC	YC2C	Avg.
FT	100	76.3	73.9	45.7	154	87.5
LoRA	5.17	75.5	72.9	44.6	140.9	83.5
DoRA (Ours)	5.19	76.3	74.1	45.8	145.4	**85.4**

图 4: 多模态微调任务上的准确度比较。

Robustness of DoRA towards different rank settings

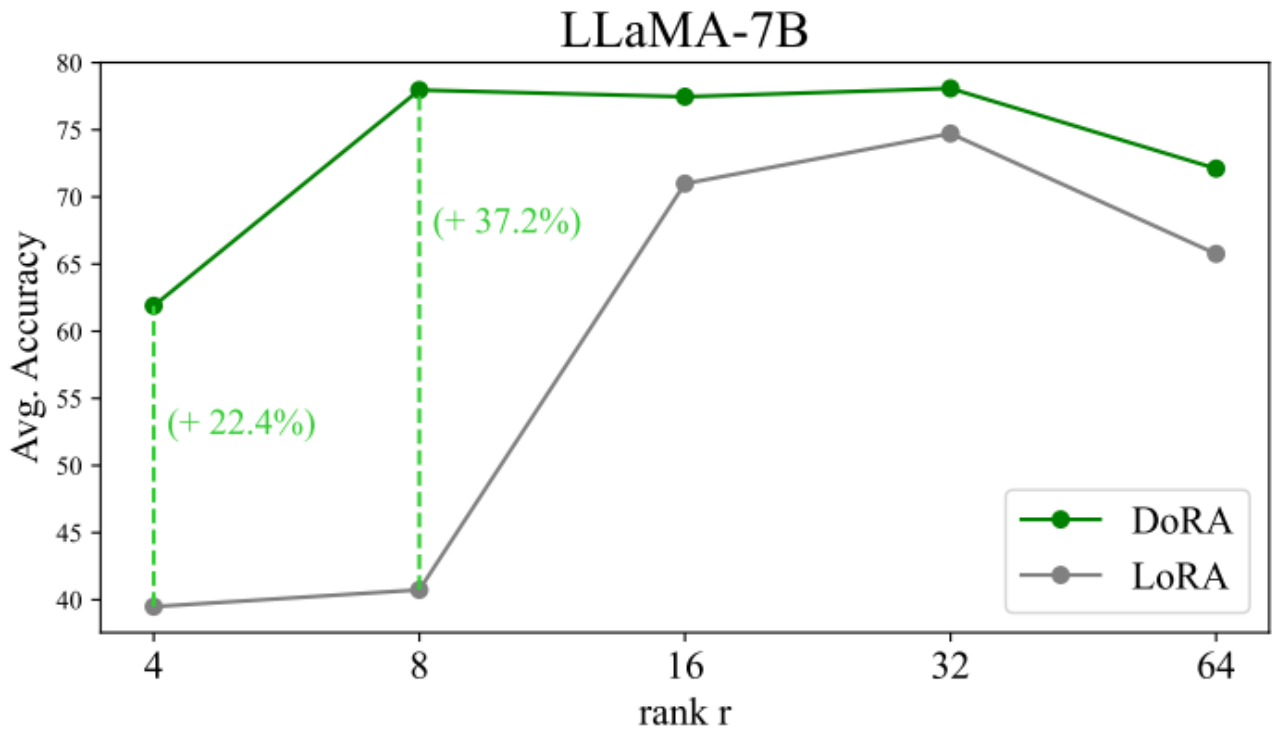


图 5: 在常识推理任务中，LLaMA-7B 的 LoRA 和 DoRA 不同秩的平均准确度。

Text-to-Image Generation

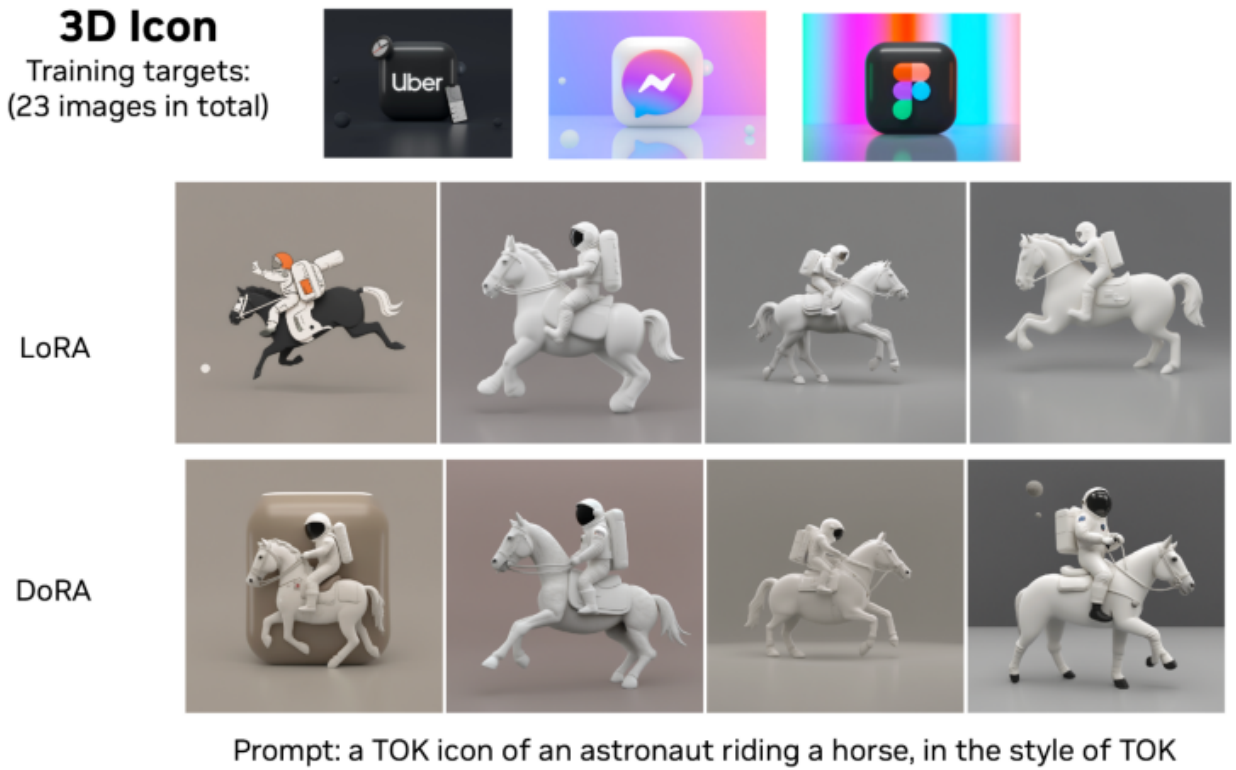


图 6: 使用在 3D Icon 训练集上微调的 LoRA 和 DoRA 的 SDXL 生成的图像。

Text-to-Image Generation

图 7: 使用在乐高训练集上微调的 LoRA 和 DoRA 的 SDXL 生成的图像。

- 解耦去优化本身会提高训练稳定性，DoRA 这种方式可以理解为归一化到超球面去优化。
- DoRA 分解为幅值和方向，取得了效果，但感觉有效的原因不够本质。
- 原因一：DoRA 让微调时的行为接近 FT，负斜率是接近了，但是更新的 ΔD 和 ΔM 明显大于 LoRA 和 FT。
- 原因二：DoRA 并未证明 FT 的 ΔD 和 ΔM 分布和性能正相关，动机存疑。

[1] 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.

[2] Tim Salimans and Durk P Kingma.
Weight normalization: A simple reparameterization to accelerate training of deep neural networks.
Advances in neural information processing systems, 29, 2016.

Thanks!