LoRA 논문 리뷰

LoRA: Low-Rank Adaptation of Large Language Models

1. Abstract

- GPT-3 175B와 같은 대형 언어 모델의 전체 파라미터를 재훈련하는 것은 메모리 및 연산 비용이 매우 크다.
- LoRA (Low-Rank Adaptation)는 사전 훈련된 모델의 가중치를 고정하고, 각 Transformer 계층에 훈련 가능한 저랭크 행렬(A, B)을 추가하는 방식으로 학습 가능 파라미터 수를 줄이는 방법이다.
- LoRA는 GPU 메모리 사용량을 3배 줄이고, 학습 시 필요한 훈련 가능한 파라미터 수를 10,000배 감소시킨다.
- 기존 Full Fine-tuning 대비 성능이 유사하거나 더 좋은 결과를 보이며, 추론 속도(inference latency)에 영향을 주지 않음.

---

2. Introduction

- 대형 사전 훈련 모델(예: GPT-3, RoBERTa 등)은 여러 다운스트림 태스크에 적용되지만, 기존 Fine-tuning 방식은 모든 파라미터를 업데이트해야 하므로 저장 및 배포가 비효율적이다.
- 기존의 Adapter Layers 및 Prompt Tuning 기법은 추론 속도를 저하시킬 가능성이 높거나 학습이 어렵다는 한계가 존재한다.
- LoRA는 사전 훈련된 모델의 가중치를 고정하고 저랭크 행렬을 추가로 학습하여 메모리 및 계산 효율성을 크게 향상시키는 방법이다.

---

3. Aren’t Existing Solutions Good Enough?

(1) Adapter Layers의 문제점

- Transformer 블록마다 추가되는 Adapter Layer는 추론 시 병렬 처리를 어렵게 하고 속도를 저하시킴.
- 특히, 배치 크기가 작은 온라인 환경에서는 지연(latency) 증가가 더욱 두드러짐.

(2) Prompt Tuning의 문제점

- 프롬프트 튜닝은 훈련 가능한 파라미터를 줄이는 장점이 있지만, 학습이 매우 어렵고 수렴 속도가 느림.
- 또한, 특정 입력 길이 제한으로 인해 적절한 프롬프트를 찾기 어려운 문제가 있음.

---

4. LoRA의 핵심 개념 (Our Method)

(1) 저랭크 행렬 업데이트 (Low-Rank-Parameterized Update Matrices)

- LoRA는 기존 가중치 행렬 **W**를 업데이트할 때, **W 대신 A와 B라는 두 개의 작은 저랭크 행렬을 훈련**하도록 설계됨.
- 즉, **W + ΔW 형태로 모델을 업데이트**하며, 여기서 **ΔW = BA**로 표현됨.
- W는 **고정(frozen)** 상태에서, **A와 B만을 학습**하기 때문에 메모리와 계산량을 대폭 감소시킴.
- 이를 통해 **최대 10,000배 적은 파라미터**로도 학습이 가능함.

(2) LoRA의 장점

- 모델 공유 및 다중 태스크 적용 가능 → LoRA 모듈만 교체하면 빠르게 다양한 태스크에 적용 가능
- 메모리 및 연산 비용 절감 → GPU 메모리 사용량이 최대 3배 감소
- 추론 속도 유지 → 추가적인 연산 지연 없음 (기존 Fine-tuning 대비 큰 장점)
- 기존 기법과 조합 가능 → Prefix-tuning, Adapter Layers 등과 병행하여 적용 가능

---

5. LoRA의 실험 결과 (Experimental Results)

- GPT-3 175B 모델을 LoRA로 학습했을 때 **VRAM 소비량이 1.2TB → 350GB로 감소**함.
- LoRA를 적용한 GPT-3 모델의 학습 속도가 **Full Fine-tuning 대비 25% 더 빠름**.
- 다양한 NLP 태스크(예: RoBERTa, DeBERTa, GPT-2, GPT-3)에서 **Full Fine-tuning과 유사하거나 더 나은 성능을 보임**.

5.1 Baselines

- **Fine-Tuning (FT)**:
  - 모든 파라미터를 업데이트하는 전통적인 방법.

    - 다운스트림 태스크마다 동일한 크기의 모델을 저장해야 하는 비효율성.
- **Bias-only (BitFit)**:
    - Pre-trained 모델의 bias 파라미터만 학습.
- **Prefix-embedding tuning (PreEmbed)**:
    - 입력 토큰에 특별한 prefix를 추가해 성능 향상.
- **Prefix-layer tuning (PreLayer)**:
    - 특정 레이어의 입력 임베딩을 수정하여 성능 개선.
- **Adapter tuning**:
    - Transformer 레이어 내에 어댑터 레이어를 삽입하여 학습.
- **LoRA**:
    - 가중치 행렬의 저순위 분해를 통해 메모리 사용량 감소.

5.2 RoBERTa Base/Large

- GLUE 벤치마크를 통해 성능 평가.
- RoBERTa base와 large 모델로 다양한 학습 방법 비교.
- LoRA는 Fine-Tuning과 유사하거나 더 나은 성능을 보여줌.

5.3 DeBERTa XXL

- GLUE 벤치마크에서 DeBERTa XXL 모델을 통해 성능 평가.
- LoRA는 Fine-Tuning 성능에 근접한 결과를 보이며, 메모리 사용량이 현저히 낮음.

5.4 GPT-2 Medium/Large

- E2E NLG Challenge에서 평가 수행.
- LoRA는 BLEU, ROUGE-L, CIDEr 등 여러 메트릭에서 기존 방법보다 우수한 성능을 보임.

5.5 GPT-3 175B

- GPT-3 175B 모델에서 WikiSQL, MultiNLI, SAMSum 데이터셋으로 평가.
- LoRA는 Fine-Tuning 수준의 성능을 보여주면서도 메모리 사용량이 크게 절감됨.

---

6. Related Works

6.1 Transformer Language Models

- Transformer 기반 모델은 NLP 분야에서 우위를 점하고 있음.
- BERT, GPT-2, GPT-3 등의 발전과 파라미터 효율성의 필요성.

6.2 Prompt Engineering and Fine-Tuning

- Prompt 구성 및 학습을 통한 성능 향상.
- GPT-3 175B의 경우 메모리 문제로 인해 전체 Fine-Tuning이 어렵다는 점 지적.

6.3 Parameter-Efficient Adaptation

- Adapter layers, Prefix-tuning 등의 시도와 성능 분석.
- LoRA는 기존 방법 대비 메모리 및 성능 효율성이 우수.

6.4 Low-Rank Structures in Deep Learning

- 머신러닝 모델은 내재적으로 저순위 구조를 갖는 경우가 많음.
- LoRA는 이러한 특성을 이용해 파라미터 수를 크게 줄이는 전략 채택.

Conclusion

- LoRA는 **사전 훈련된 모델의 가중치를 고정하고 저랭크 행렬을 학습하는 방식**으로, **Fine-tuning 대비 연산 비용을 크게 줄이면서도 성능을 유지**하는 혁신적인 접근법이다.
- **추론 속도에 영향을 주지 않으며, 기존 Fine-tuning 기법보다 훨씬 가벼운 메모리 사용량을 제공**한다.
- LoRA는 다양한 NLP 모델과 조합이 가능하며, **특히 대형 언어 모델을 효율적으로 활용하는 데 매우 적합한 방법**이다.