LoRA: Low-Rank Adaptation of Large Language Models

1. INTRODUCTION

- **배경**: Natural Language Processing의 주요 패러다임은 대규모 데이터로 Pre-training 후, 특정 task나 도메인에 맞게 모델을 adaptation시키는 것임.
- **문제점**: 모델 규모가 커지면서 모든 parameter를 재훈련하는 full Fine-Tuning은 비현실적이 됨. 예를 들어, 175B개의 parameter를 가진 GPT-3를 각 task마다 독립적으로 Fine-Tuning하여 배포하는 것은 비용이 엄청나게 많이 듦.
- **제안**: 'Low-Rank Adaptation (LoRA)'이라는 새로운 방법을 제안함. 이 방법은 사전 훈련된 모델의 가중치는 그대로 두고(freeze), Transformer 아키텍처의 각 레이어에 훈련 가능한 'rank decomposition matrices'를 주입함. 이를 통해 downstream task를 위한 훈련 가능한 parameter 수를 대폭 줄임.
- **주요 성과**:
  - GPT-3 175B Fine-Tuning과 비교 시, LoRA는 훈련 가능한 parameter 수를 10,000배, GPU 메모리 요구량을 3배 줄일 수 있음.
  - RoBERTa, DeBERTa, GPT-2, GPT-3에서 Fine-Tuning과 동등하거나 더 나은 성능을 보임.
  - 훈련 가능한 parameter가 더 적고, 훈련 처리량이 높으며, Adapter 방식과 달리 추가적인 inference latency가 없음.
  - 언어 모델 적응 과정에서 나타나는 'rank-deficiency' 현상에 대한 경험적 탐구를 통해 LoRA의 효과를 설명함.

2. PROBLEM STATEMENT

- Full Fine-Tuning의 가장 큰 단점은 각 downstream task에 대해 원본 모델(Φ0)과 동일한 크기의 새로운 parameter 집합(ΔΦ)을 학습해야 한다는 것임.
- 본 논문에서는 ΔΦ를 훨씬 작은 크기의 parameter 집합(Θ)으로 인코딩하여, $|\Theta| \ll |\Phi_0|$를 만족시키는 더 parameter-efficient한 접근 방식을 채택함. LoRA는 이 ΔΦ를 low-rank representation을 사용해 효율적으로 인코딩하는 방법임.

3. AREN'T EXISTING SOLUTIONS GOOD ENOUGH?

- 기존의 효율적인 adaptation 방법들은 크게 두 가지 전략으로 나뉨.
  1. **Adapter Layers 추가**:
     - Transformer 블록마다 작은 신경망(Adapter)을 추가하는 방식임.
     - **단점**: 모델의 깊이가 늘어나 순차적으로 처리해야 하므로 추론 시 지연 시간을 발생시킴(inference latency). 특히 배치 크기가 1인 온라인 추론 환경에서 지연이 눈에 띄게 증가함.
  2. **Prefix-Tuning (프롬프트 튜닝)**:
     - 입력 시퀀스 앞에 특정 'prefix'를 추가하고, 이 prefix에 해당하는 embedding vector만 훈련하는 방식임.
     - **단점**: 최적화가 어렵고 성능이 불안정함. Adaptation을 위해 시퀀스 길이의 일부를 사용해야 하므로, 실제 task를 처리하는 데 사용할 수 있는 context 길이가 줄어드는 근본적인 한계가 있음.
- 이러한 기존 방법들은 효율성과 모델 품질 사이의 trade-off 관계를 가짐.

4. OUR METHOD

- **핵심 가설**: 사전 훈련된 대규모 언어 모델은 본질적으로 'low intrinsic dimension'에 존재하며, 모델을 특정 task에 적응시킬 때 가중치의 변화량(update) 또한 'low intrinsic rank'를 가질 것이라는 가설에서 출발함.
- **작동 원리**:
  - 기존의 사전 훈련된 weight matrix W0는 동결시킴.
  - 가중치 업데이트 ΔW를 두 개의 작은 행렬 B와 A의 곱(ΔW=BA)으로 표현함. 여기서 B∈Rd×r, $A \in \mathbb{R}^{r \times k}$이며, rank r은 원래 차원 d,k보다 훨씬 작음 (r≪min(d,k)).
  - 훈련 중에는 W0는 업데이트하지 않고, 오직 A와 B 행렬만 훈련시킴.
  - 수정된 forward pass는 h=W0x+BAx가 됨.
- **LoRA의 장점**:
  1. **저장 공간 및 작업 전환 효율성**: 하나의 사전 훈련된 모델을 공유하면서, 각기 다른 task를 위한 작은 LoRA 모듈(A,B 행렬)만 교체하면 됨.

2. **훈련 효율성 및 하드웨어 장벽 완화**: 대부분의 parameter에 대한 그래디언트 계산이 필요 없어 VRAM 사용량을 최대 2/3까지 줄임.
3. **No Additional Inference Latency**: 배포 시점에는 W=W0+BA를 미리 계산하여 하나의 행렬로 합칠 수 있어 추가적인 inference latency가 발생하지 않음.
4. **다른 방법과의 결합 용이성**: Prefix-Tuning과 같은 다른 adaptation 방법들과 쉽게 결합하여 사용할 수 있음.

5. EMPIRICAL EXPERIMENTS

- **실험 모델**: RoBERTa, DeBERTa, GPT-2, GPT-3 175B
- **평가 과제**:
  - **NLU**: GLUE benchmark
  - **NLG**: E2E NLG Challenge, WebNLG, DART
  - **대규모 실험**: WikiSQL (NL to SQL), SAMSum (대화 요약)
- **주요 결과**:
  - 모든 모델과 task에서 LoRA는 훨씬 적은 수의 훈련 가능 parameter를 사용하면서도 full Fine-Tuning 및 여러 Adapter, Prefix-Tuning 방법들과 동등하거나 더 나은 성능을 달성함.
  - 특히 GPT-3 175B 실험에서, 다른 방법들은 훈련 가능 parameter 수가 증가할 때 성능이 하락하는 경향을 보였지만, LoRA는 안정적인 성능 향상을 보임.

6. RELATED WORKS

- 본 연구는 Transformer 언어 모델, Parameter-Efficient Adaptation, 그리고 딥러닝에서의 Low-Rank 구조에 대한 기존 연구들과 관련이 있음.
- 기존 Adapter 방식과의 핵심적인 차이점은 LoRA의 학습된 가중치가 추론 시점에 주 모델 가중치와 합쳐져 latency를 발생시키지 않는다는 점임.

7. UNDERSTANDING THE LOW-RANK UPDATES

- **어떤 가중치 행렬에 LoRA를 적용해야 하는가?**:
  - Transformer의 self-attention 모듈에 있는 4개의 weight matrices (Wq,Wk,Wv,Wo) 중, query (Wq)와 value (Wv) 행렬에 동시에 LoRA를 적용했을

때 가장 좋은 성능을 보였음.

- **최적의 rank는 얼마인가?**:
  - 놀랍게도 매우 작은 rank (r=1 또는 r=2)만으로도 경쟁력 있는 성능을 달성함.
  - r=8과 r=64로 학습된 LoRA 모듈의 subspace 유사도를 분석한 결과, 두 경우 모두 상위 singular value에 해당하는 방향이 상당 부분 겹치는 것을 확인함. 이는 높은 rank가 반드시 더 의미 있는 정보를 포착하는 것은 아님을 시사함.
- **업데이트 행렬(ΔW)과 원본 행렬(W)의 관계**:
  - ΔW는 W의 상위 singular direction을 단순히 반복하는 것이 아니라, W에 이미 존재하지만 강조되지 않았던 특정 방향들을 증폭시키는 역할을 함.
  - 결론적으로, LoRA는 사전 훈련 과정에서 학습되었지만 일반적인 목적 때문에 억제되었던, 특정 downstream task에 중요한 특징들을 '증폭'시키는 메커니즘으로 작동한다고 해석할 수 있음.

8. CONCLUSION AND FUTURE WORK

- **결론**: LoRA는 대규모 언어 모델을 효율적으로 적응시키기 위한 강력하고 실용적인 전략임. Inference latency 없이, 적은 메모리와 저장 공간으로 Fine-Tuning과 동등하거나 그 이상의 성능을 달성하여, 다양한 맞춤형 모델을 효율적으로 배포할 수 있는 길을 열었음.
- **향후 연구 방향**:
  1. LoRA를 다른 효율적인 adaptation 방법들과 결합.
  2. LoRA를 통해 Fine-Tuning의 근본적인 메커니즘을 더 깊이 이해.
  3. LoRA를 적용할 weight matrix를 선택하는 더 원칙적인 방법론 연구.
  4. ΔW의 rank-deficiency 현상을 모델 압축이나 이해에 활용하는 연구.