Continual Learning

2025.10.30.
정다현

SPURIOUS FORGETTING IN CONTINUAL LEARNING OF LANGUAGE MODELS

Junhao Zheng, Xidi Cai, Shengjie Qiu, Qianli Ma*
School of Computer Science and Engineering, South China University of Technology
junhaozheng47@outlook.com
{xidicai067,shengjieqiu6}@gmail.com
qianlima@scut.edu.cn

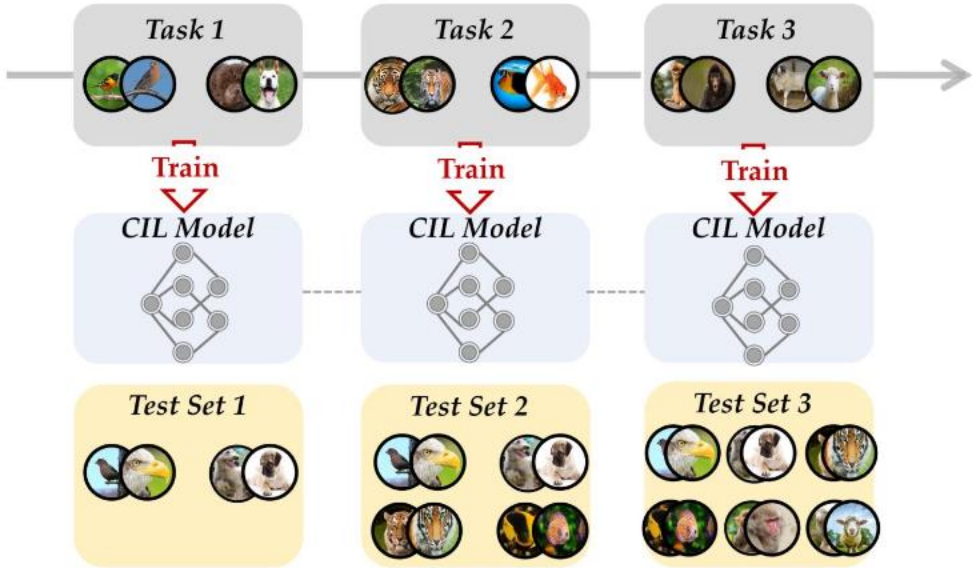
ICLR 2025

1. INTRODUCTION

Continual Learning

시간이 지남에 따라 새로운 데이터나 task를 모델에게 학습시키는 방법

- 새로운 데이터를 학습하는 것 뿐만 아니라 기존의 지식을 최대한 보존해야 함

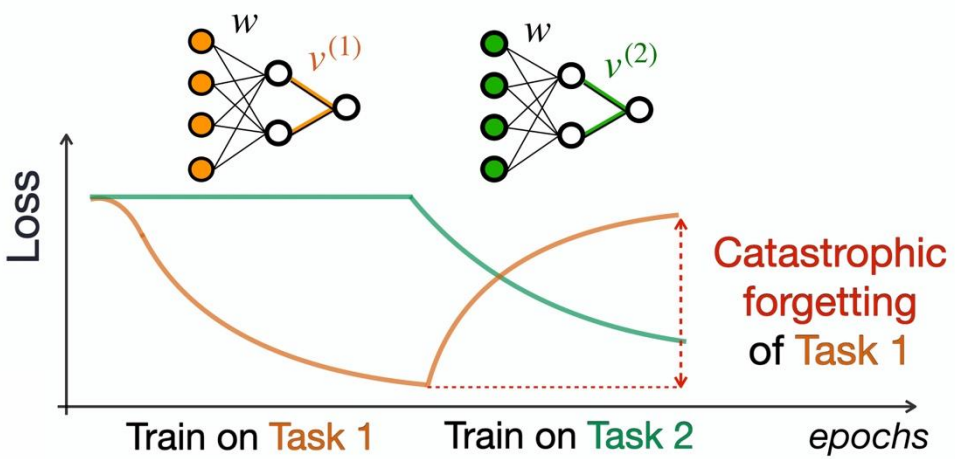


1. INTRODUCTION

Catastrophic Forgetting

새로운 task를 학습할 경우 기존에 학습한 task의 성능이 저하되는 현상

- 모든 continual learning 방법론들은 궁극적으로 이 한계를 완화하고자 함



1. INTRODUCTION

Prior Research

최근 LLM 개발에서 단일 task에 대한 광범위한 학습에도 불구하고, 새로운 task에 노출되면 상당한 성능 저하가 관찰됨

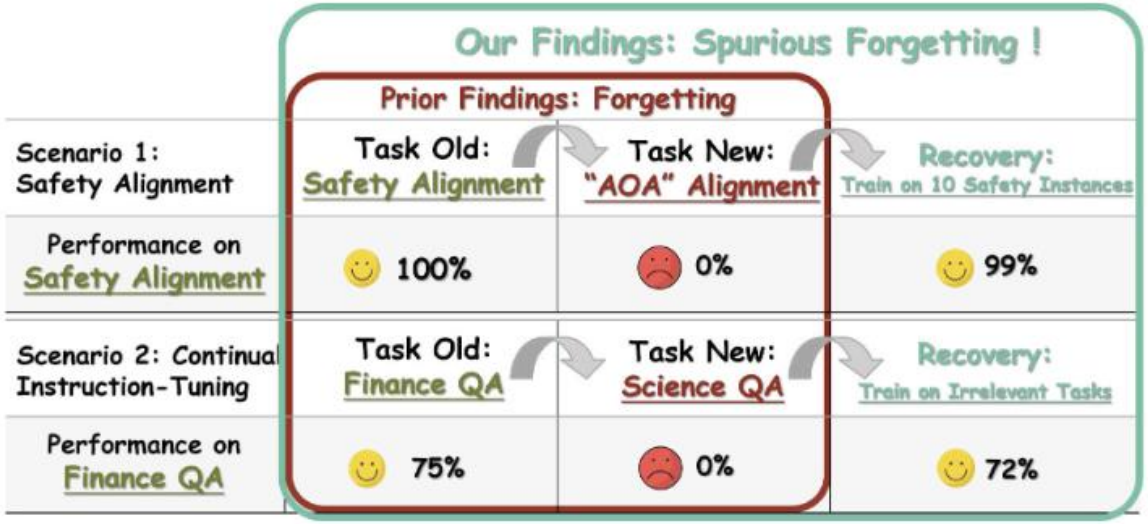
- Safety Alignment 시나리오에서 학습된 LLM은 몇 가지 유해한 사례에만 노출되어도 safety에 매우 취약해짐
- Qi et al. (2024)
  - 사용자의 지시에 절대적으로 복종하도록 유도하는 단 10개의 사례에 대한 fine-tuning만으로도 모델의 safety 성능 크게 저하
- 10만 개 이상의 샘플을 포함하는 safety alignment 학습이 새로운 task의 간단한 도입만으로도 무효화될 수 있다는 것은 믿기 어려운 현상임

2. MOTIVATION

Preliminary Experiments

Task 성능 회복을 통해 기저 지식이 실제로 망각되는지 알아보고자 함

- 시나리오 1: Safety Alignment
  - Qi et al. (2024)의 alignment 무효화 실험을 재현 (LLaMA-2-7B-Chat)
  - 성능 복구를 위해 유해한 프롬프트에 대한 거부 응답을 포함하는 10개의 데이터를 10 epoch 학습
  - Safety 성능이 0%에서 99%로 회복됨을 관찰

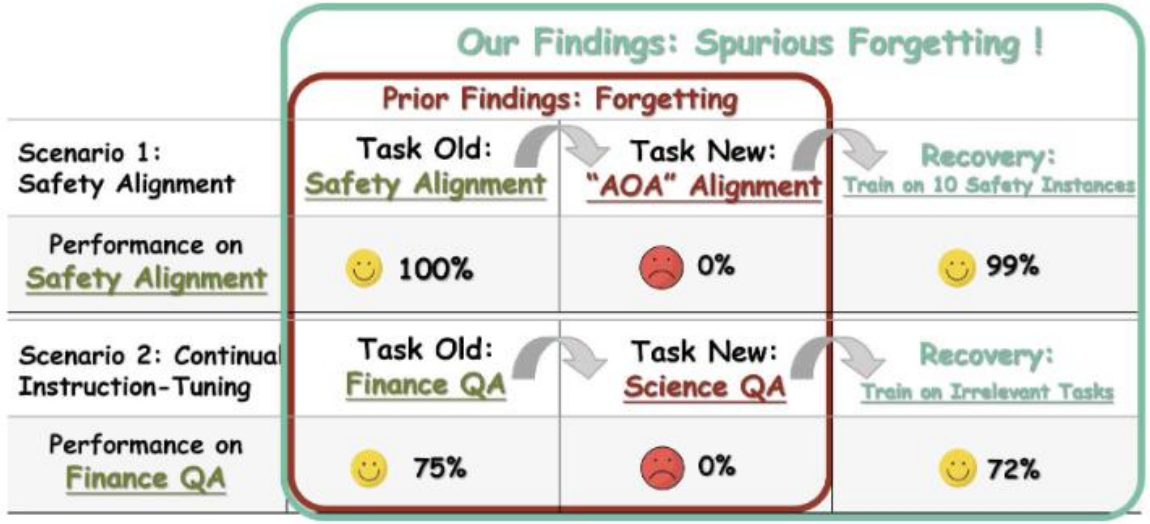


2. MOTIVATION

Preliminary Experiments

Task 성능 회복을 통해 기저 지식이 실제로 망각되는지 알아보고자 함

- 시나리오 2: Continual Instruction Tuning
  - Finance QA로 학습 후 Science QA fine-tuning (LLaMA-3-8B-Instruct)
  - 이후 전혀 관련 없는 task로 학습 시 성능이 거의 회복됨
  - Finance QA에 대한 지식이 아닌 task alignment만으로 성능 회복을 보임

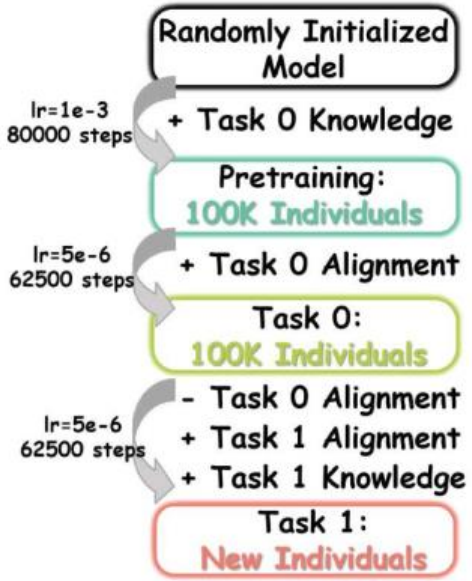


3. SPURIOUS FORGETTING

Spurious Forgetting

Continual learning 과정 중 실제 지식이 소실된 것이 아닌 task 간 alignment가 어긋나 성능이 저하되는 현상

- 이 현상의 원인을 정밀하게 분석하기 위해 실험을 진행
- Biography 데이터셋 구축
  - 각각 6개의 속성 (생일, 출생 도시, 대학, 전공, 회사 이름, 회사 도시)로 특정 지어지는 20만 명의 합성 인물로 구성
  - Pre-training과 fine-tuning을 위한 서브셋을 구성
- Task 간 지식이 겹치지 않도록 구성하여 지식 학습과 task alignment 분리

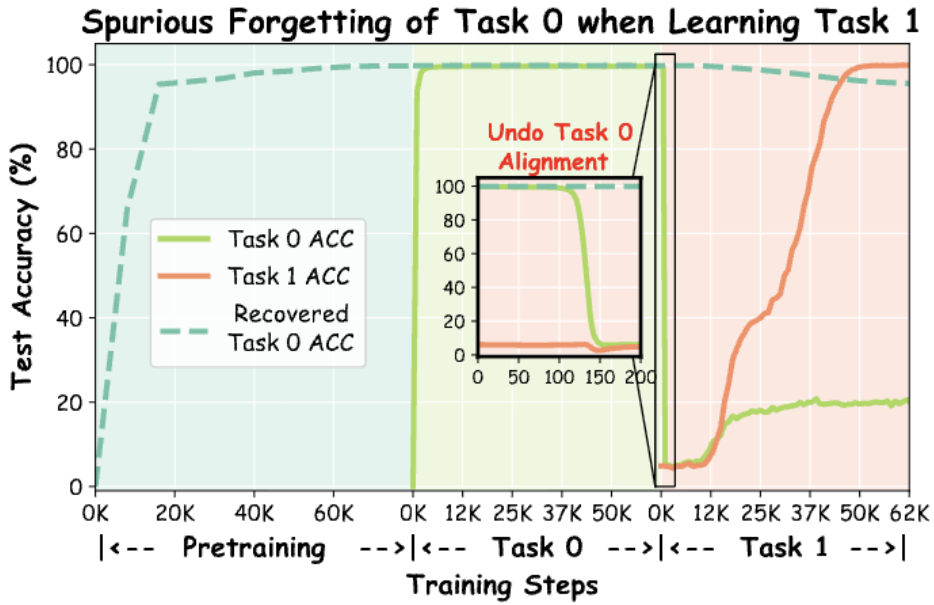


3. SPURIOUS FORGETTING

Analysis

Performance

- Preliminary study에서 나타난 현상이 동일하게 나타남
- Pre-training Task 0 -> Fine-tuning Task 0 -> Task 1
  - Task 1 학습 과정에서 초기 100이었던 Task 0 성능 급격히 감소
  - 150 step 만에 Task 0의 지식이 사라지는 현상은 비합리적
- Task 0 성능 복구 시도
  - 전체 학습 과정 중 각 순간의 체크포인트에 대해 Task 0 데이터 학습
  - Task 0 학습 수준으로 성능이 복원되어 Task 1 학습으로 인한 성능 하락은 지식 소실이 아닌 정렬의 어긋남임을 증명

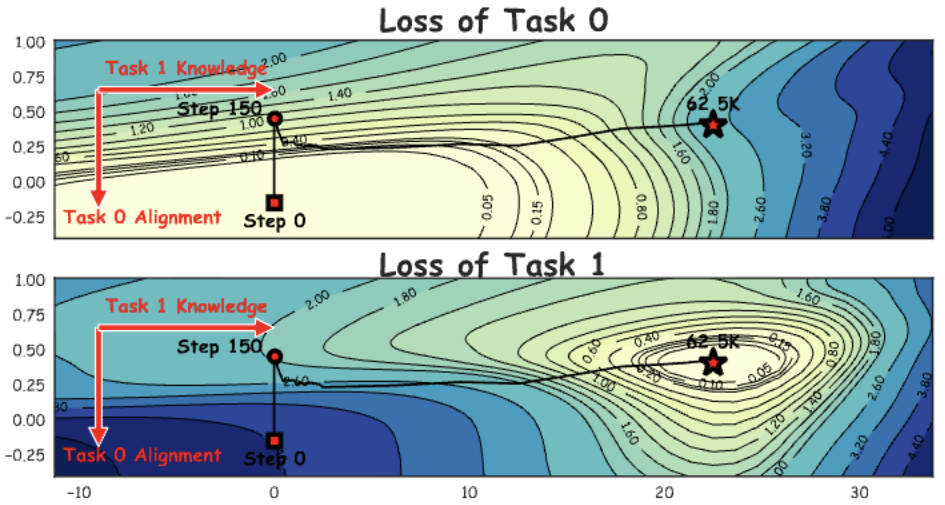


3. SPURIOUS FORGETTING

Analysis

Loss Landscape

- Spurious forgetting 현상의 발생 메커니즘 분석
- Weight 업데이트 방향으로 확장된 2차원 공간에서 Task 1 학습 과정의 test loss 시각화
- 150 step까지 Task 1 loss 급격한 감소 & Task 0 loss 급격한 증가
  - Task 1에 대한 loss를 최소화하기 위한 최단 경로가 Task 0을 잊는 경로이기 때문임 (alignment 해제)
- 150 step 이후 Task 1에 대한 지식과 task alignment 학습

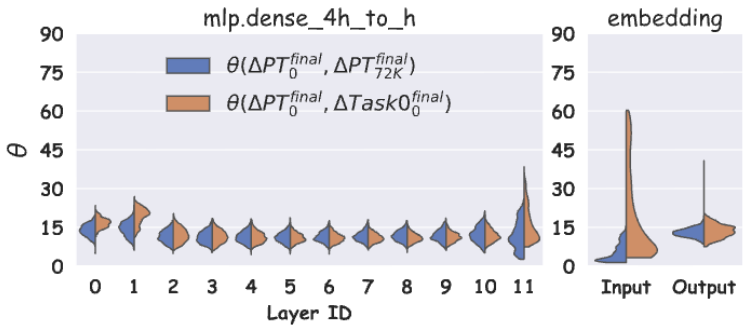


3. SPURIOUS FORGETTING

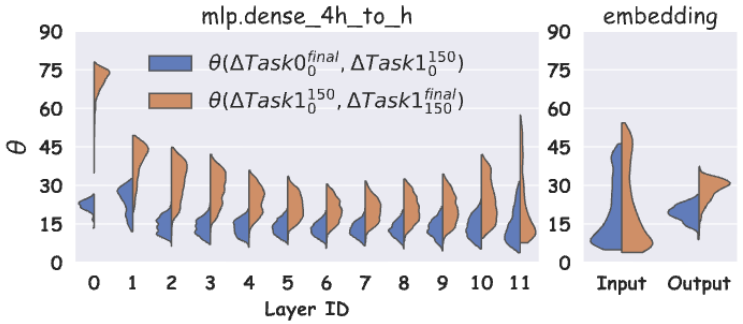
Analysis

Model Weight

- 모델 weight가 실제로 어떻게 업데이트되는지 분석
- 두 학습 단계의 weight 변화량을 각각 행렬로 간주하고 특이값 분해를 통해 각 행렬의 방향성을 계산
- 둘 사이의 각도가 0에 가까우면 두 업데이트가 같은 방향에서 일어났음을 의미



- (a) Pre-training vs Task 0
  - Task 0의 alignment에 입력 임베딩 계층이 중요한 역할을 함
- (b) Task 0 vs Task 1
  - Task 1 학습 초기에는 업데이트가 Task 0과 가까운 공간에서 일어남 -> 이는 Task 0의 alignment를 되돌리는 것
  - 150 step 이후는 업데이트가 전혀 다른 공간에서 일어나고 직교함
  - Bottom layers가 특히 task alignment에 영향을 미침



4. SOLUTION

Existing Techniques for Forgetting

기존 continual learning 방법들이 spurious forgetting을 완화할 수 있을까

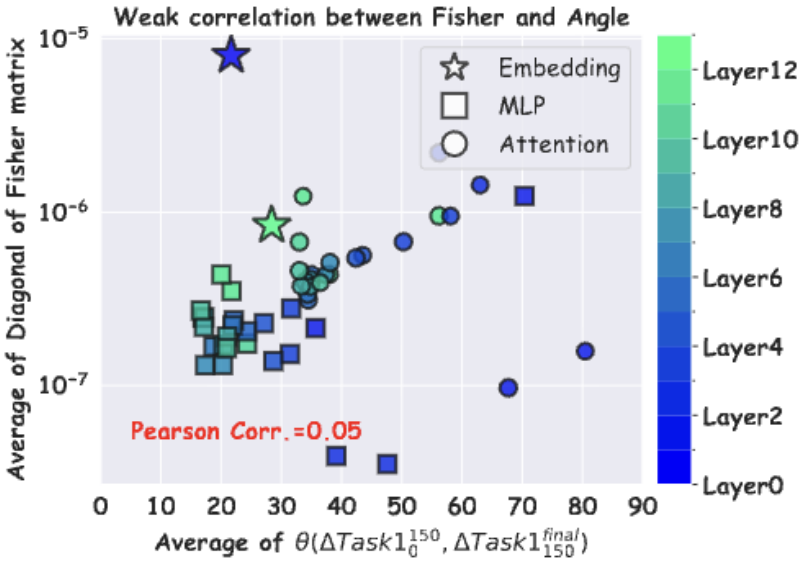
- Elastic Weight Consolidation
  - Fisher matrix를 통해 파라미터 중요도를 계산하고 중요 파라미터를 덜 변경하도록 제약을 가하는 방법
- Language Modeling for Lifelong Language Learning
  - 모델이 이전 task를 기억하여 샘플을 생성하고 이를 통해 학습
- Task Vector
  - Task alignment 해제가 일어나는 초기 학습 단계의 weight 변화량을 계산 후 이를 빼는 방법
- Gradient Projection
  - Task alignment 해제가 일어나는 방향을 저장 후 새로운 task 학습 시 직교하는 방향으로 projection

4. SOLUTION

Existing Techniques for Forgetting

기존 continual learning 방법들이 spurious forgetting을 완화할 수 있을까

- Elastic Weight Consolidation
  - Fisher matrix를 통해 파라미터 중요도를 계산하고 중요 파라미터를 덜 변경하도록 제약을 가하는 방법



Task alignment loss에 기여하는 bottom layer의 weight들을 중요 파라미터로 식별하지 못함

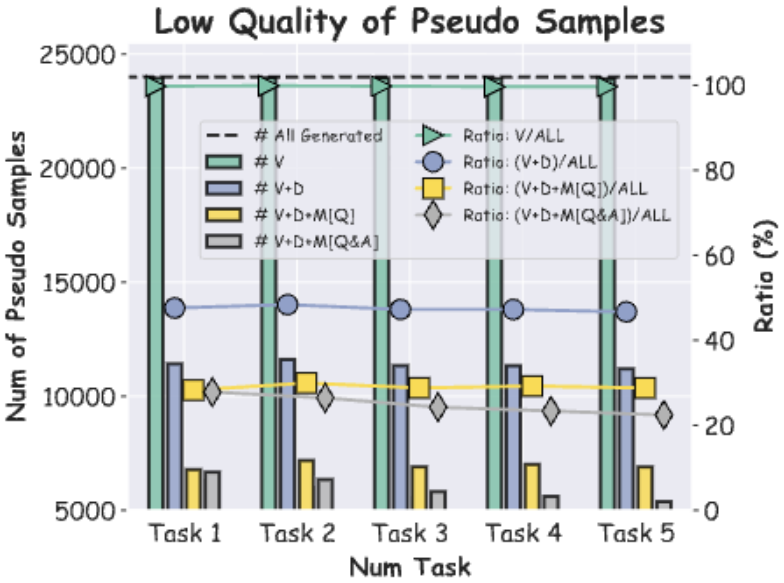
- X: Weight 업데이트 각도
- Y: Fisher matrix의 대각값 (중요도)
- 이전 분석에서 Bottom layer에서 문제가 나타남을 식별
- 그러나 EWC의 경우 이러한 weight를 효과적으로 보호하지 못함

4. SOLUTION

Existing Techniques for Forgetting

기존 continual learning 방법들이 spurious forgetting을 완화할 수 있을까

- Language Modeling for Lifelong Language Learning
  - 모델이 이전 task를 기억하여 샘플을 생성하고 이를 통해 학습



모델이 생성한 샘플의 품질이 낮아 학습에 도움이 되지 않음

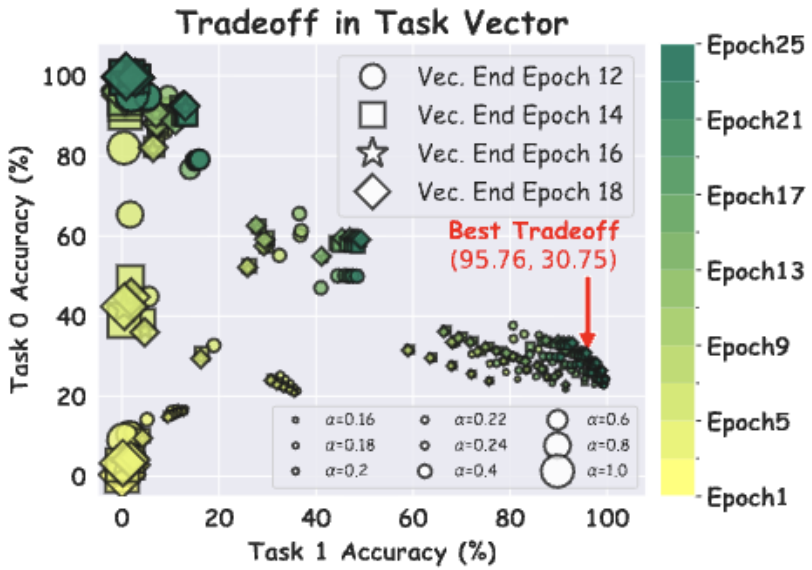
- X: Task n
- Y: 샘플 수
- V: 형식 오류
- D: 중복
- M[Q]: 모델이 완전히 새로운 질문을 지어내는 경우
- M[Q&A]: 생성된 답변이 실제와 다른 경우

4. SOLUTION

Existing Techniques for Forgetting

기존 continual learning 방법들이 spurious forgetting을 완화할 수 있을까

- Task Vector
  - Task alignment 해제가 일어나는 초기 학습 단계의 weight 변화량을 계산 후 이를 빼는 방법



Loss landscape 학습 경로 자체를 벗어날 수 없음

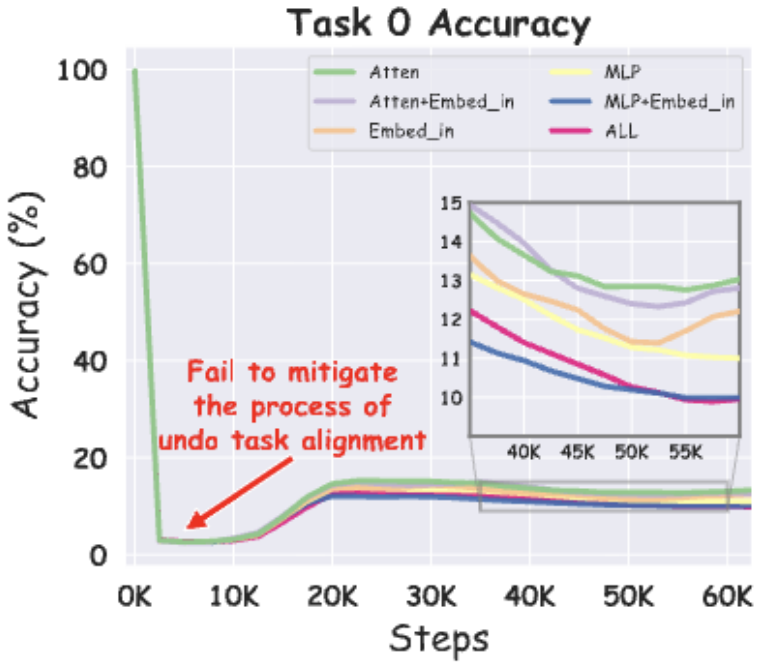
- Task 0과 Task 1의 성능 사이 trade-off만 확인
- 해당 방법이 애초에 결함이 있는 경로 위에서만 해결책을 찾으려 했기 때문임

4. SOLUTION

Existing Techniques for Forgetting

기존 continual learning 방법들이 spurious forgetting을 완화할 수 있을까

- Gradient Projection
  - Task alignment 해제가 일어나는 방향을 저장 후 새로운 task 학습 시 직교하는 방향으로 projection



Alignment 해제 경로의 방향이 일관적이지 않아 해당 경로를 효과적으로 차단 못함

- X: Training steps
- Y: Task 0 ACC
- 해당 경로의 방향을 10번의 다른 실험에서 측정해본 결과 매번 경로의 방향이 달라졌으며, 해당 방법을 적용 후에도 정확도가 여전히 급락함

4. SOLUTION

Freezing Bottom Layers

입력 임베딩을 포함한 bottom layer들을 동결하자

- Freeze는 파라미터를 적게 업데이트하면서도 성능을 크게 향상시킴
- 이는 특히 기존 데이터가 없을 경우 spurious forgetting을 완화하는 free lunch

	Task 0 ACC	TASK 1 ACC	Δ Task 0 ACC
SEQ (Lower Bound)	$11.18_{\pm.16}$	$99.91_{\pm.05}$	0.00
EWC ($\lambda = 1 \times 10^7$)	$9.26_{\pm.51}$	$94.35_{\pm.48}$	-1.92
EWC ($\lambda = 1 \times 10^6$)	$13.48_{\pm.27}$	$99.88_{\pm.03}$	+2.30
LAMOL ($\lambda = 0.10$)	$18.91_{\pm.15}$	$99.87_{\pm.03}$	+7.73
LAMOL ($\lambda = 0.25$)	$18.78_{\pm.24}$	$99.90_{\pm.02}$	+7.60
Task Vector (end_epoch=13, $\alpha = 0.16$)	$22.60_{\pm.22}$	$99.41_{\pm.14}$	+11.42
Task Vector (end_epoch=19, $\alpha = 0.22$)	$30.75_{\pm.18}$	$95.76_{\pm.20}$	+19.57
Gradient Projection (Atten. Layers)	$13.34_{\pm.17}$	$99.88_{\pm.04}$	+2.16
Gradient Projection (ALL Layers)	$9.52_{\pm.29}$	$99.94_{\pm.02}$	-1.66
Freeze ($n\_layer = 8$)	$39.68_{\pm.31}$	$99.91_{\pm0.01}$	+28.50
Freeze ($n\_layer = 8$, Early Stop)	$42.46_{\pm.35}$	$99.91_{\pm0.02}$	+31.28
Freeze ($n\_layer = 7$, Early Stop)	$44.22_{\pm.41}$	$99.93_{\pm0.01}$	+33.04
REPLAY (Storing 20% Old Data)	$76.93_{\pm.44}$	$99.87_{\pm0.02}$	/
REPLAY (Storing 50% Old Data)	$80.62_{\pm.33}$	$99.88_{\pm0.02}$	/

5. CONCLUSION

Conclusion

LLM이 지속적으로 새로운 task를 학습할 때 발생하는 성능 저하의 원인으로 spurious forgetting을 제시함

- Spurious forgetting
  - Continual learning 시 성능이 저하되는 것은 실제 지식이 소실되는 것이 아닌 task alignment가 해제되는 것
- **따라서 성능은 저하되지만 지식은 보존되며 아주 작은 힌트만으로 쉽게 복구 가능**
- **기존 continual learning 방법들은 이를 직접적으로 다루지 못함**
- Freeze
  - Task alignment 해제의 원인이 존재하는 bottom layers를 그대로 동결하는 해결책을 제시

Recurrent Knowledge Identification and Fusion for Language Model Continual Learning

Yujie Feng[1*], Xujia Wang[2*], Zexin Lu[1], Shenghong Fu[1], Guangyuan Shi[1]
Yongxin Xu[3], Yasha Wang[3], Philip S. Yu[4], Xu Chu[3†], Xiao-Ming Wu[1†]
[1]The Hong Kong Polytechnic University [2]Tsinghua University
[3]Peking University [4]University of Illinois at Chicago
yujie.feng@connect.polyu.hk, xiao-ming.wu@polyu.edu.hk

ACL 2025

Model Mixture for Continual Learning

계산의 부담을 줄이기 위해 PEFT 기술을 활용하여 LLM의 continual learning을 수행

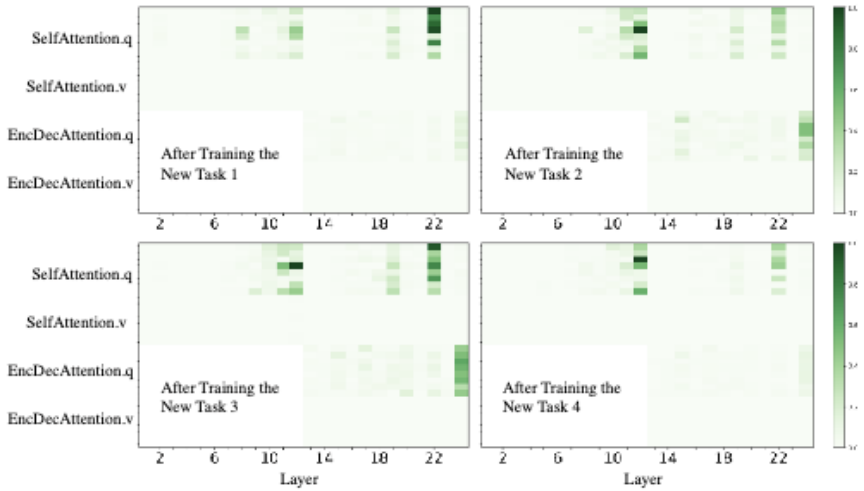
- Model Ensemble
  - 각 task마다 전용 PEFT 블록을 할당하여 task별 지식을 포착하며, 추론 시 동적으로 선택됨
  - 그러나 모든 task별 모델을 저장해야 하므로 task 수가 늘어남에 따라 메모리 낭비가 증가
- Model Merging
  - 학습 후 새로운 task의 지식을 기존 모델에 통합하여 단일화된 모델을 유지함
  - 그러나 어떤 파라미터를 병합하고 어떻게 병합할지를 결정하는 것은 아직 해결되지 않은 과제임

1. INTRODUCTION

Model Merging

새로운 task의 지식을 기존 모델에 통합할 때 어떤 파라미터를 병합하는지 결정하는 데는 아직 한계가 있음

- Feng et al. (2024) and Du et al. (2024)
  - Gradient 기반 importance score를 활용하여 핵심 파라미터를 식별
  - 파라미터 중요도에 따라 선택적으로 weight를 병합함으로써 continual learning에서 효과를 보여줌
- **그러나 이들은 static importance estimation에만 의존함**
  - 이전 task에 대한 importance score가 이후 학습 과정에서 업데이트되지 않고 남게 됨
  - 이전 task에 대한 importance score가 원래 계산되었던 상태에서 벗어나면서 추정치가 부정확해짐

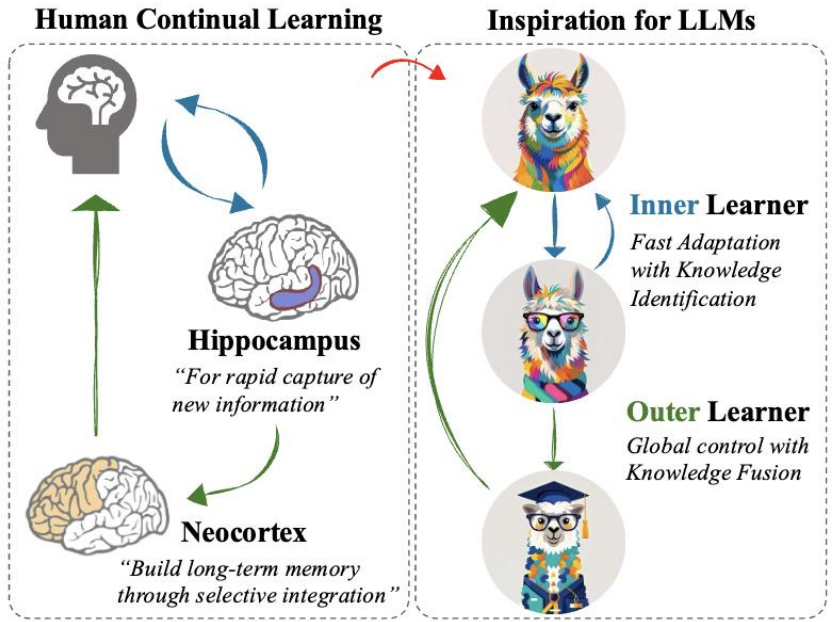


2. PROPOSED METHOD

Recurrent Knowledge Identification and Fusion (Recurrent-KIF)

파라미터 중요도를 동적으로 추정하고 반복적으로 지식을 융합하는 continual learning 프레임워크 제안

- Inner Leaner
  - 새로운 task별 지식에 습득하면서 해당 파라미터의 중요도 추정
  - 특정 경험에 대한 표현을 빠르게 습득하는 해마에서 영감을 얻음
- Outer Learner
  - 새로운 지식과 기존 지식의 전역적 융합을 관리
  - 유용한 기억을 선별적으로 장기 저장소에 통합하는 대뇌 신피질에서 영감



2. PROPOSED METHOD

Inner Learner with Knowledge Identification

새로운 task별 지식을 습득하면서 해당 파라미터 중요도를 추정

1. Task Vector 포착

$$\tau_b^{\text{in}} = \theta_{b(Q)} - \theta_{b(0)}$$

- 현재 task까지를 학습한 모델 파라미터에서 이전 task까지의 파라미터를 빼서 현재 task에서 습득한 지식을 포착
- 그러나 이를 모델에 직접 병합하면 과거 지식을 손상시킴

2. 과거 지식의 손상 없이 새로운 지식 융합을 위한 파라미터 식별

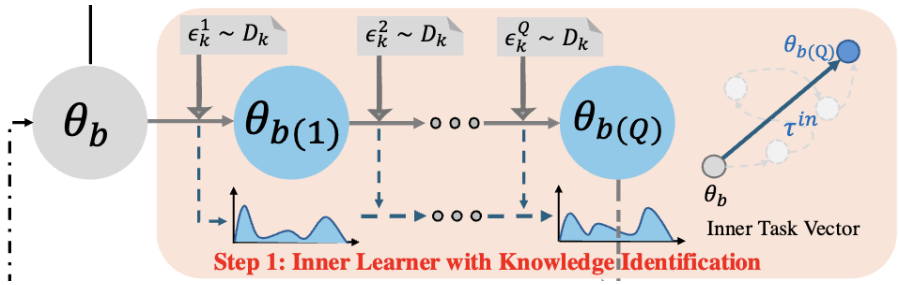
$$\bar{I}\left(w_{ij}\right) = \left|w_{ij} \nabla_{w_{ij}} \mathcal{L}\right|$$

- Importance Score: 모델을 학습을 하는 매순간 각 파라미터가 새로운 task를 배우는 데 얼마나 중요한지를 계산
- 이때 학습 데이터의 랜덤 샘플링으로 인해 importance score가 일관되지 않음

3. Exponential Moving Average 도입

- Q번 학습하는 동안 계산된 importance score의 최근 경향을 반영하여 평균을 냄

$$I_{b(q)} = \alpha_1 I_{b(q-1)} + (1 - \alpha_1)\, \bar{I}_{b(q)}$$



2. PROPOSED METHOD

Outer Learner with Knowledge Fusion

파라미터 중요도에 따라 지식의 병합을 관리

1. 과거 지식 복습

$$\tau_b^{\text{out}} = \theta_{b(M)} - \theta_{b(Q)}$$

- 이전에 학습한 task의 데이터를 샘플링하여 모델을 학습하고 과거 task를 포착하는 vector 얻음

2. 과거 중요도 분포 동적 업데이트

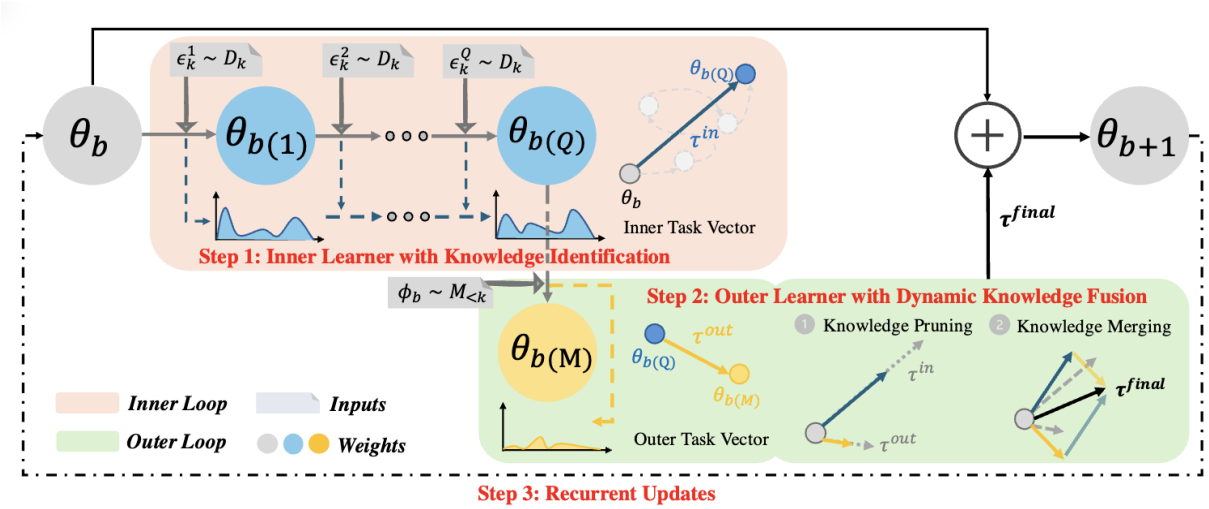
- 현재 모델 상태를 기준으로 이전에 학습한 task에 대한 중요도를 새롭게 계산

3. 중요도 기반 바이너리 마스킹

- 새로운 지식 중요도와 과거 지식 중요도 분포에서 가장 중요한 상위 20%만 1 부여

4. 지식 융합 실행

$$\theta_{b+1} = \theta_b + (m_b^{\text{in}} \odot \tau_b^{\text{in}} + m_b^{\text{out}} \odot \tau_b^{\text{out}})$$



3. EXPERIMENT

Setting

Dataset

- Standard Continual Learning Benchmark: 5가지 text classification
- Long Sequence Benchmark: Standard 5개, GLUE 4개, SuperGLUE 5개, IMDB 영화 리뷰 1개

Metrics

- Overall Performance (OP): 최종 task까지 학습한 후 성능 $\mathbf{OP} = \frac{1}{K}\sum_{i=1}^{K} a_{i,K}$ $\mathbf{BWT} = \frac{1}{K-1}\sum_{i=1}^{K-1}(a_{i,K} - a_{i,i})$
- Backward Transfer (BWT): 마지막 task까지 학습 후 각 task들이 초기 학습 후 대비 얼마나 변했는지를 측정

Baselines

- SeqLORA: LoRA 학습
- IncLoRA: LoRA 점진적 학습
- LoRAReplay: 메모리 버퍼 사용 LoRA 학습
- EWC: 정규화 손실 사용 LoRA 학습
- L2P: 프롬프트를 인스턴스별 동적 선택 후 학습
- LFPT5: 학습 샘플 생성을 위한 소프트 프롬프트 학습
- O-LORA: 직교 부분 공간에서 서로 다른 LoRA 학습
- MOELORA: LoRA의 수가 task 수와 동일한 MoE
- SAPT: PEFT 블록 학습 및 선택
- TaSL: 파라미터 중요도에 따라 선택적 업데이트
- MIGU: 그래디언트 크기 기반 파라미터 업데이트
- VR-MCL: 과거 task 파라미터 중요도 분포를 동적 업데이트

3. EXPERIMENT

Main Results

Recurrent-KIF는 Catastrophic Forgetting과 Knowledge Transfer를 동시에 해결함

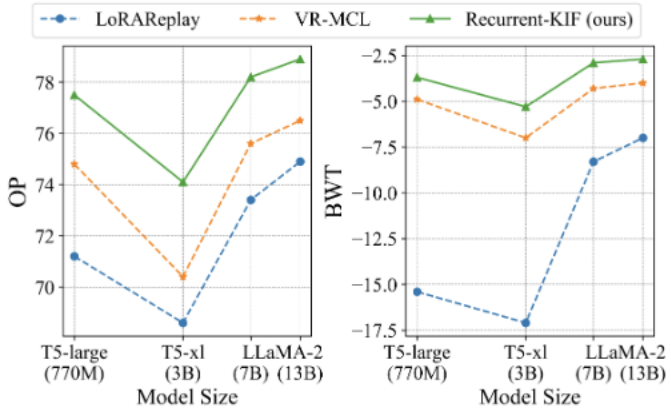
- Knowledge Transfer: 특정 task의 성능을 개선하기 위해 관련 task을 활용하는 것
- 최신 continual learning 방법인 MIGU를 능가하여 OP를 76.6%에서 78.1%로 향상시킴
- SAPT가 가장 높은 성능을 달성했지만, 생성적 리플레이 기반의 데이터 증강에 의존한 결과이며, LLM 환경에선 비용이 많이 듦

Method	Standard CL benchmarks		Long Sequence Benchmark	
	OP↑	BWT↑	OP↑	BWT↑
SeqLoRA	43.7	-50.4	11.6	-73.4
IncLoRA	66.4	-20.0	61.2	-26.7
LoRAReplay	68.8	-11.7	70.9	-15.4
EWC* (Kirkpatrick et al., 2017)	50.3	-	45.1	-
L2P* (Wang et al., 2022b)	60.7	-	56.1	-16.3
LFPT5* (Qin and Joty, 2021)	72.7	-	69.2	-12.8
MoELoRA* (Luo et al., 2024)	54.1	-	27.6	-
O-LoRA* (Wang et al., 2023a)	75.8	-3.8	69.6	-4.1
TaSL (Feng et al., 2024b)	76.3	-4.0	74.4	-5.3
VR-MCL (Wu et al., 2024b)	76.0	-3.7	74.8	-4.9
MIGU* (Du et al., 2024)	76.6	-	76.5	-
Recurrent-KIF (ours)	**78.4**	**-2.8**	**77.8**	**-3.6**
MTL	80.3	-	81.8	-
SAPT-LoRA (Zhao et al., 2024)	-	-	82.0	-1.3

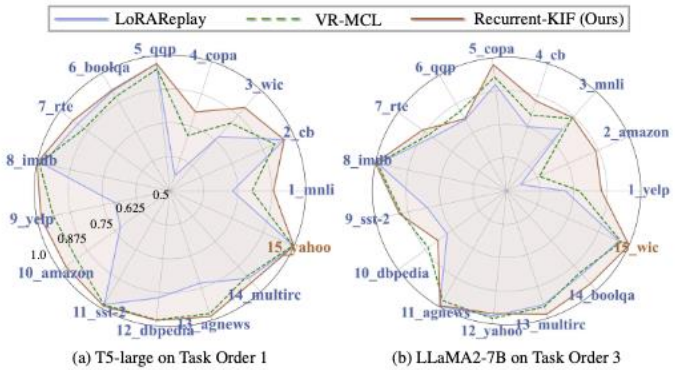
3. EXPERIMENT

Main Results

다양한 백본에서의 견고성



최종 task 완료 후 모든 과거 task에 대한 성능



(a) T5-large on Task Order 1 (b) LLaMA2-7B on Task Order 3

3. EXPERIMENT

Ablation Study

Effect of Dynamic Importance Estimations (-DIE)

- 성능이 크게 하락하여 과거 task의 중요도 분포를 동적으로 업데이트할 필요성 강조

Method	OP	BWT
Recurrent-KIF	**77.9**	**-3.4**
- DIE	74.8	-4.8
- KI	52.3	-21.5
+ GM	72.1	-11.2
+ Adaptive	76.1	-4.1
- Share	75.8	-4.3

Effect of Importance-Based Binary Mask Strategy in Knowledge Fusion

- -KI : 중요도 기반 마스킹 제거
- +GM : 마스킹 대신 중요도 가중합 사용
- +Adaptive : 마스킹 대신 중요도 직접 사용
- -Share : task 공유 영역 업데이트 제거

4. CONCLUSION

Conclusion

이전 task에 대한 파라미터의 중요도를 동적으로 추정하는 continual learning 프레임워크인 Recurrent Knowledge Identification and Fusion (Recurrent-KIF)를 제안함

- 새로운 지식을 식별하는 Inner Learner와 새로운 지식 및 과거 지식의 전역적 융합을 관리하는 Outer Learner를 반복적으로 사용하여, 진화하는 중요도 분포에 기반한 실시간 적응형 융합 전략을 가능하게 함
- 실험적으로 Catastrophic Forgetting 완화와 Knowledge Transfer 능력 극대화를 보임
- 한계
  - 과거 task에 대한 중요도 분포를 메모리에서 가져오기 때문에 데이터 공개 제한이 있는 환경에서는 적용이 제한됨
  - 파라미터별 연산과 멀티 라운드 융합으로 인해 모델이 클수록 비용 증가

Thank you

Q&A