논문 분석

Parameter-Efficient Fine-Tuning with Discrete Fourier Transform

(Ziqi Gao, Qichao Wang, ...)

1. INTRODUCTION

대형 파운데이션 모델(LFMs; Large foundation models)은 자연어 처리(NLP)와 컴퓨터 비전(CV)을 포함한 다양한 분야의 작업에서 탁월한 성능을 보여주고 있다. 이러한 뛰어난 성능 덕분에, 다양한 down-stream 작업에 LFM을 파인튜닝하는 것이 널리 퍼지게 되었다.

전체 파인튜닝(full fine-tuning)방식에서는, 각 맞춤형 작업에 적응된 새로운 모델이 원래 모델과 거의 동일한 수의 파라미터를 포함하게 된다. 모델이 점점 커지고 사용자 맞춤화 요구가 증가함에 따라, 파인튜닝된 체크포인트들을 저장해야 하는 수요도 커지게 되어, 이는 저장 공간과 메모리 사용 측면에서 높은 비용을 초래한다.

이 문제를 해결하기 위한 일반적인 방법으로 LoRA는 가중치 변화(△W)를 두 개의 low-rank 행렬 A와 B로 표현한다. 즉, $W_0 + \triangle W = W_0 + BA$ 형태이다. LoRA가 뛰어난 성능을 보이긴 하지만, 여전히 학습 가능한 파라미터 수가 많아 IT 인프라 자원 소모가 크다는 단점이 있다. 직관적인 예로는 특정 스타일의 스테이블 디퓨전 모델에 대한 LoRA 어댑터가 약 40MB의 메모리를 필요로 한다는 점이다. 이로 인해 Civitai와 같은 LFM 커뮤니티는 방대한 사용자층을 감당하기 위해 높은 저장소 및 네트워크 대역폭 비용을 부담해야 한다. 이에 떠오르는 질문이, 'LFM을 파인튜닝할 때, 학습 가능한 파라미터를 더 과감하게 압축할 수는 없는가?' 이다.

이전 연구들은 데이터 압축에서 푸리에 기저(Fourier basis)의 강력한 표현력을 보여주었다. 매우 희소한 스펙트럼 정보만으로도 고품질의 데이터를 복원할 수 있다는 것이다. 예를 들어, 1차원 신호 벡터나 2차원 이미지 행렬 등이 있다.

더 중요한 점은, 이미지가 아닌 일반적인 행렬(강한 공간적 의미를 가지지 않으며, 주파수 영역에서 희소하지 않은 행렬)을 다룰 때에도 푸리에 변환을 통해 효과적인 복원이 가능하다는 점이다.

이러한 점에 착안하여, 이 논문은 LFM을 파인튜닝할 때 가중치 변화(△W)를 희소한 스펙트럼 계수(sparse spectral coefficients)로 업데이트하는 가능성을 탐색한다.

이 논문에서는, LFM(대형 기반 모델)의 파인튜닝 시 학습해야 하는 파라미터 수를 과감하게 줄이는 것을 목표로 한다. 이를 위해, 논문에서는 FourierFT (Fourier Transform for Fine-Tuning) 라는 방법을 제안한다. 이 방법은 가중치 변화(△W)를 공간 도메인(spatial domain) 상의 행렬로 간주하고, 그것의 희소한 스펙트럼 계수(sparse spectral coefficients)를 학습한다.

구체적으로는, 먼저 모든 레이어에서 공통으로 사용할 n개의 스펙트럼 위치를 무작위로 선택한다. 이후 각 레이어별로 FourierFT는 선택된 그 n개의 위치에 해당하는 n개의 스펙트럼 계수를 학습하고, 이를 역 이산 푸리에 변환(Inverse Discrete Fourier Transform)을 통해 △W를 직접 계산한다.
따라서, 총 L_t개의 레이어를 가진 사전학습 모델을 FourierFT 방식으로 파인튜닝 할 경우, 단지 2n개의 위치 파라미터(entry parameters)와 n×L_t개의 계수 파라미터(coefficient parameters)만 저장하면 된다.

실험적으로, 우리는 제안한 FourierFT 방법을 최신 LoRA 변형들과 다른 파라미터 효율적 파인튜닝 기법들과 비교하였다. 비교 대상 작업은 다음과 같다:

가. 자연어 이해 (GLUE 벤치마크 기준),
나. 자연어 생성 (E2E 벤치마크 기준),
다. 명령어 튜닝 (LLaMA 계열 모델 사용),
라. 이미지 분류 (Vision Transformer 사용).

FourierFT는 이 4가지 작업에서 항상 LoRA와 동등하거나 더 나은 성능을 보여주었으며, LoRA가 사용하는 학습 가능한 파라미터 수의 각각 6.0%, 9.4%, 0.2%, 9.2%만을 사용해도 충분했다.

예를 들어 Figure 1에서는, 명령어 튜닝 작업에서 FourierFT는 단 64K개의 학습 파라미터만으로도 LoRA보다 더 우수한 성능을 보였고, 128K개의 파라미터만으로도 Full Fine-tuning과 유사한 점수를 기록했다.

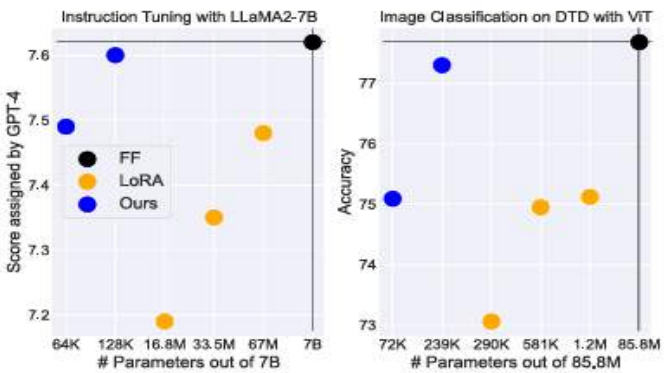

Figure 1. Summary of the performance (**y-axis**) of fine-tuning methods with different numbers (**x-axis**) of trainable parameters on NLP (left) and CV (right) tasks. The left side shows the instruction tuning task, where the LLaMA2-7B model is fine-tuned with Alpaca and evaluated by GPT-4. The right side shows the image classification task, where the Vision Transformer (ViT) is fine-tuned and tested on the DTD dataset. *Black circles* (•) represent the Full Fine-tuning (FF) method. *Orange circles* (•) represent LoRA method with $r = \{32, 64, 128\}$ (left) and $r = \{8, 16, 32\}$ (right). *Blue circles* (•) represent our proposed method with $n = \{1000, 2000\}$ (left) and $n = \{3000, 10000\}$ (right).

2. Related Works

PEFT(Parameter-Efficient Fine-Tuning)

대형 파운데이션 모델(LFM)의 급속한 확장과 함께, 이를 특정 작업에 효율적으로 적응시키는 것이 점점 더 중요하고 도전적인 과제가 되었다. 이를 위해, 효율성과 정확성 면에서 인상적인 성능을 보여주는 다양한 PEFT (파라미터 효율적 미세 조정) 기법들이 제안되었다. 기존의 PEFT 방법들은 크게 비가중치 기반(non-weight-based) 방법과 가중치 기반(weight-based) 방법의 두 가지 범주로 나눌 수 있다.

A. Non-weight-based method

비가중치 기반 방법(Non-weight-based methods)은 사전 학습된 LFM의 가중치를 직접적으로 최적화하지 않는다. 대신, 추가적인 모듈을 삽입하거나 프롬프트(prompt)와 프리픽스(prefix)를 최적화함으로써 미세 조정을 수행한다.

- 어댑터 튜닝(Adapter tuning): 어댑터(adapter)라고 불리는 경량 뉴럴 모듈을 사전학습된 레이어 사이에 삽입하여 사용하는 방식이다. 이 방식은 사전학습된 가중치는 동결(freeze) 시키고, 어댑터만 효율적으로 미세 조정하여 특정 작업에 적합하게 한다.
- 프롬프트 튜닝(prompt tuning)과 프리픽스 튜닝(prefix tuning): 모델의 레이어에 추가적인 프롬프트나 프리픽스 토큰을 삽입하여 조정한다.

B. Weight-based method

가중치 기반(weight-based) 방법은 LoRA로 대표되며, 이는 원래의 가중치에 합쳐질 수 있는(weight change) 변화량을 학습하는 방식이다. 이 방식은 추론 시 지연을 피하기 위해 사전학습된 가중치와 weight change를 합칠 수 있다는 장점이 있다. LoRA의 핵심 혁신은 저랭크 행렬의 곱으로 가중치 변화량을 근사하는 데 있다. 이를 기반으로 한 확장 방법에는 다음이 있다.

- AdaLoRA (Zhang et al., 2023): LoRA 방식을 확장하여 가중치 행렬의 중요도 스코어에 따라 파라미터 예산을 분배하는 기법.
- Q-LoRA (Dettmers et al., 2023): 4비트 NormalFloat 형식으로 양자화된 사전학습 모델에 대해 LoRA를 적용하면서 역전파(gradient)를 수행하는 방식.

이번 논문에서는 저랭크 구조를 따르기보다는, 푸리에 기저(Fourier basis)의 강력한 표현력을 이용하여 가중치 기반 방식으로 파라미터를 대폭 줄이는 것에 초점을 맞춘다.

C. Sparse Fourier Transform in Deep Learning

희소 푸리에 변환(SFT)은 최근 다양한 딥러닝 분야에서 활발히 활용되고 있다. SFT 기술은 주로 중요하거나 혹은 무작위로 선택된 희소한 스펙트럼 항을 사용하여 표현 학습(representation learning)을 수행하는 것이 핵심이다.

이 기술의 중요한 활용 사례 중 하나는 행렬 복원(matrix recovery)이다. Patel et al. (2011)은 희소한 푸리에 정보만으로 이미지를 복원하는 그래디언트 기반 압축 센싱

(compressed sensing) 방법을 설계했고, Shechtman et al. (2014)은 **희소 푸리에 계수(sparse Fourier coefficients)**를 활용하여 데이터 복원 성능을 향상시키는 효율적인 위상 복원(phase retrieval) 방법을 제안했다.

중요한 점은 Chen & Chi (2013), Yang & Xie (2016), Gao et al. (2022) 등의 연구에 따르면, 원본 데이터가 주파수 영역에서 희소하지 않더라도, SFT는 매우 적은 수의 파라미터로도 데이터를 효과적으로 복원할 수 있다는 것이다.

비록 기존 연구에서는 딥러닝 모델의 가중치 행렬을 SFT로 복원하는 것에 대한 연구는 부족하지만, 위에서 언급한 연구들은 이 논문의 접근 방식에 대한 잠재적 가능성을 뒷받침해준다.

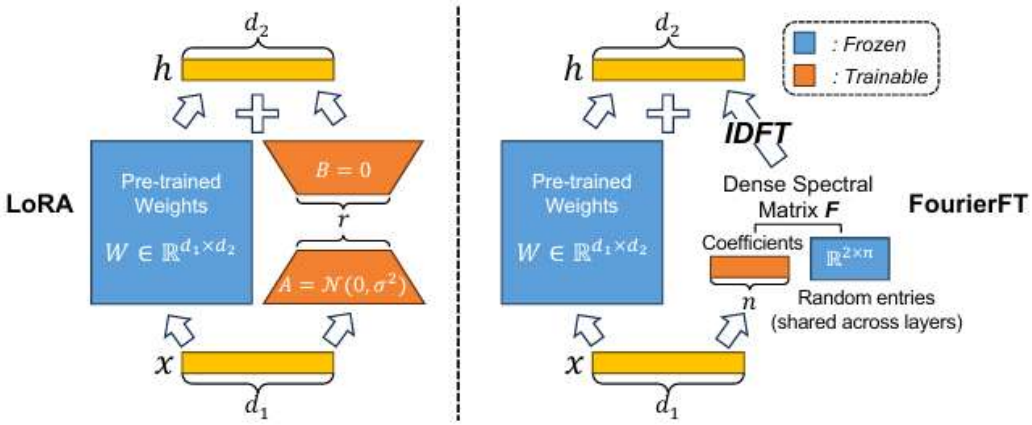

Figure 2. **Overview of LoRA (left) and our FourierFT (right) method.** In LoRA, only low-rank (r) matrices A and B are trained. The weight change is represented by their multiplication, i.e., $\Delta W = BA$. For each pre-trained weight W, the theoretical number of trainable parameters in LoRA is $r \times (d_1 + d_2)$. In FourierFT, we first randomly generate the spectral entry matrix $\mathbb{R}^{2 \times n}$, which is shared across all layers to reduce parameter storage requirements. The complete spectral matrix is formed by a trainable coefficient vector $\mathbb{R}^n$ located at selected entries and 0s at the remaining entries. We obtain the weight change ΔW by directly performing inverse discrete Fourier transform (IDFT) on the updated spectral matrix. For all L adapted layers, FourierFT needs to store $n \times (2 + L)$ parameters.

3. Method

이 논문에서는 FourierFT를 제안하는데, 이는 이산 푸리에 변환(Discrete Fourier Transform)에 기반한 파라미터 효율적인 파인튜닝 방법이다. FourierFT는 사전학습된 가중치(weight)의 변화만 학습한다는 LoRA의 원칙을 따르지만, LoRA와는 달리 저랭크 구조(low-rank structure)를 채택하지 않고, **푸리에 기저(Fourier basis)의 스펙트럼 계수(spectral coefficients)를 학습한다.**

구체적으로는, (1) 먼저 스펙트럼 항의 위치(entry positions)를 나타내는 행렬을 무작위로 초기화하며, 이 행렬은 모든 계층(layer)에 대해 고정(frozen)되어 공유된다. (2) 그런 다음, 선택된 위치에 있는 스펙트럼 계수만 학습 가능하도록 설정한다. 이 학습 가능한 값들이 전체 스펙트럼 행렬을 구성한다. (3) 마지막으로, 이 스펙트럼 행렬에 대해 역 이산 푸리에 변환(inverse DFT)을 적용하여, 공간 영역(spatial domain)에서의 대응 가중치 변화(△W)를 생성

한다.

A. Forward Pass

우리는 LoRA 기반 방법들이 채택한 방식처럼, 가중치 변화(weight changes)만 학습하는 패러다임을 따른다. 이 접근 방식은 사전 학습된 가중치(W_0)와 그 변화량(ΔW)을 병합하여 추론 시 지연을 피할 수 있다는 장점이 있다.

형식적으로, 각 사전학습된 가중치 행렬은 $W_0 \in R$(d1xd2)로 정의하고, 파인튜닝을 위한 가중치 변화량은 $\Delta W \in R$(d1xd2)로 정의한다. LoRA는 순전파(forward pass) 과정에서 가중치 변화 ΔW를 저랭크 분해(low-rank decomposition) 형태로 파라미터화하는 것을 목표로 한다.

$$h = W_0 x + \Delta W x = W_0 x + BAx, \tag{1}$$

where $B \in \mathbb{R}^{d_1 \times r}$ and $A \in \mathbb{R}^{r \times d_2}$ with the rank $r \ll \min(d_1, d_2)$ are trainable matrices.

FourierFT의 장점은, 직교성(orthogonality)과 표현력(expressiveness)을 지닌 Fourier basis 덕분에 유의미한 가중치 변화(weight change)를 효과적으로 복원할 수 있다는 점이다. 이러한 특성은, 훨씬 적은 파라미터만으로도 LoRA와 비슷한 성능을 달성할 수 있을 가능성을 시사한다.

우리는 먼저 이산 2D 스펙트럼 항목들을 담은 엔트리 행렬 $E \in R(2 \times n)$를 무작위로 초기화하고, 이어서 스펙트럼 계수 $c \in R(n)$를 정규분포(Gaussian distribution)를 따라 무작위로 초기화한다.

$$F = \textsc{ToDense}(E, c) \tag{2}$$

$$S_{p,q} = \sum_{j=0}^{d_1-1} \sum_{k=0}^{d_2-1} F_{j,k} e^{i2\pi\left(\frac{p}{d_1} j + \frac{q}{d_2} k\right)} \tag{3}$$

$$\begin{aligned} h &= W_0 x + \Delta W x \\ &= W_0 x + \alpha \Re(S) x. \end{aligned} \tag{4}$$

구체적으로, 식 (2)에 나오는 TODENSE는 스펙트럼 행렬 F∈R(d1xd2)를 구성하는 것을 의미한다. 즉, F(j,k)=cl(또는 0) 이 되는데, 이때 $j=E(0,l)$ 그리고 $k=E(1,l)$인 경우엔 cl값을 넣고, 그 외의 경우엔 0을 넣는 방식이다. 식 (3)에서는 역 이산 푸리에 변환 (inverse discrete Fourier transform) 을 사용해, 공간 도메인 행렬 S를 계산한다. 여기서 i는 허수 단위

(imaginary unit) 를 나타낸다. 마지막으로 식 (4)에서는, 복소수 행렬 S의 실수 부분만 추출한 뒤 α값으로 스케일링한다. 이때 실수 부분은 $R(S)$로 표기된다. 참고로, 모든 레이어는 각각 다른 c벡터를 학습하지만,E 행렬과 α 값은 모든 레이어에서 공유된다.

- Initialization for the Entry Matrix E

이전 연구들은 weight change(가중치 변화) 에서 스펙트럼 항목(spectral entries)의 중요성에 관한 연구가 부족했다. 그래서 이 논문에서는 이 공백을 메우기 위해 **조절 가능한 주파수 바이어스 (adjustable frequency bias)** 라는 기법을 도입했다. 이 바이어스를 통해 특정 주파수 영역의 항목들이 더 자주 샘플링되도록 만들 수 있는데, 단순히 전체 $d1 \times d2$ 크기의 스펙트럼 행렬에서 완전히 무작위로 샘플링(no bias) 하는 것뿐 아니라, 중심 주파수가 있는 특정 영역(예: 저주파, 중간 주파수, 고주파)을 더 많이 선택하도록 편향된 샘플링도 수행할 수 있다. 이를 수학적으로 설명하면 다음과 같다:

특정 entry (u,v) (여기서 $0 \le u \le d1-1$, $0 \le v \le d2-1$)에 대해 샘플링 확률을 모델링하기 위해 가우시안 밴드패스 필터(Gaussian bandpass filter) 를 적용한다. 이 필터는 Gonzales & Wintz (1987) 의 방식에 기반한다.

$$p(u,v) = \exp\left(-\left(\frac{\mathcal{D}^2 - f_c^2}{\mathcal{D}W}\right)^2\right), \qquad (5)$$

여기서 D는 스펙트럼 행렬 상의 한 점 (u,v)로부터 원점(행렬의 중심)까지의 거리를 의미하고,fc는 우선적으로 선택하고자 하는 중심 주파수 (favored central frequency)를, W는 밴드 폭 (bandwidth) 을 나타낸다. Figure 3에서는 768×768 크기의 스펙트럼 행렬에 대해 다양한 fc값을 사용하고, W=200으로 설정했을 때의 샘플링 확률 맵을 시각화한 결과를 보여준다.

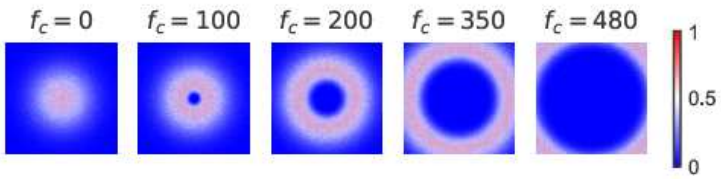

Figure 3. Visualization of entry sampling probability at different favored central frequencies f_c.

또한 다음 사항을 유의해야 한다:

특별히 명시하지 않는 한, FourierFT는 기본적으로 frequency bias 없이, 즉 완전히 무작위 entry 초기화 방식(no frequency bias)을 사용한다.

B. Parameter Summary

논문에서는 LoRA와 FourierFT의 학습 가능한 파라미터 수를 Table1에 요약했다.

Table 1. Theoretical number of trainable parameters and storage requirements for fine-tuning. For both LoRA and FourierFT methods, only the query and value layers are tuned within the transformer architectures. The configurations that are exactly chosen in the 'Experiments' Section are highlighted.

Base Models	LoRA			FourierFT		
	r	# Trainable Parameters	Required Bytes	n	# Trainable Parameters	Required Bytes
RoBERTa Base	4	147K	574KB	200	4.8K	18.8KB
	8	295K	1.13MB	200	24K	94KB
RoBERTa Large	4	393K	1.5MB	200	9.6K	36.5KB
	8	786K	3MB	1000	48K	183KB
GPT-2 Medium	4	350K	1.34MB	500	24K	94KB
	8	786K	3MB	1000	48K	188KB
GPT-2 Large	4	737K	2.81MB	500	36K	141KB
	8	1.47M	5.74MB	1000	72K	282KB
LLaMA-2 7B	16	8.39M	32.8MB	1000	64K	250KB
	64	33.5M	131.1MB	2000	128K	500KB
LLaMA-2 13B	16	13.1M	51.2MB	1000	80K	312KB
	64	52.4M	204.8MB	2000	160K	625KB
ViT Base	8	295K	1.13MB	3000	72K	281KB
	16	590K	2.25MB	10000	239K	934KB
ViT Large	8	786K	2.93MB	3000	144K	563KB
	16	1.57M	6MB	10000	480K	1.83MB

LoRA는 각 레이어마다 학습 가능한 두 개의 행렬 A와 B를 사용하는데, 파인튜닝 대상 레이어 수를 Lt, 가중치의 차원을 d=d1=d2, 랭크(rank)를 r이라고 할 때, LoRA의 전체 학습 파라미터 수는 다음과 같다: $|\Theta| = 2 \times d \times Lt \times r$

반면, FourierFT는 다음과 같다: $|\Theta| = n \times Lt$ (n은 학습 가능한 스펙트럼 계수의 수)

특히, 모델의 스케일(깊이와 너비)가 커질수록, FourierFT의 파라미터 효율성 이점이 더욱 두드러지게 나타나는데(예: RoBERTa-Base -> RoBERTa-Large), 이는 아마도 LoRA의 파라미터 수가 width d와 선형적인 관계를 갖는 반면, FourierFT는 그렇지 않기 때문일 것이다.

4. Experiments

자연어 처리(NLP)와 컴퓨터 비전(CV) 분야에서 FourierFT의 성능을 평가한다. NLP 분야에서는 다음과 같은 fine-tuning 실험을 수행했다:

- RoBERTa (Base 및 Large) 모델을 사용하여 자연어 이해 태스크 수행
- GPT-2 (Medium 및 Large) 모델을 사용하여 자연어 생성 태스크 수행
- LLaMA 계열 모델 (7B 및 13B)을 사용하여 instruction tuning 태스크 수행

CV 분야에서는 다음을 수행했다: Vision Transformer (Base 및 Large) 모델을 이미지 분류 태스크에 대해 fine-tuning

마지막으로, 다음과 같은 Ablation Study도 함께 진행했다:

- 주파수 바이어스(frequency bias) 의 영향 분석
- 파라미터 확장성(parameter scalability) 분석
- Fourier basis의 표현력(expressiveness) 평가

이 논문에서는 FourierFT 기법을 널리 사용되는 효율적인 파인튜닝(PEFT) 방법들과 비교한다. 비교 대상 기법들은 다음과 같다.

- Full Fine-tuning(FF): 모든 파라미터를 학습하는 방식. 모델은 사전 학습된 가중치와 편향으로 초기화되며, 모든 파라미터에 대해 그래디언트 업데이트가 수행됨.
- BitFit: 편향 벡터만 파인튜닝. 나머지 모든 파라미터는 고정된 상태로 두고, 오직 bias만 업데이트 됨.
- Adapter Tuning: AdapterH는 Self-Attention과 FNN 사이에 2계층 어댑터를 삽입하고, residual connection을 따름. AdapterH의 변형에는 다음과 같은 것들이 있음.
  - AdapterL: MLP 모듈 뒤, LayerNorm 이후에만 어댑터 적용 -> 더 적은 파라미터 사용.
  - AdapterP: FNN 이후에 어댑터 삽입. 어댑터 위치, 개수 등을 포함한 설정을 그리드 서치로 탐색하여 선택됨.
  - AdapterD: 활성화되지 않은 어댑터 레이어는 제거함으로써 파라미터의 효율성을 더욱 높임.
- LoRA: 현재 가장 널리 쓰이는 PEFT 기법으로, 학습 가능한 저랭크 행렬을 사용해 가중치의 변화를 파라미터화한다.
- DyLORA: LoRA haepf의 최적 랭크를 동적으로 선택하면서도, 하이퍼파라미터 탐색 없이 학습할 수 있도록 설계된 기법.
- AdaLoRA: SVD를 기반으로 한 파인튜닝 기법으로, 중요도를 고려해 불필요한 특이값을 제거하며 랭크를 효율적으로 할당.

A. Natural Language Understanding

이 논문에서는 GLUE 벤치마크(General Language Understanding Evaluation)를 사용하여 제안한 기법을 평가했다. GLUE는 단일 문장 분류, 유사성 및 패러프레이즈 과제, 자연어 추론 과제 등 다양한 자연어 이해(NLU) 과제들로 구성돼 있다. 이를 위해, 논문에서는 사전 학습된 RoBERTa Base 및 RoBERTa Large 모델을 파인튜닝하여 성능을 비교 및 분석하였다.

두 모델(RoBERT) 모두에서 FourierFT는 각 레이어마다 1000개의 학습 가능한 스펙트럼 계수(n=1000)를 사용할 수 있도록 설정하였다. 스펙트럼 항목은 주파수 편향 없이 무작위로 샘플링되며, 모든 레이어(Base: 24개, Large: 48개)에서 동일한 항목 구성이 공유된다. GLUE의 6개의 데이터셋 모두에 대해, 학습률과 스케일링 계수를 하이퍼파라미터로 조정하였다. 실험 설정으로는 트랜스포머 블록 내의 쿼리와 벨류(query, value) 가중치만 파인튜닝하고, 분류 헤드 전체 파인튜닝한다.

각 실험은 5개의 서로 다른 random seed로 수행되며, 각 실험에서 가장 성능이 좋았던 epoch의 결과를 사용하여 중앙값을 보고했다. 전반적으로 FourierFT는 훨씬 적은 학습 파라미터 수로도 기존 PEFT 기법들보다 우수하거나 동등한 성능을 달성했고, 특히 FourierFT는 CoLA에서 RoBERTa Base를 full fine-tuning한 경우와 RTE에서 RoBERTa Large를 full fine-tuning한 경우를 모두 능가하는 성능을 보였다. LoRA의 파라미터 수는 모델의 너비와 깊이에 따라 선형적으로 증가하기 때문에, LoRA의 파라미터 수는 Base->Large 전환 시 약 2.7배 증가하지만 FourierFT는 2배 증가에 불과하다. 그럼에도 불구하고 FourierFT는 LoRA와 유사한 성능을 보였으며, 이는 더 큰 모델에 대해서도 높은 확장 가능성(scalability)를 보여준다.

Table 3. Results from GPT-2 Medium and Large models on the E2E benchmark. We present the result from the final epoch. For all metrics, higher values indicate better performance. * indicates that the results are taken from prior works. Best results are shown in **bold**.

Model	Method	# Trainable Parameters	BLEU	NIST	METEOR	ROUGE-L	CIDEr
GPT-2 Medium	FT*	354.92M	68.2	8.62	46.2	71.0	2.47
	Adpt^{L}*	0.37M	66.3	8.41	45.0	69.8	2.40
	Adpt^{L}*	11.09M	68.9	8.71	46.1	71.3	2.47
	Adpt^{H}*	11.09M	$67.3_{\pm.6}$	$8.5_{\pm.07}$	$46.0_{\pm.2}$	$70.7_{\pm.2}$	$2.44_{\pm.01}$
	LoRA	0.35M	$68.9_{\pm.3}$	$8.76_{\pm.06}$	$46.6_{\pm.1}$	$71.5_{\pm.1}$	$\mathbf{2.53}_{\pm.03}$
	FourierFT	0.048M	$\mathbf{69.1}_{\pm.1}$	$\mathbf{8.82}_{\pm.05}$	$\mathbf{47.0}_{\pm.3}$	$\mathbf{71.8}_{\pm.1}$	$2.51_{\pm.02}$
GPT-2 Large	FT*	774.03M	68.5	8.78	46.0	69.9	2.45
	Adpt^{L}*	0.88M	$69.1_{\pm.1}$	$8.68_{\pm.03}$	$46.3_{\pm.0}$	$71.4_{\pm.2}$	$2.49_{\pm.0}$
	Adpt^{L}*	23.00M	$68.9_{\pm.3}$	$8.70_{\pm.04}$	$46.1_{\pm.1}$	$71.3_{\pm.2}$	$2.45_{\pm.02}$
	LoRA	0.77M	$70.1_{\pm.3}$	$8.83_{\pm.02}$	$46.8_{\pm.2}$	$\mathbf{72.0}_{\pm.3}$	$2.47_{\pm.02}$
	FourierFT	0.072M	$\mathbf{70.2}_{\pm.2}$	$\mathbf{8.90}_{\pm.02}$	$\mathbf{47.0}_{\pm.2}$	$71.8_{\pm.1}$	$\mathbf{2.50}_{\pm.02}$

Table 4. The average scores on MT-Bench and Vicuna assessed by GPT-4. † indicates updating the layers other than `lm_head`. Higher score is better.

Model	Method	# Trainable Parameters	MT-Bench	Vicuna
LLaMA1-7B	LoRA†	159.9M	$5.05_{\pm.3}$	$\mathbf{6.85}_{\pm.4}$
	LoRA	33.5M	$4.99_{\pm.3}$	$6.81_{\pm.3}$
	FourierFT	0.064M	$\mathbf{5.09}_{\pm.6}$	$\mathbf{6.85}_{\pm.8}$
LLaMA1-13B	LoRA†	250.3M	$\mathbf{5.28}_{\pm.6}$	$7.02_{\pm.3}$
	LoRA	52.4M	$5.21_{\pm.4}$	$6.97_{\pm.4}$
	FourierFT	0.08M	$5.23_{\pm.3}$	$\mathbf{7.14}_{\pm.5}$
LLaMA2-7B	LoRA†	159.9M	$5.19_{\pm.1}$	$7.38_{\pm.3}$
	LoRA	33.5M	$\mathbf{5.20}_{\pm.3}$	$7.35_{\pm.6}$
	FourierFT	0.064M	$5.18_{\pm.3}$	$\mathbf{7.49}_{\pm.4}$
LLaMA2-13B	LoRA†	250.3M	$5.78_{\pm.2}$	$7.89_{\pm.5}$
	LoRA	52.4M	$5.80_{\pm.2}$	$7.89_{\pm.6}$
	FourierFT	0.08M	$\mathbf{5.82}_{\pm.3}$	$\mathbf{7.92}_{\pm.5}$

B. Natural Language Generation

논문에서는 FourierFT의 성능을 E2E 자연어 생성(NLG) 태스크에서 평가한다. GPT-2 Medium 및 GPT-2 Large 모델을 파인튜닝하며, 이 두 모델은 모두 디코더 전용 모델이고, 각 24개, 36개의 transformer 블록을 가지고 있다.

LoRA와 FourierFT 모두 동일한 방식으로 GPT-2 Medium과 Large를 5 epoch 동안 학습한다. 학습률은 선형 스케줄러를 사용하며, 배치 사이즈 및 학습률은 튜닝한다. 각 실험은 3번 실행 후 평균을 보고하며, 각 run의 마지막 에폭 결과를 사용한다.

결과로, FourierFT는 대부분의 지표에서 가장 좋은 성능을 달성했고, 더 중요한 점은 FourierFT가 LoRA 대비 훨씬 적은 파라미터 수로 이 성능을 달성한다는 것이다.

C. Instruction Tuning

Instrunction tuning은 프롬프트-응답 쌍으로 구성된 데이터셋을 사용하여 언어 모델을 파인튜닝하는 과정을 의미한다. 본 논문에서는 LoRA 및 FourierFT를 적용하여 LLaMA와 LLaMA2 계열 모델을 파인튜닝한다. 이 모델들은 Alpaca 데이터셋으로 파인튜닝했고, 평가 방법은 MT-Bench 및 Vicuna Eval에서 정의된 질문들에 대해 응답을 생성해서 평가했다.

LoRA는 r=64로 설정했고, 언어 모델링 헤드(lmhead)를 제외한 모든 선형 계층을 업데이트 및 쿼리(WQ) 및 값(WV) 행렬만 업데이트했다. FourierFT 설정 역시 WQ와 WV만 업데이트 했고, n=1000으로 설정했다. 학습은 단 1 epoch만 수행했고, 모든 응답에 대한 평균 점수를 결과로 보고했다.

결과로, LLaMA-13B가 7B 모델보다 확실히 강력한 표현력을 가짐을 알 수 있었고, FourierFT는 LoRA의 성능에 매우 근접하거나 약간 능가하는 모습을 보였다. LoRA 대비 0.2% 미만의 파라미터만 사용하면서 말이다.

D. Image Classifcation
E. Study
F. Basis Expressiveness

생략.

5. Conclusion

본 논문에서는 대형 파운데이션 모델의 파인튜닝을 위한 매우 낮은 저장 메모리 사용을 목표로 한다. 이러한 접근은 다양한 도메인, 과제 또는 사용자 맞춤 설정에 따라 여러 개의 파인튜닝을 가능하게 한다. 이를 위해, 논문에서는 가중치 변화(weight change)를 공간 도메인 행렬로 간주하고, 주파수 도메인에서 희소 계수(sparse coefficients)만을 학습하는 단순하면서도 강력한 파인튜닝 방법을 제안한다.

LoRA 계열의 기존 방식들과 비교했을 때, 본 방법은 NLP 및 CV 과제 전반에 걸쳐 학습 가능한 파라미터 수를 약 8배에서 최대 500배까지 감소시킨다.