Parameter-Efficient Fine-Tuning (LoRA-based)

2025년도 동계 세미나

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
안성욱

Outline

- Introduction
  - Parameter-Efficient Fine-Tuning (PEFT)
    - LoRA-based methods
- VeRA: Vector-based Random Matrix Adaptation
  - ICLR 2024
- DoRA: Weight-Decomposed Low-Rank Adaptation
  - ICML 2024 Oral

Introduction to PEFT

- Parameter-Efficient Fine-Tuning (PEFT)
  - Prompt-based methods
  - Adapter-based methods
  - Mapping-based methods
  - LoRA-based methods

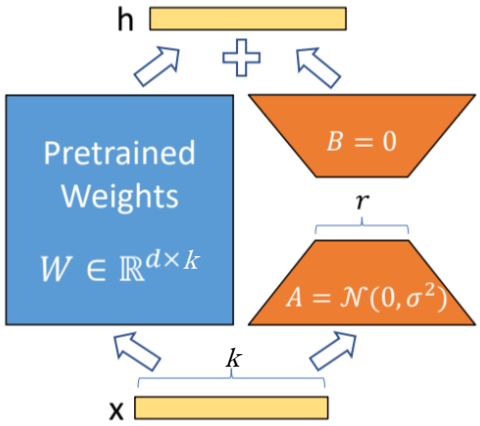
1) Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." *arXiv preprint arXiv:2106.09685* (2021).

Introduction to PEFT

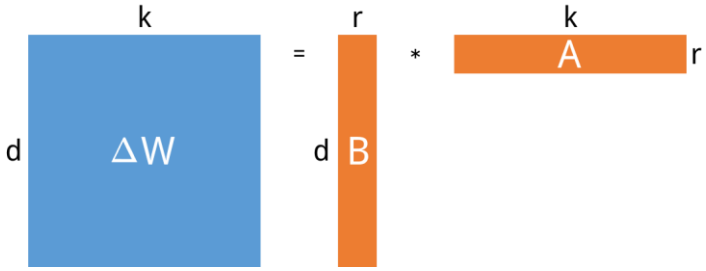
- LoRA: Low-Rank Adaptation of Large Language Models[1)]
  - 배경
    - 효과적인 fine-tuning을 위한 방법론 제시
  - 문제
    - 기존 fine-tuning 방식은 parameter 수와 hardware(GPU) 요구 사항이 높음
  - LoRA의 목적
    - 모델의 pre-trained weight을 고정한 채, weight의 변화량을 low-rank matrix로 표현함으로써 parameter efficiency를 극대화함

Introduction to PEFT

- LoRA: Low-Rank Adaptation of Large Language Models[1)]
  - W 고정, A와 B만 학습
    - $h = W_0x + \Delta Wx = W_0x + BAx$
  - 왜 low-rank인가?
    - A와 B를 통해 ΔW를 근사하는 과정
      - 선형대수학의 SVD와 유사함
    - ΔW를 직접 저장하는 대신 A와 B로 표현하여 저장 공간을 절약하고 계산량을 줄임



< LoRA architecture >

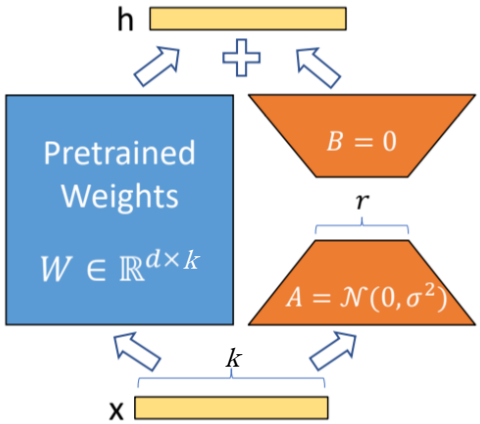


$B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}$

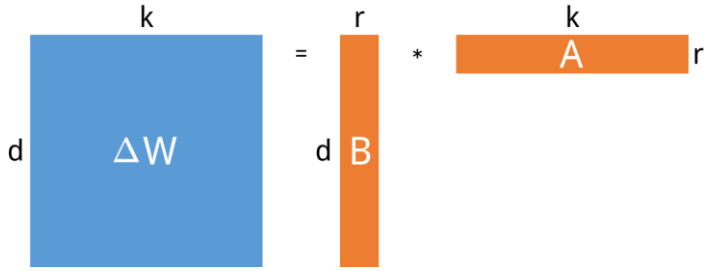
$r \ll \min(d, k)$

Introduction to PEFT

- LoRA: Low-Rank Adaptation of Large Language Models[1)]
  - A와 B의 초기화
    - A: Gaussian normal distribution을 사용한 초기화
      - Random 초기화를 통한 다양성 확보
    - B: 영행렬로 초기화
      - 처음 단계에서 pre-trained weight만으로 모델 출력 결정



< LoRA architecture >



$$B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}$$

$$r \ll \min(d, k)$$

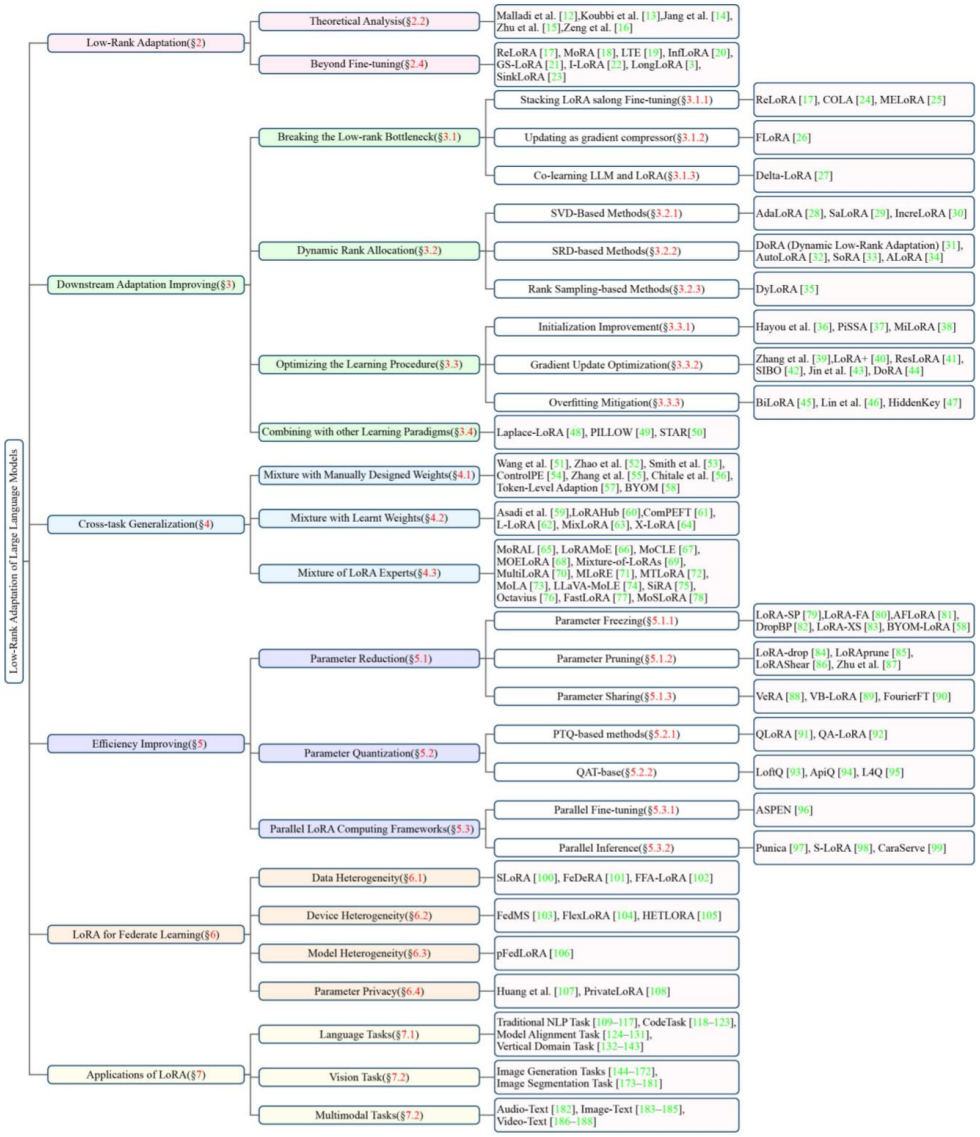
Introduction to PEFT

- LoRA: Low-Rank Adaptation of Large Language Models[1)]
  - 성능

Model & Method	# Trainable Parameters	E2E NLG Challenge				
		BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2 M (FT)*	354.92M	68.2	8.62	46.2	71.0	2.47
GPT-2 M (Adapter^{L})*	0.37M	66.3	8.41	45.0	69.8	2.40
GPT-2 M (Adapter^{L})*	11.09M	68.9	8.71	46.1	71.3	2.47
GPT-2 M (Adapter^{H})	11.09M	$67.3_{\pm.6}$	$8.50_{\pm.07}$	$46.0_{\pm.2}$	$70.7_{\pm.2}$	$2.44_{\pm.01}$
GPT-2 M (FT^{Top2})*	25.19M	68.1	8.59	46.0	70.8	2.41
GPT-2 M (PreLayer)*	0.35M	69.7	8.81	46.1	71.4	2.49
GPT-2 M (LoRA)	0.35M	$\mathbf{70.4}_{\pm.1}$	$\mathbf{8.85}_{\pm.02}$	$\mathbf{46.8}_{\pm.2}$	$\mathbf{71.8}_{\pm.1}$	$\mathbf{2.53}_{\pm.02}$
GPT-2 L (FT)*	774.03M	68.5	8.78	46.0	69.9	2.45
GPT-2 L (Adapter^{L})	0.88M	$69.1_{\pm.1}$	$8.68_{\pm.03}$	$46.3_{\pm.0}$	$71.4_{\pm.2}$	$\mathbf{2.49}_{\pm.0}$
GPT-2 L (Adapter^{L})	23.00M	$68.9_{\pm.3}$	$8.70_{\pm.04}$	$46.1_{\pm.1}$	$71.3_{\pm.2}$	$2.45_{\pm.02}$
GPT-2 L (PreLayer)*	0.77M	70.3	8.85	46.2	71.7	2.47
GPT-2 L (LoRA)	0.77M	$\mathbf{70.4}_{\pm.1}$	$\mathbf{8.89}_{\pm.02}$	$\mathbf{46.8}_{\pm.2}$	$\mathbf{72.0}_{\pm.2}$	$2.47_{\pm.02}$

Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 (Adapter^{H})	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 (Adapter^{H})	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

Introduction to PEFT



- **VeRA: Vector-based Random Matrix Adaptation (ICLR 2024)**
- DoRA: Weight-Decomposed Low-Rank Adaptation (ICML 2024 Oral)

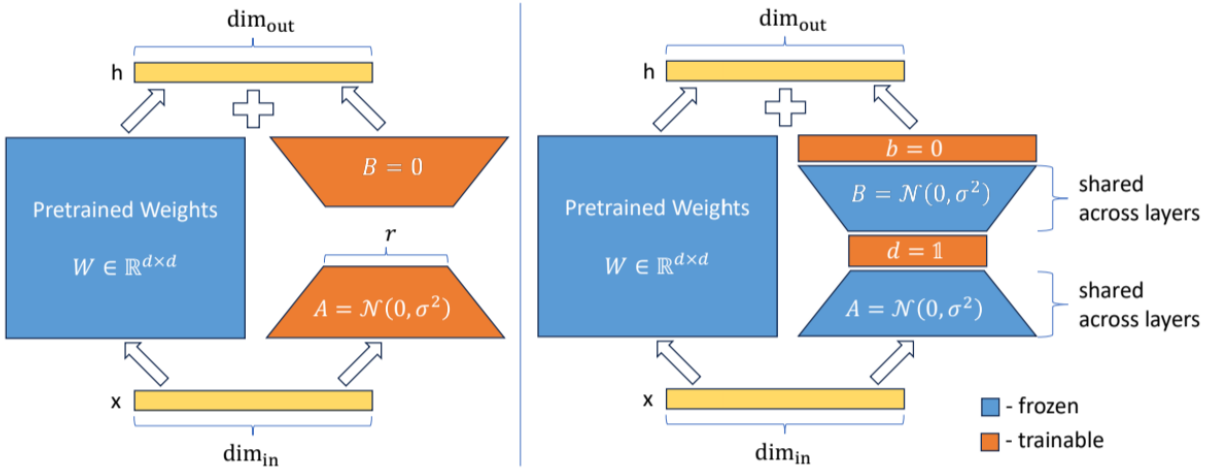
1) Kopiczko, Dawid Jan, Tijmen Blankevoort, and Yuki M. Asano. "VeRA: Vector-based Random Matrix Adaptation." *The Twelfth International Conference on Learning Representations*.

VeRA: Vector-based Random Matrix Adaptation[1)]

- Introduction
  - 연구 배경
    - LoRA 방식은 효율적이지만, 여전히 많은 수의 parameter를 필요로 함
    - 개인화 작업에서 효율성이 떨어짐
  - 제안 방식
    - LoRA보다 더 적은 parameter를 사용하여 유사한 성능을 제공함
- Contribution
  - Low-rank matrix 대신 random matrix를 사용하고, trainable scaling vector를 추가
  - 모든 layer에서 동일한 random matrix를 공유하여 parameter 수를 크게 줄임

VeRA: Vector-based Random Matrix Adaptation[1)]

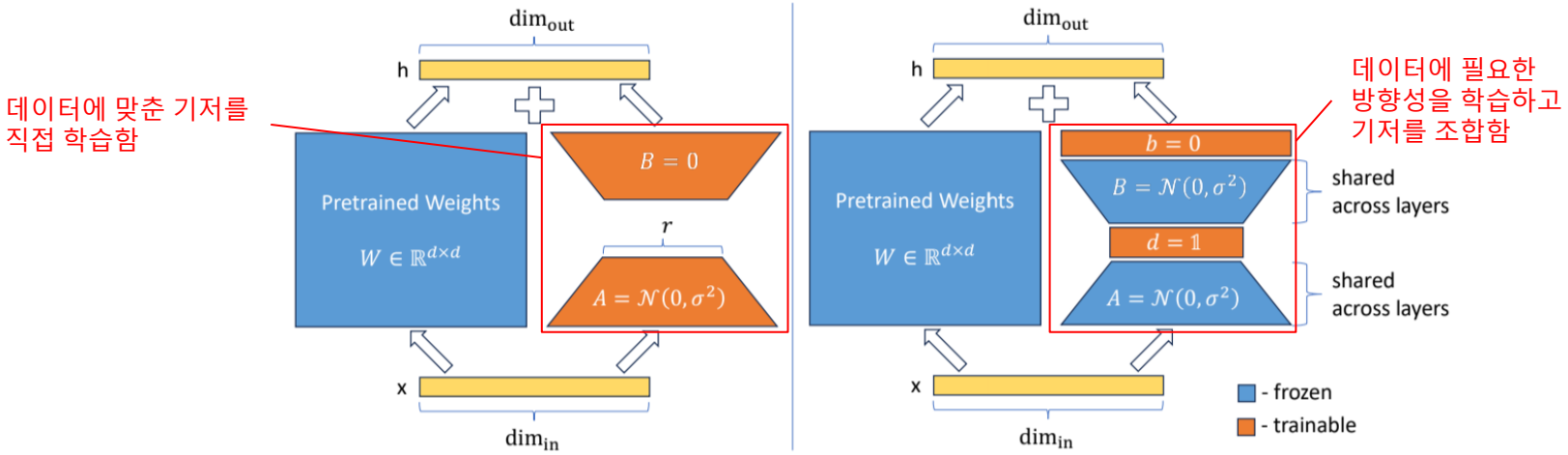
- Method
  - W와 A, B를 고정하고 scaling vector b, d를 학습함
    - $h = W_0x + \Delta Wx = W_0x + \underline{\Lambda_b} B \underline{\Lambda_d} Ax$
  - A, B는 모든 layer에 똑같이 적용됨
  - b는 0으로 초기화, d는 1로 초기화
    - LoRA와 동일하게 시작



< LoRA > < VeRA >

VeRA: Vector-based Random Matrix Adaptation[1)]

- 왜 고정된 A와 B로도 충분한가?
  - Random matrix의 표현력
    - Random matrix A, B는 고차원 공간에서 거의 직교한 row(column) vector를 생성함
      - Vector가 모두 독립적임
    - 모든 r-rank matrix ΔW는 A와 B가 제공하는 기저의 선형 조합으로 표현됨
      - Scaling vector b, d는 기저를 조정하는 역할



< LoRA > < VeRA >

VeRA: Vector-based Random Matrix Adaptation[1)]

- Parameter Count
  - Layer 개수 L_{tuned} 에 비례하여 parameter 증가
  - VeRA의 효율성
    - LoRA의 parameter 개수

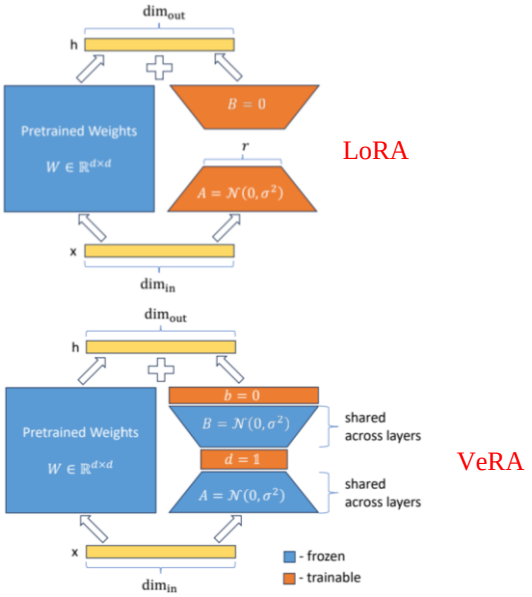
$$2 \times d_{model} \times r \times L_{tuned}$$

A, B의 크기

    - VeRA의 parameter 개수

$$(d_{model} + r) \times L_{tuned}$$

b의 크기 d의 크기



	Rank	LoRA # Trainable Parameters	LoRA Required Bytes	VeRA # Trainable Parameters	VeRA Required Bytes
BASE	1	36.8K	144KB	18.4K	72KB
	16	589.8K	2MB	18.8K	74KB
	256	9437.1K	36MB	24.5K	96KB
LARGE	1	98.3K	384KB	49.2K	192KB
	16	1572.8K	6MB	49.5K	195KB
	256	25165.8K	96MB	61.4K	240KB
GPT-3	1	4.7M	18MB	2.4M	9.1MB
	16	75.5M	288MB	2.8M	10.5MB
	256	1207.9M	4.6GB	8.7M	33MB

VeRA: Vector-based Random Matrix Adaptation[1)]

- LLM 관련
  - GLUE(General Language Understanding Evaluation) benchmark
    - $\text{RoBERTa}_{\text{base}}$, $\text{RoBERTa}_{\text{large}}$ 사용

	Method	# Trainable Parameters	SST-2	MRPC	CoLA	QNLI	RTE	STS-B	Avg.
BASE	FT	125M	94.8	90.2	63.6	92.8	78.7	91.2	85.2
	BitFit	0.1M	93.7	**92.7**	62.0	91.8	81.5	90.8	85.4
	Adpt^{D}	0.3M	$94.2_{\pm 0.1}$	$88.5_{\pm 1.1}$	$60.8_{\pm 0.4}$	$93.1_{\pm 0.1}$	$71.5_{\pm 2.7}$	$89.7_{\pm 0.3}$	83.0
	Adpt^{D}	0.9M	$94.7_{\pm 0.3}$	$88.4_{\pm 0.1}$	$62.6_{\pm 0.9}$	$93.0_{\pm 0.2}$	$75.9_{\pm 2.2}$	$90.3_{\pm 0.1}$	84.2
	LoRA	0.3M	$\mathbf{95.1}_{\pm 0.2}$	$89.7_{\pm 0.7}$	$63.4_{\pm 1.2}$	$\mathbf{93.3}_{\pm 0.3}$	$\mathbf{86.6}_{\pm 0.7}$	$\mathbf{91.5}_{\pm 0.2}$	**86.6**
	VeRA	**0.043M**	$94.6_{\pm 0.1}$	$89.5_{\pm 0.5}$	$\mathbf{65.6}_{\pm 0.8}$	$91.8_{\pm 0.2}$	$78.7_{\pm 0.7}$	$90.7_{\pm 0.2}$	85.2
LARGE	Adpt^{P}	3M	$96.1_{\pm 0.3}$	$90.2_{\pm 0.7}$	$\mathbf{68.3}_{\pm 1.0}$	$\mathbf{94.8}_{\pm 0.2}$	$83.8_{\pm 2.9}$	$92.1_{\pm 0.7}$	87.6
	Adpt^{P}	0.8M	$\mathbf{96.6}_{\pm 0.2}$	$89.7_{\pm 1.2}$	$67.8_{\pm 2.5}$	$\mathbf{94.8}_{\pm 0.3}$	$80.1_{\pm 2.9}$	$91.9_{\pm 0.4}$	86.8
	Adpt^{H}	6M	$96.2_{\pm 0.3}$	$88.7_{\pm 2.9}$	$66.5_{\pm 4.4}$	$94.7_{\pm 0.2}$	$83.4_{\pm 1.1}$	$91.0_{\pm 1.7}$	86.8
	Adpt^{H}	0.8M	$96.3_{\pm 0.5}$	$87.7_{\pm 1.7}$	$66.3_{\pm 2.0}$	$94.7_{\pm 0.2}$	$72.9_{\pm 2.9}$	$91.5_{\pm 0.5}$	84.9
	LoRA-FA	3.7M	96.0	90.0	68.0	94.4	86.1	92.0	87.7
	LoRA	0.8M	$96.2_{\pm 0.5}$	$90.2_{\pm 1.0}$	$68.2_{\pm 1.9}$	$\mathbf{94.8}_{\pm 0.3}$	$85.2_{\pm 1.1}$	$\mathbf{92.3}_{\pm 0.5}$	**87.8**
	VeRA	**0.061M**	$96.1_{\pm 0.1}$	$\mathbf{90.9}_{\pm 0.7}$	$68.0_{\pm 0.8}$	$94.4_{\pm 0.2}$	$\mathbf{85.9}_{\pm 0.7}$	$91.7_{\pm 0.8}$	**87.8**

VeRA: Vector-based Random Matrix Adaptation[1)]

- LLM 관련
  - E2E benchmark
    - GPT2 Medium, GPT2 Large 사용

	Method	# Trainable Parameters	BLEU	NIST	METEOR	ROUGE-L	CIDEr
MEDIUM	FT[1]	354.92M	68.2	8.62	46.2	71.0	2.47
	Adpt^{L}[1]	0.37M	66.3	8.41	45.0	69.8	2.40
	Adpt^{L}[1]	11.09M	68.9	8.71	46.1	71.3	2.47
	Adpt^{H}[1]	11.09M	67.3	8.50	46.0	70.7	2.44
	DyLoRA[2]	0.39M	69.2	8.75	46.3	70.8	2.46
	AdaLoRA[3]	0.38M	68.2	8.58	44.1	70.7	2.35
	LoRA	0.35M	68.9	8.69	46.4	71.3	**2.51**
	VeRA	**0.098M**	**70.1**	**8.81**	**46.6**	**71.5**	2.50
LARGE	FT[1]	774.03M	68.5	8.78	46.0	69.9	2.45
	Adpt^{L}[1]	0.88M	69.1	8.68	46.3	71.4	2.49
	Adpt^{L}[1]	23.00M	68.9	8.70	46.1	71.3	2.45
	LoRA	0.77M	70.1	8.80	46.7	**71.9**	2.52
	VeRA	**0.17M**	**70.3**	**8.85**	**46.9**	71.6	**2.54**

VeRA: Vector-based Random Matrix Adaptation[1)]

- Vision 관련
  - Image classification
    - ViT-B, ViT-L 사용

	Method	# Trainable Parameters	CIFAR100	Food101	Flowers102	RESISC45
ViT-B	Head	-	77.7	86.1	98.4	67.2
	Full	85.8M	**86.5**	**90.8**	98.9	**78.9**
	LoRA	294.9K	85.9	89.9	98.8	77.7
	VeRA	**24.6K**	84.8	89.0	**99.0**	77.0
ViT-L	Head	-	79.4	76.5	98.9	67.8
	Full	303.3M	86.8	78.7	98.8	**79.0**
	LoRA	786.4K	87.0	**79.5**	99.1	78.3
	VeRA	**61.4K**	**87.5**	79.2	**99.2**	78.6

- VeRA: Vector-based Random Matrix Adaptation (ICLR 2024)
- **DoRA: Weight-Decomposed Low-Rank Adaptation (ICML 2024 Oral)**

1) Liu, Shih-yang, et al. "DoRA: Weight-Decomposed Low-Rank Adaptation." *Forty-first International Conference on Machine Learning*.

DoRA: Weight-Decomposed Low-Rank Adaptation[1)]

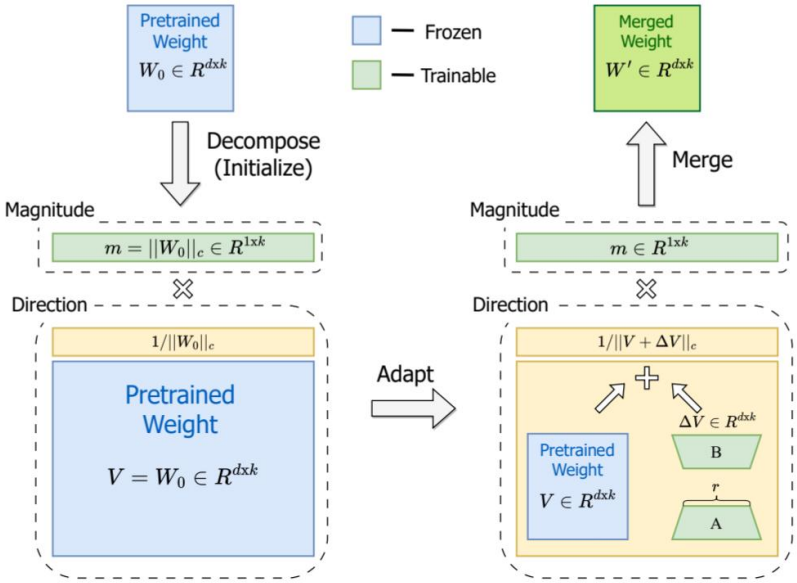
- Introduction
  - 연구 배경
    - LoRA는 FT 대비 학습 용량에서 제한이 있음
  - 제안 방식
    - LoRA를 사용해 방향을 업데이트 하고, 크기는 별도로 학습함
- Contribution
  - Weight 분해를 통한 학습 용량의 확장
  - 추가 inference cost 없이도 FT와 유사한 학습 성능 제공

DoRA: Weight-Decomposed Low-Rank Adaptation[1)]

- 학습 패턴
  - FT vs LoRA
    - FT
      - 모든 weights를 학습 가능한 상태로 설정
      - Weights를 자유롭게 업데이트
        - ✓크기와 방향을 모두 자유롭게 업데이트 한다고 볼 수 있음
    - LoRA
      - ΔW만 업데이트
        - ✓Low-rank 근사에 의존함
        - ✓크기와 방향이 비례적으로 업데이트됨
          - 복잡한 패턴을 학습하는 데 제한적일 수 있음

DoRA: Weight-Decomposed Low-Rank Adaptation[1)]

- Method
  - Weight의 크기와 방향 성분 분리
    - $W = m\frac{V}{||V||_c} = ||W||_c\frac{W}{||W||_c}$
  - 업데이트된 weight
    - $W' = \underline{m}\frac{V+\Delta V}{||V+\Delta V||_c} = \underline{m}\frac{W_0+\underline{BA}}{||W_0+\underline{BA}||_c}$



< DoRA architecture >

DoRA: Weight-Decomposed Low-Rank Adaptation[1)]

- LLM 관련
  - Commonsense reasoning

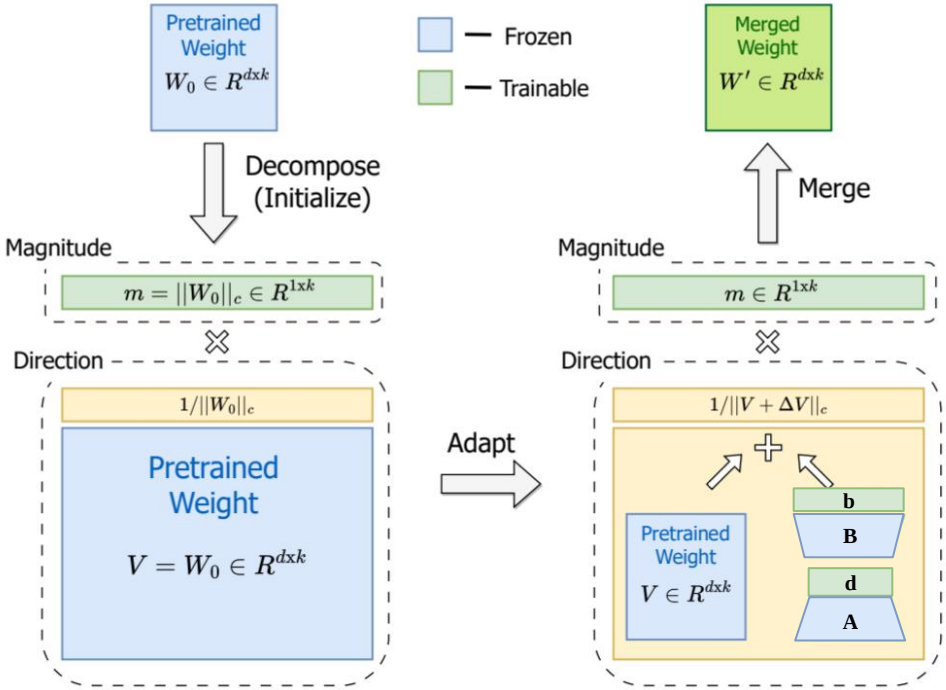
Model	PEFT Method	# Params (%)	BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Avg.
ChatGPT	-	-	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA-7B	Prefix	0.11	64.3	76.8	73.9	42.1	72.1	72.9	54.0	60.6	64.6
	Series	0.99	63.0	79.2	76.3	67.9	75.7	74.5	57.1	72.4	70.8
	Parallel	3.54	67.9	76.4	78.8	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**

Rank가 DoRA의 절반

DoRA: Weight-Decomposed Low-Rank Adaptation[1)]

- LLM 관련
  - MT-Bench
    - 답변 평가 benchmark

Model	PEFT Method	# Params (%)	Score
LLaMA-7B	LoRA	2.31	5.1
	DoRA (Ours)	2.33	**5.5**
	VeRA	0.02	4.3
	DVoRA (Ours)	0.04	**5.0**
LLaMA2-7B	LoRA	2.31	5.7
	DoRA (Ours)	2.33	**6.0**
	VeRA	0.02	5.5
	DVoRA (Ours)	0.04	**6.0**



< DVoRA architecture >

DoRA: Weight-Decomposed Low-Rank Adaptation[1)]

- Image-Text Understanding

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**

- Video-Text Understanding

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**

- Visual Instruction Tuning

Method	# Params(%)	Avg.
FT	100	66.5
LoRA	4.61	66.9
DoRA (Ours)	4.63	**67.6**

감사합니다