공학석사 학위논문

# 파라미터 효율적 전이 학습 기법의 균질성 분석

Analysis on the Uniformity of Parameter Efficient Transfer Learning Method

2023 년 2 월

서울대학교 대학원

컴퓨터 공학부

박 충 현

공학석사 학위논문

# 파라미터 효율적 전이 학습 기법의 균질성 분석

Analysis on the Uniformity of Parameter Efficient Transfer Learning Method

2023 년 2 월

서울대학교 대학원

컴퓨터 공학부

박 충 현

# 파라미터 효율적 전이 학습 기법의 균질성 분석

Analysis on the Uniformity of Parameter Efficient Transfer Learning Method

지도교수 이 상 구

이 논문을 공학석사 학위논문으로 제출함

2022 년 12 월


서울대학교 대학원

컴퓨터 공학부

박 충 현


박충현의 석사 학위논문을 인준함

2022 년 12 월

| | | |
|---|---|---|
| 위 원 장 | 문 봉 기 | (인) |
| 부위원장 | 이 상 구 | (인) |
| 위 원 | 강 유 | (인) |

# 초 록

사전학습 언어 모델은 문장 내 단어를 예측하는 학습 과정을 통해 전반적인 언어 능력을 습득한 모델이다. 자연 언어 처리는 이러한 사전학습 언어 모델의 도움을 받아 다양한 분야에서 괄목할 만한 성과를 이뤄내고 있다. 사전학습 결과를 기반으로 다른 응용 태스크를 학습할 경우 기존 모델에 비해 높은 성능을 보이며, 성능 향상 이외에도 같은 태스크에 대한 학습 결과들을 서로 가깝게 만드는 효과를 얻을 수 있다. 이러한 특성을 활용해 다수의 학습에서 얻은 가중치의 평균값을 학습 태스크에 대한 하나의 모델로 이용할 수 있다. 이 경우 추론 과정에 추가적인 연산 없이 학습 모델의 성능을 개선할 수 있다.

사전학습 언어 모델은 크기가 클 수록 더 좋은 성능을 보이며, 더 좋은 모델을 얻기 위해 천억 개 이상의 파라미터를 가진 거대 언어 모델이 만들어지고 있다. 거대 언어 모델은 기존 사전학습 모델을 상회하는 성능을 보이고 있지만, 각 응용 태스크에 맞게 조정하고 태스크 별 학습 결과를 저장하기에는 크기가 너무 크다는 단점이 있다. 이에 따라, 모델 전체를 학습하는 대신 적은 수의 파라미터만으로 모델 학습을 대신하는 파라미터 효율적 전이 학습 기법이 제안되고 있다.

기존 파라미터 효율적 전이 학습 관련 연구에서 적은 파라미터만 학습해 모델 자체를 미세 조정하는 수준의 학습 결과를 얻을 수 있는 다양한 기법을 제시하였다. 하지만, 각 학습 결과 사이의 유사성이 파라미터 효율적 전이 학습할 때도 유지되는지에 대한 연구는 부족하다. 본 논문에서는 대표적인 파라미터 효율적 전이 학습 기법인 LoRA를 사용할 때 데이터셋 별로 서로 다른 학습 결과 사이 유사성을 점검한다. 모델 자체를 미세 조정한 결과에 비해서는 유사성이 감소하지만, 학습 파라미터 수에 관계없이 주어진 입력에 대한 모델의 출력 표현이 서로 비슷하고, 손실 landscape 상에서 LoRA 가중치의 최종 위치가 동일한 분지에 모인다는 사

실을 확인하였다. 이러한 성질은 해당 태스크를 학습하여 높은 성능을 얻기 위해 필요한 파라미터 양에 영향을 받는다.

LoRA 학습 결과 사이의 유사성을 바탕으로, 서로 다른 학습 가중치들 사이의 단순 가중합으로 모델 가중치를 대체하는 단순한 모델 앙상블 기법을 파라미터 효율적 학습 결과에 적용하였다. 탐욕 알고리즘으로 모델을 선정했을 때, 추론 시 계산량을 단일 모델 수준으로 유지하면서, 더욱 향상된 성능을 얻을 수 있음을 경험적으로 보였다.

**주요어**: 파라미터 효율적 전이 학습, Linear Mode Connectivity, 특징 유사도, LoRA, 사전학습 언어 모델, 앙상블 모델, 미세 조정

**학번**: 2021-20229

# 목차

# 그림 목차

# 표 목차

# 제 1 장 서론

## 1.1 연구 배경 및 내용

사전학습 언어 모델은 방대한 양의 데이터에서 언어 모델링 학습 과정을 거친 언어 모델을 말한다. 이렇게 학습된 파라미터를 바탕으로 응용 태스크의 데이터에 추가적인 미세 조정을 한 결과 사전학습 언어 모델은 자연 언어 처리 전반에서 높은 성능을 보였다. 사전학습 언어 모델은 모델의 크기가 커지고, 사전학습에 사용한 데이터의 양을 늘릴 수록 그 성능이 계속 증가하는 경향을 보여왔다. 이에 따라 사전학습 언어 모델의 크기는 빠르게 증가하였고, 현재 GPT-3[1]를 비롯해 천억 개를 넘는 파라미터를 가진 거대 언어 모델이 발표되고 있다. 거대 언어 모델은 질의 응답, 문서 번역, 요약 등 다양한 문제에서 더욱 발전된 결과를 보이고, 다단계 추론, 수학 연산 등 작은 모델이 수행할 수 없었던 태스크를 해결할 수 있다.[2]

사전학습 결과를 기반으로 학습하면 성능 향상 외에 각 학습 결과를 균일하게 만드는 효과도 얻을 수 있다. 기계학습 분야에서 널리 쓰이는 경사 하강법은 입력이 주어졌을 때 현재 시점의 모델 가중치를 기준으로 그 입력에 대한 손실을 낮출 수 있는 모델 수정 방향을 찾고, 해당 방향으로 가중치를 변경하는 과정을 반복한다. 이러한 학습 방식은 학습 종료 시 동일한 결과를 보장하지 않으며, 일반적으로 가중치를 임의의 값으로 초기화한 모델을 같은 태스크에서 학습할 때 최종 학습 가중치가 매 경우에 크게 달라진다. 이와 달리, 사전학습된 가중치를 기반으로 같은 응용 태스크에서 여러 번 학습할 경우 각각의 최종 학습 가중치들이 유사성을 보인다. [3, 4, 5]는 사전학습 모델을 미세 조정한 서로 다른 학습 결과들을 손실 landscape 상에 나타냈을 때 서로 다른 학습 결과가 소수의 공통된 분지 중 하나에 도달하는 linear mode connectivity를 서로 다른 분야에서 각각 실험적으로 확인했

다. 또한, [3]은 같은 데이터셋으로 모델을 여러 번 학습한 뒤 동일한 학습 데이터를 각 모델에 입력했을 때 출력하는 은닉 상태를 비교하였고, 사전학습된 가중치를 기반으로 학습을 시작한 모델들의 출력이 임의로 초기화한 모델들의 출력에 비해 서로 유사하다는 사실을 보고했다.

사전학습 모델을 활용한 학습이 다양한 장점을 가지고 있지만, 최근 모델의 크기 증가에 따라 새로운 문제가 대두되었다. 거대해진 모델을 미세 조정하기 위해 필요한 계산량이 크게 늘어났고, 태스크 별 학습 결과를 저장하기 위한 메모리 사용량 또한 증가하여 모델을 원하는 태스크에 미세 조정하는 작업이 어려워지고 있다. 이러한 문제를 해소하기 위한 방편으로, 모델 내에서 일부 파라미터만 선별하거나, 모델에 새로 학습할 모듈을 추가하고 해당 모듈만 학습하는 등 모델 전체를 태스크에 맞게 재학습하는 기존 미세 조정 방식을 대체할 수 있는 기법들이 최근 제안되고 있다. 이처럼 모델 크기에 비해 적은 수의 파라미터만 학습해 모델을 미세 조정하는 것을 파라미터 효율적 전이 학습이라 한다.

다양한 파라미터 효율적 전이 학습 기법이 모델 전체 파라미터의 1% 수준의 파라미터만 학습해 모델 전체를 미세 조정하는 수준의 성능을 얻었다.[6, 7, 8, 9] 하지만, 이러한 기법들이 균일한 최종 학습 결과 등 사전학습 모델의 다른 성질도 보존하는지에 대한 분석 결과는 부족하다. 본 논문은 대표적인 파라미터 효율적 전이 학습 기법인 LoRA[7]로 모델을 학습한 결과 사이의 균질성을 확인하고, 모델 자체를 미세 조정한 결과 사이 균질성과 비교한다.

본 논문은 두 가지 기준을 통해 학습 결과의 유사성을 점검하였다. 우선, 사전학습 모델을 기반으로 학습을 마친 후 동일 입력을 각 학습 모델에 넣고, 사전학습 모델의 마지막 층에서 출력된 표현의 유사도를 비교하였다. 그 결과, 사전학습 모델을 직접 학습한 것에 비해서는 유사성이 감소하였으나, 학습 결과가 서로 유사하였고, 필요 파라미터 수가 낮은 데이터셋에서 모델의 학습 결과 간 유사성이 더 컸다. 또한, 학습 파라미터 수가 다른 경우에도 높은 유사도를 얻을 수 있었다.

그 다음, 파라미터 효율적 전이 학습 결과 간에 linear mode connectivity가 나타나는지 확인하였다. 그 결과, 파라미터 효율적 학습 시에도 데이터셋이 같다면 학습 파라미터 수나 학습 seed에 관계없이 linear mode connectivity를 찾을 수 있음을 확인하였다. 데이터셋에 관계없이 파라미터 수가 매우 낮으면 다른 손실 증가 폭이 커졌고, 학습에 필요한 파라미터 수가 큰 데이터셋에서 일부 경우에 손실 장벽이 나타났다. 데이터셋에 관계 없이 학습 파라미터 수가 다른 모델들이 linear mode connectivity를 만족할 수 있었다.

파라미터 효율적 학습의 유사성을 측정한 이후, 파라미터 효율적 전이 학습에서 학습 결과 간 유사성을 기반으로 한 앙상블을 시도하였다. [10]은 사전학습 모델을 여러 번 미세 조정한 결과를 모아 평균 가중치를 가지는 단일 모델을 만들어 보았고, 해당 모델을 학습 태스크 및 유사 태스크에 적용했을 때 각각의 단일 학습 결과보다 높은 성능을 얻을 수 있었다. 본 연구는 LoRA 학습 결과에도 이러한 방식을 적용해보고, LoRA 파라미터의 가중합을 통해 학습 태스크에서 손실이 더 낮은 가중치를 얻을 수 있음을 보였다.

## 1.2 논문의 구성

본 논문은 다음과 같이 구성되었다. 2장에서 본 연구와 관련된 사전학습 모델의 성질 및 그 성질의 응용, 파라미터 효율적 전이 학습 기법 등에 대한 기존 연구들을 소개하고, 3장에서는 본 논문에서 다룰 학습 기법인 LoRA에 대해 더욱 자세히 소개한다. 그 다음, 4장에서는 실험을 통해 파라미터 효율적 전이 학습 기법 사용 시 사전학습 모델의 학습 결과 간 유사성의 변화를 살펴보고 학습 결과의 유사성을 기반으로 한 모델 앙상블 기법을 파라미터 효율적 학습에 적용한다. 마지막 5장에서는 앞선 실험 결과를 통해 결론을 내리고 추후 이러한 실험 결과를 바탕으로 진행할 수 있는 연구 방향에 대해 이야기한다.

# 제 2 장 관련 연구

## 2.1 사전학습 모델 기반 학습 결과의 유사성

모델을 원하는 응용 태스크에서 학습하기 이전에 연관된 태스크에서 다수의 데이터로 사전학습하고, 이를 바탕으로 응용 태스크에서 모델을 다시 조정하는 방법은 자연 언어 처리에 국한되지 않고 다양한 분야에서 좋은 효과를 보였다. 이에 따라 사전학습 후 미세 조정한 모델과 해당 태스크에서만 학습한 모델의 차이점에 대한 연구도 다수 발표되었고, 사전학습이 이후 응용 태스크에서의 학습 결과를 유사하게 만든다는 사실이 확인되었다.

[11, 12, 13]는 학습 데이터 순서에 차이를 두고 같은 태스크를 학습한 동일 구조의 두 네트워크의 가중치를 비교하였고, 중간에 오차가 급증하는 구간 없이 서로 다른 두 네트워크를 연결하는 곡선 경로를 찾았다. 이러한 경로의 존재성을 mode connectivity라 하고, 두 모델 사이를 잇는 경로를 직선으로 제한한 것이 linear mode connectivity이다. [3]은 임의로 초기화된 일반적인 모델과 사전학습 모델을 분석하였고, 사전학습 모델을 미세 조정한 경우에만 linear mode connectivity를 만족한다는 것을 확인하였다. 이를 통해 [3]는 손실 landscape 상에 모델을 나타낼 때, 사전학습 모델이 미세 조정 이후에 공통의 넓은 분지에 도달한다고 주장하였다. 자연 언어 처리 분야에서는 [4]가 BERT를 텍스트 분류 태스크에서 미세 조정한 모델 다수의 linear mode connectivity를 분석하였고, 조정을 마친 모델들이 정확히 하나의 분지를 이루지는 않지만 적은 수의 특정한 분지 중 하나에 도달한다는 사실을 확인하였다.

사전학습 모델 기반 학습은 공통 입력에 대한 모델의 출력 표현을 유사하게 만드는 효과도 가진다. [3]은 사전학습 가중치 기반 미세 조정 모델과 주어진 태

스크에서만 학습한 모델의 출력 표현 사이 유사성을 비교하였다. 그 결과, 같은 입력이 주어질 때 사전학습 결과를 기반으로 학습한 모델들은 서로 유사한 표현을 출력했으며, 사전학습하지 않은 모델은 학습 이후 출력의 유사도가 매우 낮았다.

## 2.2 파라미터 효율적 전이 학습

파라미터 효율적 전이 학습은 사전학습 언어 모델을 미세 조정할 때 모델 전체 파라미터를 수정하는 대신에, 모델에 비해 매우 적은 수의 파라미터를 학습하는 방식을 말한다. 다양한 파라미터 효율적 전이 학습 기법들이 모델 전체를 미세 조정하는 수준의 성능을 얻었다. [6]는 매 트랜스포머 블록 내의 각 잔차 연결 직전에 학습을 위한 작은 모듈을 직렬로 연결하고, 추가시킨 모듈만 학습시킨다. [7]는 모델 내에 존재하는 기 학습된 파라미터에 낮은 차수를 갖는 추가 파라미터를 병렬적으로 연결하고, 추가된 파라미터를 학습시킨다. [8]는 모델을 그대로 유지하고, 특정 태스크의 입력 시퀀스 앞에 붙는 가상의 토큰을 학습시키는 각 층 별로 방식으로 모델을 미세 조정한다. [9]은 모델 내 각 파라미터의 가중치는 유지하면서 편향 값만 학습한다. [14]는 모델 내부의 각 활성화 층마다 요소별로 곱해줄 벡터를 학습하고, 이를 통해 모델의 활성화 값만 변경한다.

본 논문에서는 [7]에서 제안한 LoRA를 사용한다. LoRA를 선택한 이유는 크게 세 가지이다. 우선, LoRA는 대표적인 파라미터 효율적 전이 학습 기법으로, 파라미터 효율적 학습을 다루는 논문 전반에서 널리 쓰이고 있다.[15, 16, 17] 두 번째로, 미세 조정 수준의 학습 결과를 얻을 수 있는 서로 다른 파라미터 효율적 학습 기법이 서로 유사하다. [15]는 LoRA를 포함해 미세 조정 수준의 성능을 보인 파라미터 효율적 학습 방법들이 공통 구성 요소를 가지고 있음을 보였다. [16]는 학습 데이터가 충분할 경우 이 학습 기법들을 같이 쓰더라도 단일 기법 대비 성능 향상이 적다는 것을 보였다. 그러므로, 대표적인 단일 기법을 분석한 결과가 다른 기법에도 적용될 것으로 기대할 수 있다. 마지막으로, LoRA를 위해 추가한 각

학습 가중치는 해당 가중치와 병렬 연결된 기존 가중치에 학습 가중치를 더해주는 방식으로 병합할 수 있어 추가 파라미터 양이 서로 다른 두 모델의 파라미터를 비교하기 쉽다. [5]는 본 논문과 독립적으로 사전학습 모델의 미세 조정 및 파라미터 효율적 학습 기법의 일종인 어댑터[6]로 학습할 때 나타나는 linear mode connectivity의 변화를 분석했다. 본 논문은 LoRA 학습 결과를 분석했으며, 학습 결과를 결정하는 주 요인인 학습 파라미터 수에 따른 차이를 비교해 학습 파라미터 수가 서로 다른 경우에도 학습 결과가 공통의 분지로 모인다는 사실을 밝혔다.

## 2.3 모델 앙상블

앙상블 기법은 같은 태스크에 적용할 수 있는 서로 다른 모델이 여럿 있을 때, 각각의 모델들을 조합하여 각각의 단일 모델보다 더 좋은 모델을 만드는 것을 말한다. 가장 일반적인 앙상블 기법은 로짓 기반 앙상블으로, 각각의 모델에서 주어진 입력에 대한 출력 예측 결과를 계산하고, 각 예측 결과를 합산해 최종 예측 값을 결정한다. 이러한 방식은 앙상블하는 모델의 수만큼 추론 과정에서 계산 비용이 증가하고 시간이 오래 걸린다는 단점이 있다. 특히, 최근 모델이 거대해지면서 추론을 여러 번 거치는 것에 대한 부담이 더욱 커졌다.

사전학습 언어 모델을 미세 조정정했다면, 더 단순한 앙상블 기법을 사용할 수 있다. [3]은 사전학습 가중치를 기반으로 학습한 모델의 linear mode connectivity 등의 특성을 확인하며 이를 기반으로 한 개선된 앙상블 기법의 가능성을 제안하였고, [10]은 사전학습 모델의 성질에 기반한 앙상블 기법인 모델 수프를 고안하였다. 모델 수프는 단일 모델을 다양한 하이퍼파라미터에서 학습하고, 학습한 가중치들의 평균 가중치를 갖고 구조 및 크기가 같은 모델을 만든다. 이러한 앙상블은 주어진 입력을 각 모델에 넣고 출력한 로짓 값을 앙상블하는 일반적인 앙상블 방식과 성능이 유사하고, 모델 크기 증가 없이 출력을 한 번만 계산하기 때문에 로짓 기반 앙상블에 비해 계산량은 적다.

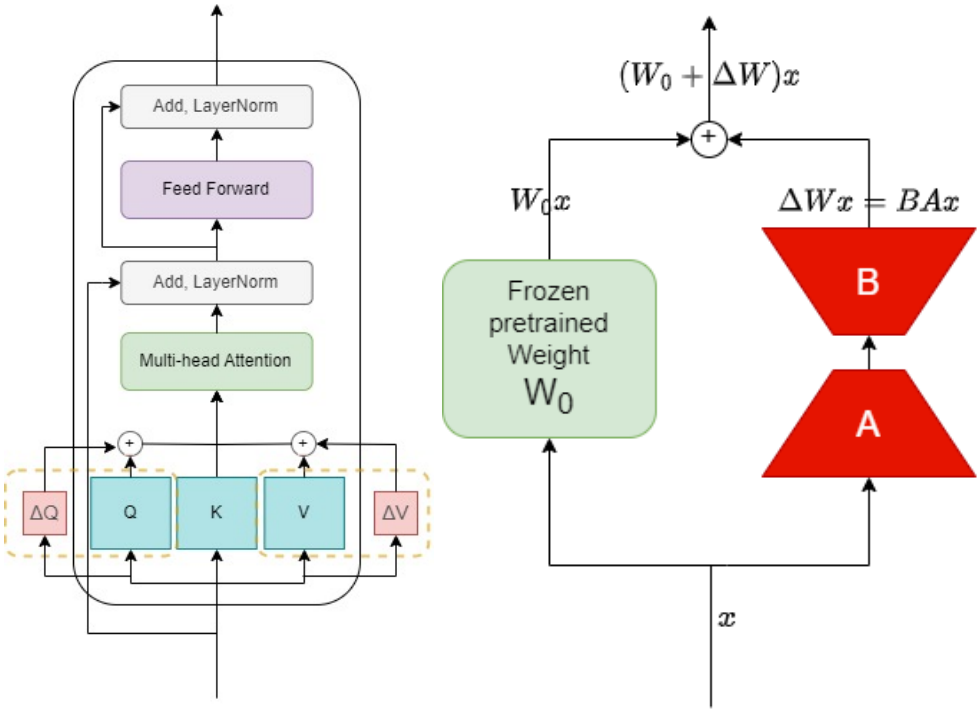
# 제 3 장 배경 지식

## 3.1 LoRA

[6]을 시작으로 모델의 미세 조정 수준의 학습이 가능한 다양한 파라미터 효율적 전이 학습 기법이 제안되었다. 본 연구는 그 중 LoRA[7]를 사용한다. LoRA는 학습할 모델 내에 있는 파라미터를 모두 학습하는 대신, 일부 학습 파라미터를 선택한다. 그 후, 선택한 파라미터를 직접 학습하지 않고, 해당 파라미터의 역할을 대신 수행하는 파라미터를 해당 파라미터에 병렬적으로 연결한다. 각 파라미터에 연결된 LoRA 모듈은 차원 축소 행렬 A와 차원 확대 행렬 B로 구성되며, 추가 모듈과 기존 파라미터의 입출력 크기는 동일하지만 투사 과정을 통해 전체 학습 파라미터 수를 줄인다. 이 때, 기존 임베딩을 투사시키는 차원을 LoRA의 계수라고 한다. LoRA 모듈을 추가했을 때 모델이 입력을 처리하는 과정은 그림 1와 같다.

트랜스포머 구조의 모델은 여러 개의 트랜스포머 블록으로 이루어져 있고, 각각의 트랜스포머 블록은 multi-head attention 모듈과 다층 퍼셉트론 층을 포함한다. LoRA는 그림 1 (a)와 같이 블록 내 attention 모듈 중 쿼리 값 행렬만 학습에 활용하였고, 다수의 태스크 및 모델을 LoRA로 학습해 모델 전체를 미세 조정한 수준의 결과를 얻었다. 본 논문에서도 항상 각 모델의 attention 모듈에 있는 쿼리 값 행렬에 LoRA를 연결한다.

LoRA를 학습시킬 경우 학습 파라미터의 수는 모델 크기에 비해 매우 작지만, 모델의 학습 성능은 그대로 유지된다. [7]는 다양한 데이터셋 및 모델에서 LoRA의 성능을 측정하였고, 전체 모델의 1% 수준의 학습 파라미터만 사용해 미세 조정과 동등한 수준의 성능을 얻었다.




(a) 트랜스포머 블록 및 LoRA 모듈의 구조 (b) LoRA의 입력 처리 방식

[그림 1] LoRA의 예시

# 제 4 장 실험 및 분석

## 4.1 데이터셋 및 모델

파라미터 효율적 전이 학습 기법으로 학습할 경우 모델을 미세 조정한 수준의 학습 성능을 얻기 위해 필요한 학습되는 파라미터의 수는 학습 상황마다 다르다. 이 때 학습 기법, 사전학습 언어 모델의 종류 외에도 학습하는 데이터셋의 종류에 따라 학습에 필요한 파라미터의 수가 변한다. [15]에 따르면, MNLI, SST-2 등의 텍스트 분류 태스크에서는 모델 크기의 1% 미만의 파라미터로 미세 조정과 동등한 학습 결과를 얻을 수 있다. 반면에, 문서 요약이나 번역 등의 태스크에서는 동일한 학습 기법을 썼을 때 10%가 넘는 파라미터를 써도 미세 조정 수준의 학습이 되지 않는다.

본 논문은 학습 파라미터 양에 따른 학습 결과의 변화를 관찰하기 위해 파라미터 요구량이 적은 텍스트 분류 태스크와 요구량이 많은 문서 번역 태스크에서 각각 실험을 진행하였다. 텍스트 분류와 문서 번역 문제에 적합한 사전학습 모델 구조가 서로 다르므로, 각 실험 별로 적절한 구조의 모델을 선택하였다.

### 4.1.1 QQP

| **Dataset** | **Train** | **Valid** | **Test** |
|---|---|---|---|
| QQP | 363846 | 40430 | 390965 |
| En-Ro | 610320 | 1999 | 1999 |

[표 1] 실험에 쓰인 각 데이터셋의 데이터 개수

[7]를 비롯한 다양한 파라미터 효율적 전이 학습 기법들은 자연어 이해 태스크에서 적은 파라미터만 학습해 모델 전체를 미세 조정하는 수준의 효과를 냈다.

| **Question1** | **Question2** | **is_duplicate** |
|---|---|---|
| How is the life of a math student? Could you describe your own experiences? | Which level of prepration is enough for the exam jlpt5? | 0 |
| How do I control my horny emotions? | How do you control your horniness? | 1 |
| What causes stool color to change to yellow? | What can cause stool to come out as little balls? | 0 |
| What can one do after MBBS? | What do i do after my MBBS ? | 1 |

[표 2] QQP 데이터 예시

본 논문에서는 모델 크기 대비 낮은 학습 파라미터 양으로 학습할 수 있는 데이터셋으로 대표적인 자연어 이해 성능 벤치마크인 GLUE[18]에 포함된 데이터셋인 QQP[19]를 사용하였다. QQP는 주어진 두 개의 질문이 정확히 같은 내용을 담고 있는지 판단하는 태스크를 다루며, 표 2을 통해 예시를 확인할 수 있다. 표 1에 기재된 대로, QQP는 36만개 이상의 학습 데이터와 39만개 이상의 테스트 데이터 등 다수의 데이터를 가지고 있다.

자연어 이해 태스크는 일반적으로 인코더 구조의 사전학습 언어 모델을 이용해 좋은 결과를 얻을 수 있다. QQP 학습은 대표적인 인코더 구조 사전학습 언어 모델인 BERT-base 모델을 사용하였고, Huggingface 라이브러리[20]에서 제공하는 사전학습 가중치로 모델을 초기화하였다. LoRA 학습 시 [7]에서 제안한 대로 모델의 각 트랜스포머 블록에 추가 파라미터를 연결하였고, 각 블록에서는 attention 층 중 쿼리, 값 행렬에만 LoRA 파라미터를 연결하였다. LoRA의 배율은 4로 고정하였고, 계수는 [1, 2, 4, 8, 16, 32] 범위 내에서 선택하였다. 사전학습 모델을 태스크에 맞게 학습하려면 학습된 모델의 출력을 기반으로 적절한 레이블을 예측하기 위한 분류기를 모델 뒤에 추가해야 한다. 본 실험은 LoRA의 특성을 확인하기

위한 것이므로 분류기는 하나의 선형층으로 이루어진 단순한 구조를 사용했다.

GLUE 벤치마크는 각 데이터셋 별 성능 측정 기준을 지정해놓았다. 그 중 QQP에서의 학습 성능은 정확도와 F1 지표를 통해 측정한다. 본 논문에서도 GLUE의 공식적인 평가 기준을 따라 정확도와 F1 지표를 기준으로 모델을 평가한다. 정확도는 모델의 전체 예측 중 올바른 예측을 한 비율이다. F1 지표는 식 4.1로 계산할 수 있다. recall은 재현율로, 실제 중복인 문장 중 모델도 중복으로 예측한 것의 비율을 말한다. precision은 모델의 정밀도를 뜻하며, 모델이 중복으로 예측한 데이터 중 실제로도도 중복인 데이터의 비율이다.

$$\mathrm{F1} = 2 * \frac{precision * recall}{precision + recall} \tag{4.1}$$

### 4.1.2 WMT16 En-Ro

[15, 21]는 문서 번역 태스크에 파라미터 효율적 전이 학습 기법을 적용할 경우 다른 태스크에 비해 더 많은 학습 파라미터를 추가해야 좋은 성능을 얻을 수 있음을 보였다. 본 논문에서는 대표적인 기계 번역 데이터셋인 WMT16[22] 중 En-Ro 데이터셋을 사용하였다. 해당 데이터셋의 각 데이터는 뜻이 같은 루마니아어, 영어 문장 쌍으로 구성되어 있으며, 이를 통해 모델이 주어진 영어 문장을 루마니아어로 번역하도록 학습한다. En-Ro 데이터셋의 예시는 3에서 확인할 수 있다. 학습 데이터셋에는 610320 개의 학습 데이터가 포함되어 있고, 검증셋과 시험셋에는 각각 1999개의 데이터가 있다.

인코더-디코더 구조의 모델은 입력된 텍스트를 적절한 임베딩으로 나타내고, 이를 활용해 번역한 문장을 출력할 수 있어 문서 번역에 적합하다. 번역 태스크를 위한 사전학습 모델은 영어, 루마니아어를 포함한 다양한 언어에 대해 사전학습되어 있고 인코더-디코더 구조를 이루고 있는 MBART-Large-CC25를 사용하였다. 사전학습 가중치는 BERT와 마찬가지로 Huggingface 라이브러리[20]에 저장된 값을 사용하였다. BERT와 마찬가지로 모델 끝에 연결해 출력을 단어로 변환하는

역할을 하는 모듈은 단순한 선형층 하나만 사용하였다.

[7]는 LoRA를 제안할 때 인코더-디코더 구조의 모델이 필요한 태스크에서 실험하지 않았고, 인코더와 디코더 블록 사이 cross attention에 LoRA를 추가해야 하는지 명시하지 않았다. 본 논문에서는 [15]을 따라 인코더 및 디코더 블록 내의 self attention 및 cross attention 모두 LoRA를 연결하였다. LoRA의 배율은 4, 계수는 [1, 4, 16, 64, 256]에서 선택하였고, 그 외의 학습 하이퍼파라미터는 [15]의 실험 설정을 따라 정하였다.

En-Ro 학습 결과는 BLEU 지표를 통해 평가한다. BLEU는 언어 모델이 생성한 문장을 평가할 때 가장 널리 쓰이는 성능 지표로, 모델이 생성한 문장이 정답 문장 안에 얼마나 포함되어 있는지를 평가한다. BLEU의 계산 과정은 식 4.2와 같다.

$$\text{BLEU} = min(1, \frac{output\_length}{reference\_length})(\prod_{i=1}^{4} precision_i)^{\frac{1}{4}} \tag{4.2}$$

| **Romanian** | **English** |
|---|---|
| Depunere de documente: a se vedea procesul-verbal | Documents received: see Minutes |
| (Die Sitzung wird um 15.25 Uhr unterbrochen und um 18.00 Uhr wiederaufgenommen). | (The sitting was suspended at 3.25 p.m. and resumed at 6.00 p.m.) |
| Ridicarea şedinţei | Closure of sitting |
| 6. | 6. |
| Dichiaro interrotta la sessione del Parlamento europeo. | I declare the session of the European Parliament adjourned. |

[표 3] WMT16 Romanian-English 데이터 예시

## 4.2 균질성 분석 지표

### 4.2.1 Convexity Gap

Convexity gap은 두 모델 사이의 linear mode connectivity 측정을 위한 지표이다. Linear mode connectivity의 일반적인 평가 과정은 다음과 같다. 우선, 평가하려는 모델을 같은 데이터셋에 대해 입력 순서를 바꿔가며 반복 학습하고, 그 결과로 얻은 가중치를 저장한다. 이제 가중치를 기준으로 두 모델을 연결하는 직선 경로를 동일 간격으로 나누어 두 모델을 선형 보간한 가중치 N개를 얻고, 학습을 통해 얻었던 두 가중치와 보간한 가중치들의 성능을 비교해 보간 중 생긴 오차 증가 폭을 측정한다. 이러한 오차 증가 폭을 오차 장벽이라 한다.

기존 연구들은 변동 폭을 계산하기 위해 다양한 평가 기준을 고안하였다. 서로 다른 순서로 학습한 두 모델의 가중치를 $\theta_1, \theta_2$, 모델 $\theta$의 error를 $\varepsilon(\theta)$라고 하자. [23]은 각 선형 보간 가중치의 오차를 보간에 쓰인 두 학습된 모델의 시험셋에 대한 오차의 평균과 비교하였다:

$$\mathrm{E}(W_1, W_2) = \sup_{\alpha} \varepsilon\left(\alpha W_1 + (1-\alpha) W_2\right) - \frac{\varepsilon\left(W_1\right) + \varepsilon\left(W_2\right)}{2} \quad \alpha \in [0,1] \tag{4.3}$$

[24]은 모델의 손실 $L(\theta)$를 기준으로 하고, 보간에 쓰인 실제 학습 결과인 $L(\theta_1)$과 $L(\theta_2)$ 사이에 처음부터 존재하는 차이를 고려해 아래의 기준을 제시하였다:

$$\mathrm{BH}(\theta_1, \theta_2) = \sup_{\alpha} \left[L\left(\alpha\theta_1 + (1-\alpha)\theta_2\right) - \left(\alpha \mathrm{L}\left(\theta_1\right) + \mathrm{L}\left((1-\alpha)\theta_2\right)\right)\right] \quad \alpha \in [0,1] \tag{4.4}$$

[4]는 [24]의 계산 방식을 더 일반화한 convexity gap을 제안하였다. Convexity gap은 선형 보간된 경로 상의 모든 부분 경로에서 가중치 쌍 사이의 손실을 비교하고, 그 중 최대 오차 상승 폭을 지표로 삼는다:

$$\mathrm{CG}(\theta_1, \theta_2) = \underset{\gamma,\beta}{sup}\, \mathrm{BH}(\gamma\theta_1 + ((1-\gamma)\theta_2), \beta\theta_1 + ((1-\beta)\theta_2)) \quad \beta, \gamma \in [0,1] \tag{4.5}$$

본 논문에서는 convexity gap을 통해 두 모델의 linear mode connectivity를 판별한다. 이는 convexity gap이 두 학습 결과 사이의 오차 변화를 가장 일반적으로 확인할 수 있는 지표이며, 과거 연구에서도 convexity gap을 통한 분석이 더 유의미한 결과를 얻을 수 있었기 때문이다. [4]는 특정 태스크에서 학습한 모델을 유사 태스크에 적용할 때 손실 분지에 따른 모델 성능 차이를 구별하였다. 이 때, convexity gap을 기준으로 구분했을 때 [24]를 사용할 때보다 모델의 성능을 더 잘 대변하는 결과를 얻었다.

보간의 대상인 서로 다른 두 LoRA 파라미터에 대해 각 모듈을 연결한 기존 파라미터를 $W_1, W_2$, 모듈 내 down projection 행렬을 $A_1, A_2$, up projection 행렬을 $B_1, B_2$, 그리고 LoRA 가중치에 곱해져 출력의 배율을 조정하는 상수를 $s_1, s_2$라 하자. 이 때, 입력 x에 대한 각 LoRA 파라미터의 출력은 $s_1B_1A_1x, s_2B_2A_2x$이다. 두 LoRA 가중치를 계수 $\alpha$로 보간하는 방법은 두 가지가 있다. 하나는 down projection 행렬을 $\alpha\sqrt{s_1}A_1+(1-\alpha)\sqrt{s_2}A_2$, up projection 행렬을 $\alpha\sqrt{s_1}B_1+(1-\alpha)\sqrt{s_2}B_2$으로 각각 보간하는 방식이고, 다른 하나는 $\alpha s_1B_1A_1 + (1-\alpha)s_2B_2A_2$ 형태로 한 번에 보간하는 방식이다. 본 논문에서는 후자를 선택하였다. LoRA는 두 개의 층으로 이루어져 있지만, 이는 기존 파라미터를 낮은 차원에서 학습하려는 목적을 위한 것일 뿐이고 down projection 직후의 출력은 따로 사용되지 않는다. 그러므로 $B_1A_1$, $B_2A_2$의 값만이 중요하고, $B_1$과 $B_2$, $A_1$과 $A_2$의 유사성 은 불필요하다. 이들 사이의 유사도와 $B_1A_1$, $B_2A_2$의 유사도 사이에 직접적인 관련성이 없으므로 전자의 분석 방식은 결과를 왜곡할 수 있고, 후자의 방식이 두 학습 결과 사이 비교에 더 적절하다.

### 4.2.2 Centered Kernel Alignment

모델의 출력 표현을 비교하는 작업은 centered kernel alignment (CKA)[25]를 이용하였다. CKA로 서로 다른 모델을 비교하는 과정은 다음과 같다. 우선, 학습 데이터셋에서 N개의 입력을 고르고, 기 학습된 서로 다른 두 LoRA에 선택된

입력을 입력했을 때 모델의 마지막 층에서 나오는 출력 임베딩을 각각 계산한다. 각각의 출력 임베딩은 입력 토큰 단위로 만들어지며, 각 토큰의 평균을 통해 모델 출력에서 전체 문장을 표현하는 임베딩을 얻는다. 자연어로 작성된 입력을 모델에 입력할 때에는 입력 문장을 토큰 단위로 잘게 나누며, 이러한 토큰화 과정에서 원래 문장에는 없지만 모델 학습을 돕기 위한 특수 토큰을 추가한다. 평균 계산시 이러한 특수 토큰의 임베딩은 제외하고, 실제 문장의 정보를 담고 있는 임베딩만 사용한다. 모델 X의 $l$번째 층에서 N개의 입력에 대해 위의 방법대로 계산된 문장 임베딩을 모은 행렬을 $X_l$이라 하면, 두 모델 X, Y의 $k$, $l$번째 층에서 출력한 $X_k$와 $Y_l$의 CKA 계산식은 아래와 같다:

$$\mathrm{CKA}(X_k, Y_l) = \frac{\|X_k^T Y_l\|_F^2}{\|X_k^T X_k\|_F \|Y_l^T Y_l\|_F} \tag{4.6}$$

$\|X\|_F$ 는 X의 프로베니우스 노름이다. CKA는 0 이상 1 이하의 값을 가지며, 값이 클 수록 두 행렬의 유사도가 높다. CKA는 다른 유사도 평가 지표와 달리 모델의 출력 표현의 너비가 데이터셋의 크기보다 크더라도 적절한 유사도를 계산할 수 있는 평가 지표로, 모델의 크기가 크고 은닉 상태의 크기가 큰 사전학습 언어 모델을 평가하기에 적절하다.

CKA는 모델에서 다수의 입력에 대한 문장 임베딩을 모은 행렬을 필요로 한다. CKA 계산을 위해 각 태스크의 검증셋에서 512개의 데이터를 무작위로 선정하였고, 태스크 별 CKA 계산에 쓰인 데이터는 동일하다. 각 입력에 대한 임베딩 계산시 convexity gap 계산과 마찬가지로 기존 파라미터 및 추가 파라미터를 병합한 모델을 이용한다. 학습이 끝난 LoRA 모듈을 모델에 병합할 경우 모델의 학습 결과를 보존하면서 계산 시간을 줄일 수 있다. 사전학습 언어 모델의 마지막 층에서 출력된 입력에 대한 표현이 실제 모델의 예측 결과를 결정하므로 이 값을 기준으로 학습 모델 사이 유사도를 비교한다.

## 4.3 파라미터 효율적 학습 결과의 유사성

### 4.3.1 학습 성능

| | LoRA | | | | | | 미세조정 |
|---|---|---|---|---|---|---|---|
| | rank1 | rank2 | rank4 | rank8 | rank16 | rank32 | |
| param ratio (%) | 0.034 | 0.067 | 0.135 | 0.269 | 0.539 | 1.077 | 100 |
| Loss | 0.285 | 0.287 | 0.257 | 0.256 | 0.272 | 0.280 | 0.232 |
| Accuracy (%) | 87.9 | 87.3 | 89.0 | 89.1 | 88.3 | 88.0 | 90.6 |
| F1 (%) | 83.8 | 83.2 | 85.4 | 85.6 | 84.4 | 84.0 | 87.5 |

[표 4] QQP 데이터셋, 학습 기법 별 학습 파라미터 수 및 성능

| | LoRA | | | | | 미세조정 |
|---|---|---|---|---|---|---|
| | rank1 | rank4 | rank16 | rank64 | rank256 | |
| param ratio (%) | 0.024 | 0.097 | 0.386 | 1.545 | 6.180 | 100 |
| Loss | 3.318 | 6.439 | 3.013 | 2.95 | 2.945 | 4.106 |
| BLEU (%) | 26.44 | 30.4 | 33.66 | 35.71 | 36.11 | 37.38 |

[표 5] En-Ro 데이터셋, 학습 기법 별 학습 파라미터 수 및 성능

파라미터 효율적 전이 학습 결과의 균질성을 평가하기 전에, 사전학습 언어 모델 기반 학습 기법에 따른 성능을 비교하였다. 그 결과, LoRA를 사용하면 적은 수의 파라미터만으로 미세 조정과 유사한 수준의 학습 결과를 얻을 수 있었다. 학습에 필요한 파라미터의 수와 모델 크기 사이의 비율은 데이터셋 별로 달랐다.

표 4는 QQP 데이터셋에서 미세 조정 및 LoRA 등의 기법으로 BERT를 학습할 때 학습한 파라미터의 양과 그 검증 성능을 나타낸다. 1에서 8 사이의 계수를 갖는 LoRA 모듈로 학습할 때는 계수 증가에 따라 모델의 성능이 같이 증가했다. 하지만, 계수를 추가로 올릴 때에는 성능이 소폭 감소하였다. 모델 자체를 학습

시킨 결과와 비교하면, LoRA의 계수가 4에서 8 사이일 때 미세 조정과의 정확도 차이가 2% 이내로 나와 미세 조정과 유사한 수준의 학습이 가능함을 알 수 있었다.

표 5에서 볼 수 있듯이, En-Ro 데이터셋을 학습하기 위해서는 QQP에 비해 더 많은 학습 파라미터가 필요하다. 그 예로로, 전체 모델의 6% 수준의 파라미터를 학습할 때까지 과다한 학습 파라미터 사용으로 인한 성능의 감소가 없었다. En-Ro에서는 LoRA의 계수가 1일 때 미세 조정에 비해 성능이 크게 떨어졌으며, 1에서 256까지 계수가 증가하는 동안 모델의 성능이 꾸준히 개선되었다. 특히, LoRA 계수가 256일 때에는 미세 조정 모델과 BLEU 지표 차이가 1.27% 수준까지 좁혀져 En-Ro에서 학습 파라미터 수의 중요성을 파악할 수 있었다.

### 4.3.2 출력 표현 간 유사성

| | LoRA | | | | | | 미세조정 |
|---|---|---|---|---|---|---|---|
| | rank1 | rank2 | rank4 | rank8 | rank16 | rank32 | |
| param ratio (%) | 0.034 | 0.067 | 0.135 | 0.269 | 0.539 | 1.077 | 100 |
| CKA | 0.510 | 0.820 | 0.645 | 0.695 | 0.690 | 0.590 | 0.768 |

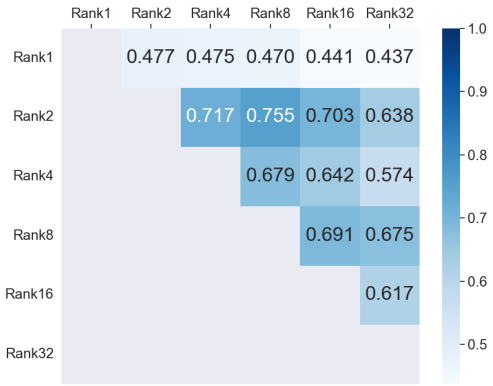
[표 6] QQP 데이터셋을 동일한 양의 학습 파라미터로 학습한 결과 사이 유사도

| | LoRA | | | | | 미세조정 |
|---|---|---|---|---|---|---|
| | rank1 | rank4 | rank16 | rank64 | rank256 | |
| param ratio (%) | 0.024 | 0.097 | 0.386 | 1.545 | 6.180 | 100 |
| CKA | 0.427 | 0.571 | 0.560 | 0.437 | 0.410 | 0.651 |

[표 7] En-Ro 데이터셋을 동일한 양의 학습 파라미터로 학습한 결과 사이 유사도

표 6는 QQP에서 학습한 미세 조정 및 동일 계수의 LoRA 학습 결과들을 CKA를 통해 비교하여 얻은 유사도를 나타낸다. 평균적으로 미세 조정의 85.7% 수준의 유사도를 유지하였고, 계수가 2인 경우 유사도가 미세 조정 결과 사이의 유사도인

0.765보다 높은 0.820으로 계산되었다.



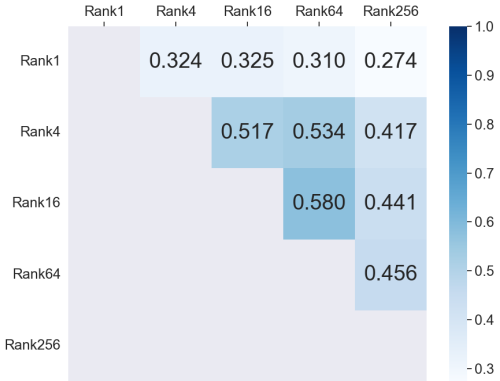

[그림 2] QQP, 서로 다른 계수의 LoRA로 학습한 가중치 사이 출력 유사도

서로 다른 계수의 LoRA 학습 결과 사이에서도 계수가 1인 경우를 제외하면 출력 표현의 CKA 값이 항상 0.574 이상으로 유지되었다. 그림 2는 서로 다른 계수의 LoRA 학습 가중치들 간 CKA 계산 결과를 나타낸 히트맵으로, 색이 진할 수록 둘 사이의 유사도가 높다. 실험 결과 계수가 1, 2일 때는 다른 계수 학습 결과와의 유사도가 동일 계수 학습 결과 내에서 측정한 유사도에 비해 항상 낮았다. 하지만, 계수 4 이상인 경우에는 인접한 계수의 학습 결과와의 유사도 값이 동일 계수 내 유사도 이상이다. 이를 통해 같은 순서로 입력을 넣어주면 LoRA의 학습 파라미터 수가 다르더라도 비슷한 학습 결과를 얻을수 있다는 것을 알 수 있다.

이제 En-Ro에서 측정한 표현 유사도를 다룬다. 표 7은 동일한 학습 파라미터 개수를 가진 미세 조정 및 LoRA 학습 모델들을 비교한 결과이다. 표를 통해 계수에 관계없이 LoRA 학습 결과 사이의 유사성이 미세 조정 결과에 비해 떨어지는 것을 확인할 수 있다. QQP와 달리 미세 조정보다 더 유사한 특징 벡터를 출력하는 경우는 찾지 못했고, 각 계수의 학습 모델 사이 유사도는 미세 조정 가중치 사이 유사도의 63.0  87.7%, 평균 73.9% 수준이었다. 특히, 학습 성능이 크게 개선되었던 64 이상의 계수에서 학습된 표현의 유사성이 더 많이 감소하였다. 이러한 현상은

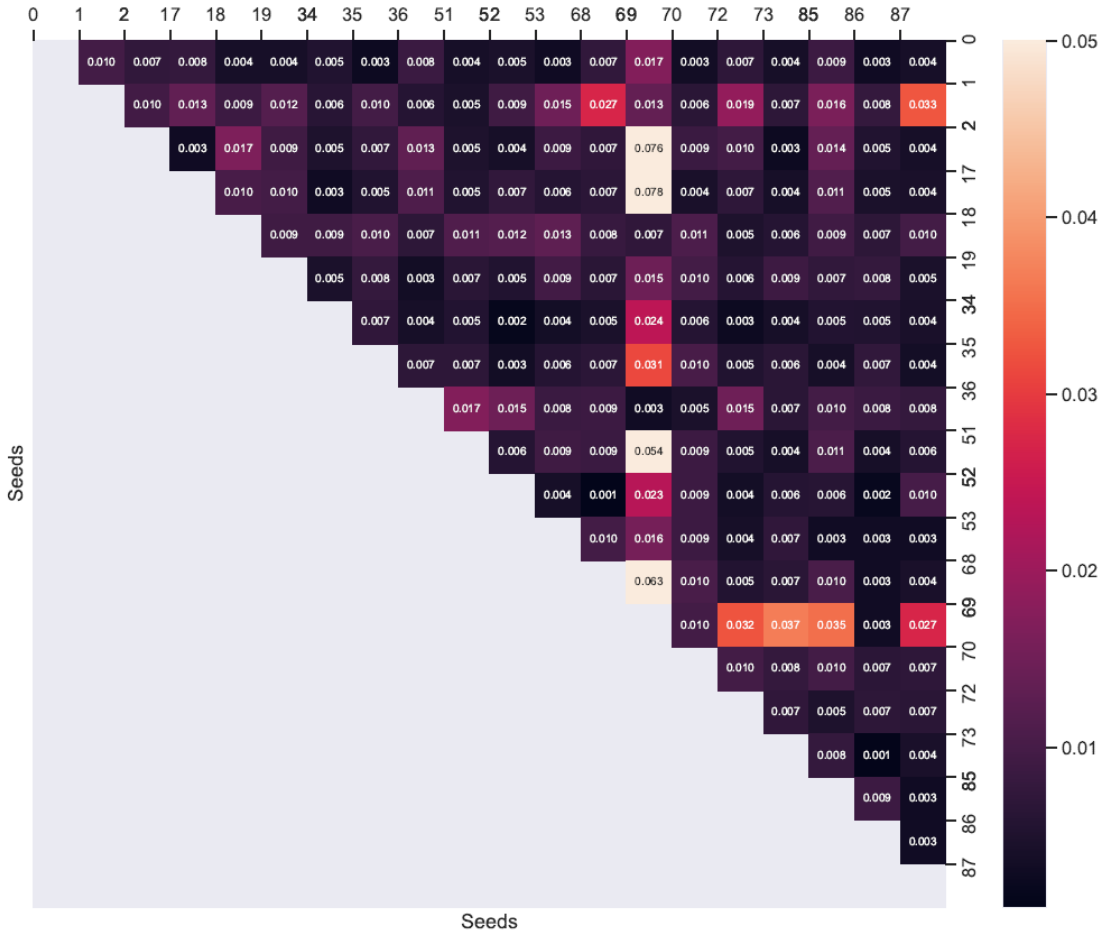
학습 수준의 문제가 아닌 LoRA 학습 기법 자체의 문제로 인해 모델 학습 결과 간 차이가 발생하였다는 것을 뜻한다.

그림 3는 같은 순서로 데이터를 학습한 서로 다른 계수의 LoRA 학습 결과를 비교한 것이다. 표 7와 그림 3를 통해 데이터 순서에 의한 출력의 차이가 학습 파라미터 수 차이보다 유사도를 더 많이 감소시키는 경우를 일부 확인할 수 있다.




[그림 3] En-Ro, 서로 다른 계수의 LoRA로 학습한 가중치 사이 출력 유사도

QQP, En-Ro에서 동일 입력에 대한 각 모델의 출력 사이 유사도 측정 결과를 비교하면, 두 데이터셋 모두 LoRA 학습 결과 간 유사도가 미세 조정 결과의 유사도에 비해 0.1 - 0.25 만큼 낮았으며, En-Ro 에서 유사도가 더 크게 감소했다. 또한, 유사도는 학습 성능과 무관하게 계수를 늘릴 수록 감소하였다. 이를 통해 LoRA 학습 방식이 사전학습 모델을 바탕으로 학습하는 것은 미세 조정과 동일하지만, LoRA로 학습할 경우 사전학습의 학습 결과를 균일화하는 효과가 약화된다는 것을 알 수 있다. 또한, 그림 2와 그림 3를 통해 서로 다른 계수의 학습 결과를 비교했을 때 데이터 입력 순서보다 학습 파라미터 수에 의한 차이가 더 적은 경우가 있음을 확인하였다.




[그림 4] QQP, 미세 조정된 BERT 사이 convexity gap.

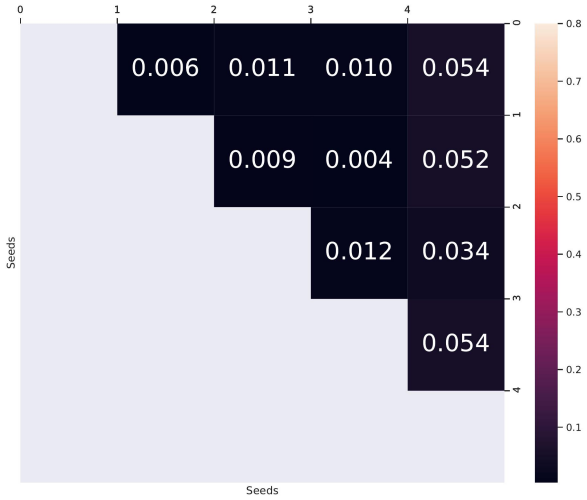
### 4.3.3 Linear Mode Connectivity

QQP에서 LoRA로 학습한 결과를 분석하기에 앞서, 사전학습했던 모델 자체를 미세 조정할 때 linear mode connectivity가 나타나는지 확인하였다. 그림 4은 BERT를 서로 다른 15개의 seed로 학습한 가중치들 사이에서 계산한 convexity gap을 히트맵으로 나타낸 것이다. X, Y축은 직선 경로 상의 양 끝 지점이 되는 두 모델의 학습 데이터 순서를 결정하는 seed를 나타내고, 색이 밝을 수록 두 모델 사이 직선 경로 상의 손실 장벽이 높다. 대부분의 모델은 직선 경로 상에서 0.02 미만의 손실 증가 폭을 가져 linear mode connectivity를 확인할 수 있었다. 일부 seed (1, 69) 만이 다른 학습 결과와 연결된 경로 상에서 다소 높은 오차 증가를 보였다. 이를 통해 사전학습된 BERT의 가중치가 학습 과정을 강하게 통제하고, 손실 landscape 상에서 미세 조정된 가중치의 최종 위치를 몇몇 안정된 분지 중 하나로 국한한다는 것을 확인할 수 있다.

앞선 실험에서 BERT 자체를 미세 조정할 경우 linear mode connectivity가 나타났다. 이제 파라미터 효율적 학습 기법인 LoRA로 학습했을 때에도 linear mode connectivity가 발생하는지 확인한다. 우선, QQP에서 학습 파라미터의 위치 및 구조가 동일한 동일 계수의 LoRA를 쓸때 서로 다른 순서로 입력을 넣어 학습한 가중치 사이 convexity gap을 계산하였다. 그 결과, 동일 계수 학습 결과들은 학습 데이터 입력 순서에 무관하게 linear mode connectivity를 유지하였다. 그림 6[1]는 각 계수 별 학습 모델 사이의 convexity gap을 나타낸다. 계산된 convexity gap은 항상 0.03 이하의 낮은 값을 보이며, 계수가 32인 경우를 제외하면 0.02 이하로 직선 경로 상 오차 상승폭이 모델 미세 조정 결과와 동일한 수준이었다.

학습 데이터의 입력 순서는 동일하지만, 학습에 쓰인 LoRA의 계수가 서로 다른 학습 가중치들 사이의 convexity gap도 측정하였다. 그 결과는 그림 7과 같다. 서로 다른 계수의 LoRA를 연결할 때, 학습 파라미터가 가장 적은 계수 1인 LoRA

[1]그림 6 - 그림 9는 4.3절 끝에 배치하였다.




[그림 5] En-Ro, 미세 조정된 MBART 사이 convexity gap.

를 제외하면 사전학습 모델의 미세 조정 수준의 linear mode connectivity가 항상 유지되었다. 이를 통해 QQP에서는 계수가 2 이상일 때 계수에 관계 없이 LoRA 학습 결과들이 가로막히지 않은 공통의 손실 분지로 향한다는 사실을 알 수 있다.

그림 5에서 볼 수 있듯이 MBART 모델은 모델 자체를 조정했을 때 사전학습에 의한 linear mode connectivity를 잘 드러낸다. 하지만 QQP와 달리, LoRA 학습 결과는 모델 미세 조정에 비해 악화된 linear mode connectivity를 보였다. 그림 8은 학습 데이터 순서가 다른 동일 계수의 LoRA 학습 결과를, 그림 9는 학습 데이터 순서가 같고 계수가 서로 다른 LoRA 학습 결과를 나타낸 것이다.

동일 LoRA 계수 기반 학습 결과끼리 비교했을 때, 계수에 관계없이 50% 이상은 서로를 잇는 경로 상에서 손실 증가 폭이 양 끝의 두 모델에서의 최대 손실 값의 10% 이하로 유지되었다. 이를 통해 파라미터 효율적 학습 시에도 학습 가중치가 공통 위치로 향할 수 있다는 것을 알 수 있다. 하지만, 미세 조정 결과와는 차이가 있었다. 모델의 미세 조정 결과에 비해 전반적으로 두 학습 결과 사이 직선 경로 상에서 손실 증가 폭이 높았고, 일부 학습 결과 사이에서 모델의 시험 손실보다 큰 폭으로 손실이 증가하는 지점이 발견되는 등 모델 자체 학습 결과에 비해 학습

결과가 잘 모이지 않았다.

데이터 입력 순서가 같지만, 계수가 다른 학습 결과 사이를 비교한 경우에도 유사한 현상이 나타났다. 그림 9 (a)처럼 모델 사이 큰 장벽이 없는 seed도 있었지만, 다른 seed에서는 모델 사이 평균적인 손실 장벽 및 최대 손실 장벽의 높이가 증가하는 결과를 보였다. 이러한 결과들을 통해 En-Ro 데이터셋에서는 LoRA를 통해 학습할 경우 linear mode connectivity를 만족하는 경우를 찾을 수 있지만, 사전학습 모델 자체를 미세 조정하는 방식에 비해서 그 빈도 및 오차 수준이 악화된다는 것을 알 수 있다.

## 4.4 가중치 기반 앙상블

앞선 절의 분석 결과는 다음 내용을 밝혔다. 동일 태스크에서 얻은 서로 다른 학습 가중치의 linear mode connectivity 및 출력 임베딩 간 유사도 측정 시, LoRA 학습 결과에서도 여전히 linear mode connectivity를 만족하거나 높은 유사도를 가지는 학습 결과들을 찾을 수 있다. 또한, 학습 파라미터 수가 서로 다른 모델 중에도 유사한 표현을 출력하며 linear mode connectivity를 만족하는 모델을 찾을 수 있다. 이번 절에서는 이러한 관찰 결과를 바탕으로 기존 사전학습 언어 모델을 미세 조정한 결과들을 앙상블할 때 사용했던 가중치 평균 기반 앙상블 방법이 파라미터 효율적 전이 학습에도 적용될 수 있는지 실험한다.

### 4.4.1 앙상블 방법

모델 수프[10]는 다양한 하이퍼파라미터로 학습한 여러 모델의 평균 가중치를 가진 단일 모델을 통해 다양한 모델을 앙상블하는 효과를 얻는다. 이 때 그 성능에 관계 없이 모든 학습 결과의 평균을 구하는 uniform soup, 검증셋에서 가장 성능이 높은 단일 모델에서 시작해 합쳤을 때 검증셋에서 성능이 더 높아지는 파라미터만 모으는 greedy soup 등의 방법을 제안했다. 본 논문은 greedy soup 방식으로 가중치를 병합한다. 또한, 모델 가중치들의 단순 평균을 구하는 대신, 탐욕적으로 찾은

최선의 가중치와 병합 대상이 되는 다른 가중치를 잇는 직선 경로를 동일 간격으로 네 번 나눠 보간한 가중치를 얻는다. 그 다음, 보간한 가중치 각각의 시험 손실을 측정하고, 그 중 가장 손실이 낮은 결과가 기존 병합된 모델보다 낮은 손실을 보일 경우 가중치를 수정한다. 이 과정을 통해 매 순간 가능한 조합 중 가장 성능이 높은 모델을 얻는다. 전체 알고리즘의 의사 코드는 Algorithm 1과 같다.

**Algorithm 1** Greedy Soup with Weight Interpolation

```
Input: Weights trained on a given task {θ_1, ..., θ_k}
                    ▷ Weights are ordered in an increasing order of loss
θ_g ← θ_1                                  ▷ Single model with the best loss
θ_new ← θ_1
l_best ← Loss(θ_g)                         ▷ Loss of the achieved model
for i ← 2 to k do
    for α ← 1 to 4 do
        θ' = (αθ_g + (5 − α)θ_i)/5
        if Loss(θ') < l_best then
            θ_new ← θ'
            l_best ← Loss(θ')
        end if
    end for
    θ_g ← θ_new
end for
Return θ_g                  ▷ Aggregated model with improved performance
```

파라미터 효율적 학습 시에 서로 다른 계수의 학습 결과 또한 공통 손실 분지로 향하는 것을 확인하였으므로, 다양한 계수의 학습 결과를 모두 활용하는 것이 최종 앙상블 결과에 도움을 줄 수 있다. 그러므로, greedy soup 계산 시 다른 순서로 학습한 동일 파라미터 수의 LoRA 학습 결과만 사용하지 않고 서로 다른 수의 파라미터를 학습한 결과들을 모두 앙상블에 이용하였다. [10]는 모델 수프에 사용할 미세 조정 결과를 얻기 전에 사전학습 모델 뒤에 연결하는 마지막 선형층을 동일하게 초기화한다. 하지만, 본 논문에서는 이러한 초기화 과정을 거치지 않고 일반적으로 초기화한 후 학습된 결과들을 조합하였다.

### 4.4.2 앙상블 결과

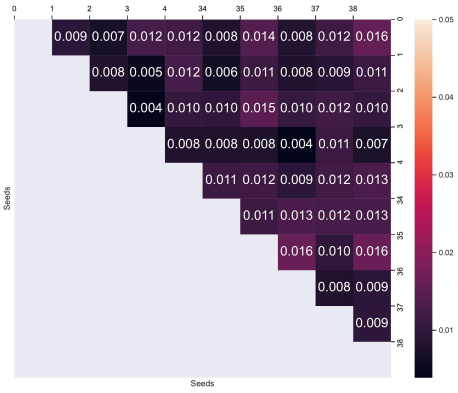
표 8와 표 9는 각각 QQP와 En-Ro에서 가중치를 병합한 앙상블 모델과 가장 낮은 손실을 보인 단일 모델을 비교한 결과를 나타낸다. QQP, En-Ro의 검증셋에서 더 낮은 손실을 갖는 모델을 찾도록 greedy soup 알고리즘을 적용한 결과, 각 모델은 시험셋에서 단일 모델보다 낮은 손실을 얻을 수 있었다. 하지만, 더 낮은 손실이 항상 다른 성능 지표의 향상으로 이어지지는 않았다. 모델 각 데이터셋의 성능 측정 기준으로 성능을 측정한 결과, En-Ro 데이터셋에서는 모델의 BLEU 지표가 0.7% 증가한 반면, QQP에서는 정확도가 0.6%, F1 지표가 0.2% 감소했다.

| | Single Model w/ Best Loss | Greedy Soup |
|---|---|---|
| Loss | 0.241 | **0.239** |
| Accuracy (%) | **90.0** | 89.4 |
| F1 (%) | **86.6** | 86.4 |

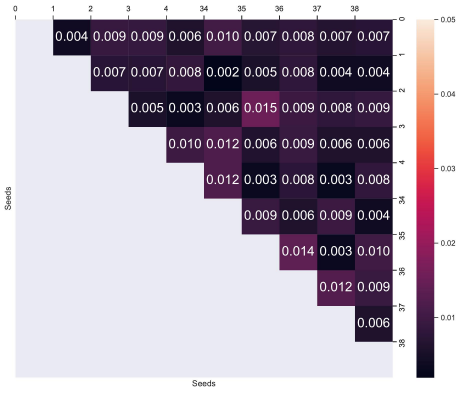
[표 8] QQP 데이터셋에서 학습한 가중치를 앙상블한 성능

| | Single Model w/ Best Loss | Greedy Soup |
|---|---|---|
| Loss | 2.984 | **2.970** |
| BLEU (%) | 35.9 | **36.6** |

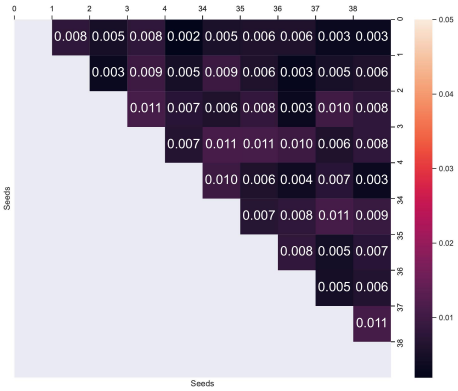
[표 9] En-Ro 데이터셋에서 학습한 가중치를 앙상블한 성능



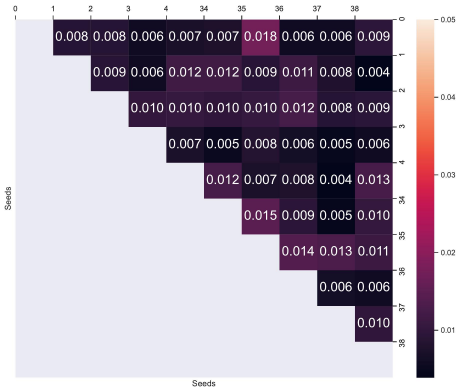

(a) Rank1




(b) Rank2

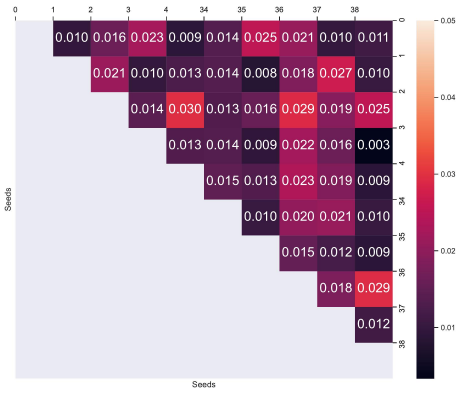



(c) Rank4




(d) Rank8


(e) Rank16




(f) Rank32

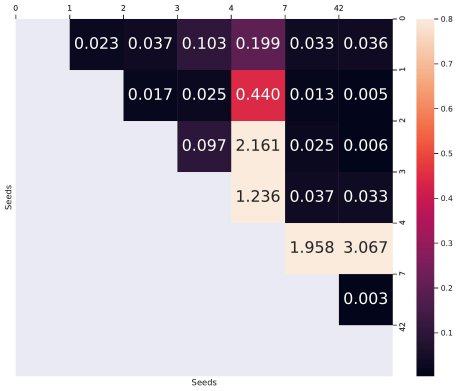
[그림 6] QQP에서 동일 계수의 LoRA 가중치 사이 convexity gap.


(a) Seed0 (b) Seed2

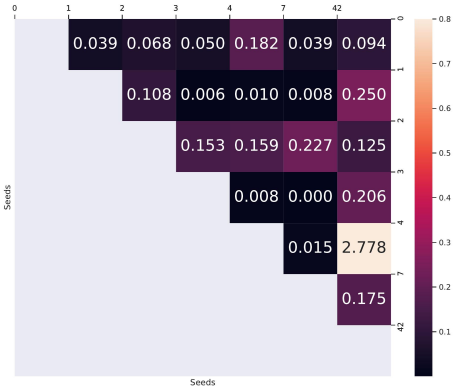
(c) Seed3 (d) Seed4

[그림 7] 동일 seed에서 서로 다른 계수의 LoRA를 QQP로 학습한 결과 사이 convexity gap.

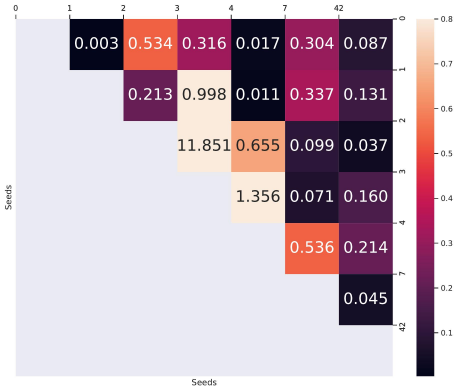

(a) Rank1



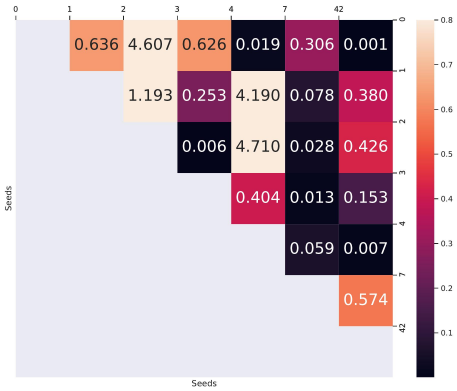

(b) Rank4




(c) Rank16

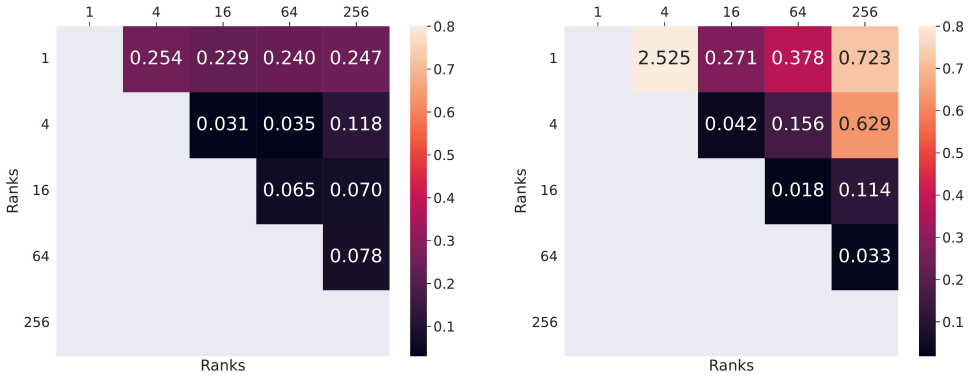



(d) Rank64




(e) Rank256

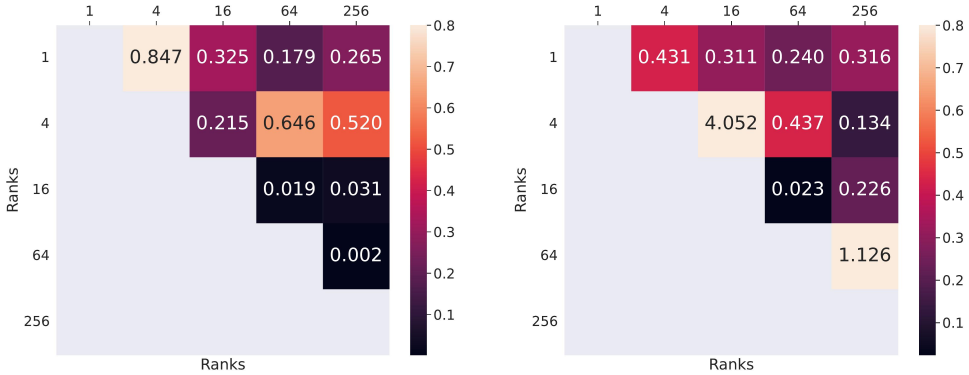
[그림 8] En-Ro, 7개 seed에 대한 동일 계수의 LoRA 학습 가중치 사이 convexity gap.




(a) Seed0



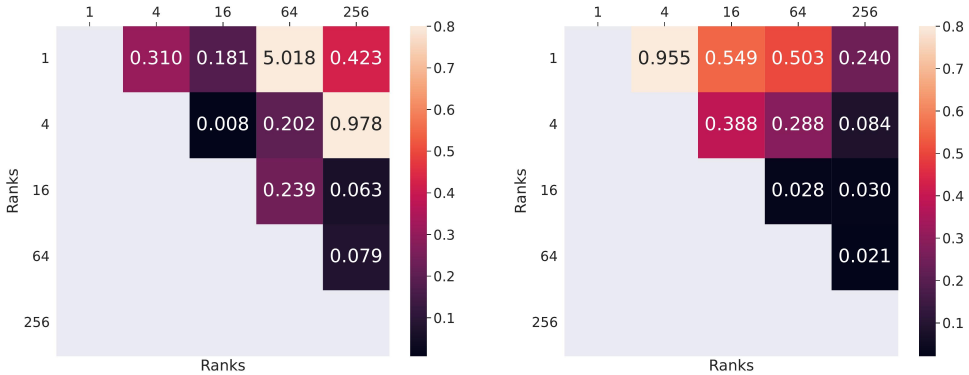

(b) Seed1




(c) Seed2




(d) Seed4




(e) Seed7




(f) Seed42

[그림 9] En-Ro, 서로 다른 계수의 LoRA 가중치 사이 convexity gap.

# 제 5 장 결론

## 5.1 결론

기존 연구는 사전학습한 모델을 응용 태스크에서 미세 조정하면 학습이 끝났을 때 데이터 입력 순서에 관계 없이 최종 모델이 균질해짐을 밝혔다. 본 논문에서는 파라미터 효율적 전이 학습 기법의 일종인 LoRA로 학습한 모델 다수를 각 모델 사이의 최종 표현 유사도 및 linear mode connectivity 측면에서 비교해, 파라미터 효율적 학습 시에도 사전학습 모델의 학습 결과가 균질한지 확인하였다. 그 결과, 데이터 입력 순서 및 학습 모듈 크기에 관계 없이 최종 표현이 유사하였고, linear mode connectivity를 만족하는 모델들을 얻을 수 있었다. 이러한 경향은 학습하기 위해 필요한 전체 모델 크기 대비 실제 학습 파라미터의 비율이 적은 태스크에서 더욱 뚜렷하게 나타났다.

기존 사전학습 모델에 대한 연구에서는 학습 결과 사이의 균질성을 기반으로 모델 가중치의 평균을 통한 앙상블 기법을 제안했다. 본 논문은 LoRA 학습 결과에 대한 분석 결과를 바탕으로 가중치 평균 앙상블 기법을 LoRA에 적용하였다. 서로 다른 학습 파라미터 수 및 입력 순서로 학습한 다수의 학습 가중치들을 모두 앙상블하였을 때, 추가 학습 과정이나 주어진 입력에 대한 추론 비용 증가 없이 학습 손실이 더 낮은 단일 모델을 얻을 수 있었으며, 일부 태스크에서는 손실 외 다른 성능 지표도 개선할 수 있었다.

## 5.2 향후 연구

본 연구에서는 적은 수의 파라미터로 모델을 학습한 결과가 더 많은 파라미터로 학습한 결과와 가깝다는 사실을 확인했다. 이를 바탕으로 적은 수의 파라미터만 학습한 결과를 조정해 더 많은 파라미터로 학습한 결과와 유사하게 만드는 것을

시도할 수 있다. 파라미터 효율적 전이 학습 기법을 쓸 경우 일반적으로 학습 파라미터 증가에 따라 성능이 증가하다가, 특정 지점 이후 유지되거나 감소하는 경향을 보인다. 파라미터 수가 부족하여 성능이 상대적으로 낮은 학습 가중치를 손실 분지 내에서 움직여 더 많은 파라미터의 학습 결과에 가깝게 만들 수 있다면, 실제 학습 과정에 필요한 파라미터 수 및 연산량을 낮출 수 있을 것이다.

인공 신경망 네트워크의 linear mode connectivity는 가지치기한 모델의 성능[23], 유사 태스크로의 일반화 경향 등과 큰 연관성이 있고[4], 다양한 가중치의 앙상블을 통한 성능 향상[10, 26], 기 학습된 정보를 잃지 않고 다른 데이터를 학습하는 지속 학습[27] 등 다양한 분야에서 활용되고 있다. 본 연구는 파라미터 효율적 학습 결과의 linear mode connectivity 만족 여부를 확인하는 데 집중하였고, 관련 기법을 다루지는 않았다. 후속 연구에서는 앞선 연구들의 mode connectivity에 기반한 기법들을 파라미터 효율적 전이 학습에도 적용할 수 있을 것이다.

# 참고문헌


[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[2] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022. Survey Certification.

[3] B. Neyshabur, H. Sedghi, and C. Zhang, "What is being transferred in transfer learning?," *Advances in neural information processing systems*, vol. 33, pp. 512–523, 2020.

[4] J. Juneja, R. Bansal, K. Cho, J. Sedoc, and N. Saphra, "Linear connectivity reveals generalization strategies," *arXiv preprint arXiv:2205.12411*, 2022.

[5] Y. Qin, C. Qian, J. Yi, W. Chen, Y. Lin, X. Han, Z. Liu, M. Sun, and J. Zhou, "Exploring mode connectivity for pre-trained language models," *arXiv preprint arXiv:2210.14102*, 2022.

[6] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient trans-

fer learning for nlp," in *International Conference on Machine Learning*, pp. 2790–2799, PMLR, 2019.

[7] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[8] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

[9] E. B. Zaken, Y. Goldberg, and S. Ravfogel, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

[10] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *International Conference on Machine Learning*, pp. 23965–23998, PMLR, 2022.

[11] C. D. Freeman and J. Bruna, "Topology and geometry of half-rectified network optimization," in *International Conference on Learning Representations*, 2017.

[12] F. Draxler, K. Veschgini, M. Salmhofer, and F. Hamprecht, "Essentially no barriers in neural network energy landscape," in *International conference on machine learning*, pp. 1309–1318, PMLR, 2018.

[13] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of dnns," *Advances in neural information processing systems*, vol. 31, 2018.

[14] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. Raffel, "Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning," *arXiv preprint arXiv:2205.05638*, 2022.

[15] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," in *International Conference on Learning Representations*, 2022.

[16] Y. Mao, L. Mathias, R. Hou, A. Almahairi, H. Ma, J. Han, S. Yih, and M. Khabsa, "Unipelt: A unified framework for parameter-efficient language model tuning," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6253–6264, 2022.

[17] Z. Yang, M. Ding, Y. Guo, Q. Lv, and J. Tang, "Parameter-efficient tuning makes a good classification head," *arXiv preprint arXiv:2210.16771*, 2022.

[18] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, 2018.

[19] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 4144–4150, 2017.

[20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, (Online), pp. 38–45, Association for Computational Linguistics, Oct. 2020.

[21] A. Üstün and A. C. Stickland, "When does parameter-efficient transfer learning work for machine translation?," *arXiv preprint arXiv:2205.11277*, 2022.

[22] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. Jimeno Yepes, P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Neveol, M. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor, and M. Zampieri, "Findings of the 2016 conference on machine translation," in *Proceedings of the First Conference on Machine Translation*, (Berlin, Germany), pp. 131–198, Association for Computational Linguistics, August 2016.

[23] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin, "Linear mode connectivity and the lottery ticket hypothesis," in *International Conference on Machine Learning*, pp. 3259–3269, PMLR, 2020.

[24] R. Entezari, H. Sedghi, O. Saukh, and B. Neyshabur, "The role of permutation invariance in linear mode connectivity of neural networks," in *International Conference on Learning Representations*, 2021.

[25] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International Conference on Machine Learning*, pp. 3519–3529, PMLR, 2019.

[26] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, *et al.*, "Robust fine-tuning of zero-shot models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022.

[27] S. I. Mirzadeh, M. Farajtabar, D. Gorur, R. Pascanu, and H. Ghasemzadeh, "Linear mode connectivity in multitask and continual learning," in *International Conference on Learning Representations*, 2020.

# Abstract

## Analysis on the Uniformity of Parameter Efficient Transfer Learning Method

Choonghyun Park
Department of Computer Science and Engineering
The Graduate School
Seoul National University

Pretrained Language Models (PLMs) are models that acquired general language skills through pretraining on tasks where the model predicts words that fit in a given text. They are making remarkable progresses in various NLP tasks. Fine tuning a model based on the pretrained weight improves the downstream task performance. It also makes models tuned on the same task close to each other. Based on these characteristics, it is possible to use the averaged weight of multiple fine tuned models as a single model. The averaged model shows improved performance without any additional inference cost.

PLMs with more parameters give better results, and recently, researchers are building Large Language Models (LLMs) with more than 100 billion parameters to make a better model. Although LLMs can outperform existing PLMs, they are too large to fine tune for each downstream task. Therefore, recent studies are proposing Parameter Efficient Transfer Learning (PETL) methods which tune the model with a small number of parameters, instead of updating the entire parameter of the model.

Previous works on PETL suggested a variety of PETL methods whose performances are on par with fine tuning the model itself. However, the similarity

of different train results based on PETL methods are lacking. Using Low-Rank Adaptation (LoRA), a representative PETL method, we examine the similarity of different train results on each dataset. Although the uniformity among LoRA train runs is worse compared to the fine tuning results, the output representations of trained models are similar for a given input, and different LoRA weights are in a common basin in the loss landscape. The required amount of trainable parameters to achieve fine tuning level performance affects these characteristics.

Based on the similarity of models trained with LoRA, we tried a simple ensemble method where the weighted sum of all trained weights is used as a weight of a single model. We empirically show that it is possible to obtain an improved model with the same size using greedy approach.

**Keywords**: Parameter Efficient Transfer Learning, Linear Mode Connectivity, Feature Similarity, LoRA, Pretrained Language Model, Ensemble Model, Fine Tuning

**Student Number**: 2021-20229