Parameter-Efficient Transfer Learning for NLP

▼ 연구 목적

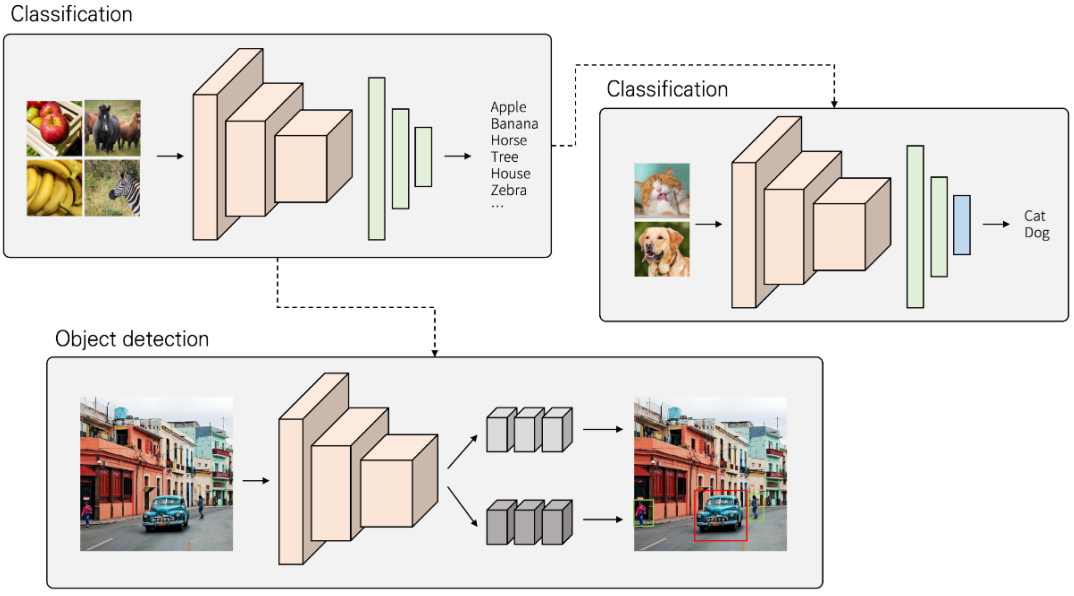
기존의 fine-tuning은 비효율적이므로 adapter modules을 이용한다.

소수의 파라미터의 추가를 통해 많은 task를 해결하도록 하는것이 목표이다.

▼ transfer learning

transfer learning techniques

- 학습된 모델을 다른 작업에 이용하는 것



1. Features-based transfer

- 학습된 모델의 특성(feature)을 다른 문제에 활용하는 방법

2. Fine-tuning

- 사전학습모델의 가중치를 미세하게 조정하는 기법

▼ Adapter

사전 학습된 모델의 네트워크의 레이어 사이에 추가되는 모듈

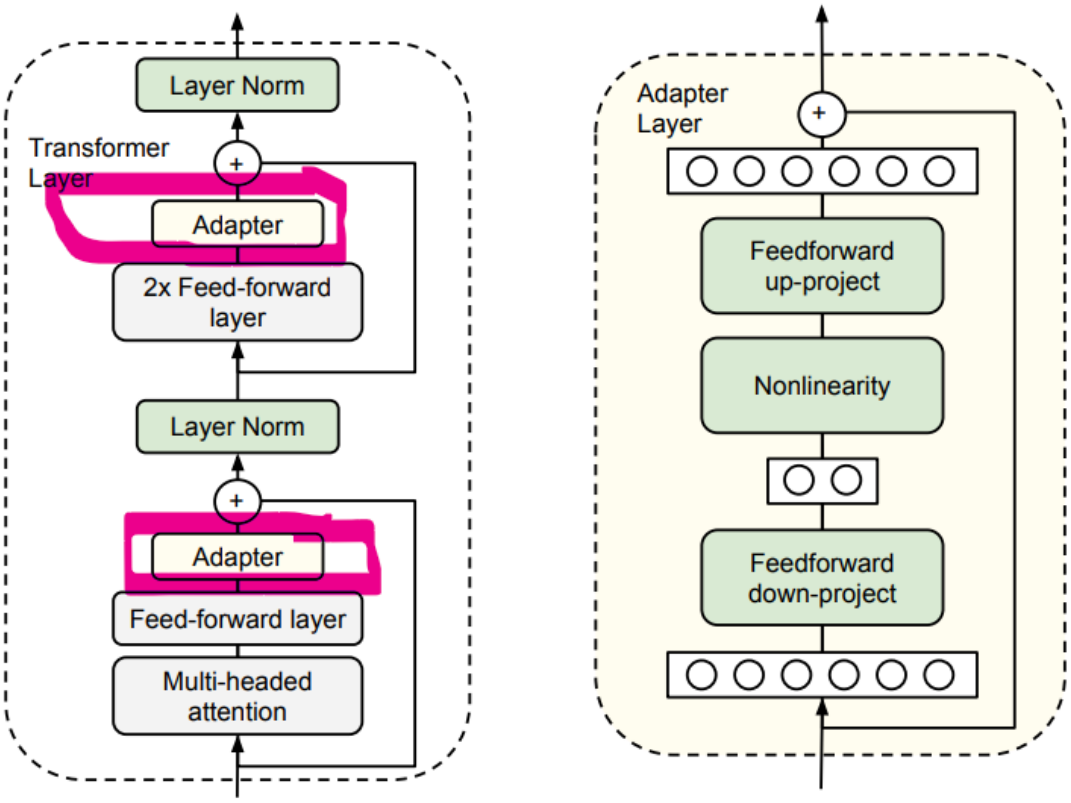
Adapter modules 주요 특징

1. 적은 수의 파라미터

- 원래 원본 네트워크의 레이어에 비해 작아야 한다.

2. a near-identity initialization

- adapted model을 안정적으로 학습시키기 위해서, 원래 네트워크가 영향을 받지 않음
- 모델의 가중치를 항등 매핑에 가깝게 설정하는 것이다. 즉 입력을 그대로 출력으로 내보내는 매핑을 사용하여 원래 네트워크가 영향을 받지 않게한다.



원래 네트워크에 새로운 레이어(adapter)를 추가하는 하는것이다.

새로운 레이어 부분의 파라미터만 훈련 시킨다. 즉 원래 파라미터는 그래도 두고 adapter의 파라미터만 학습, 이때 원래 원본 네트워크의 레이어에 비해 작아야 합니다

녹색 레이어에서 downstream data가 학습

파라미터의 수를 제한하기 위해서 bottleneck architecture 구조를 사용한다.

▼ bottleneck architecture 사용 이유

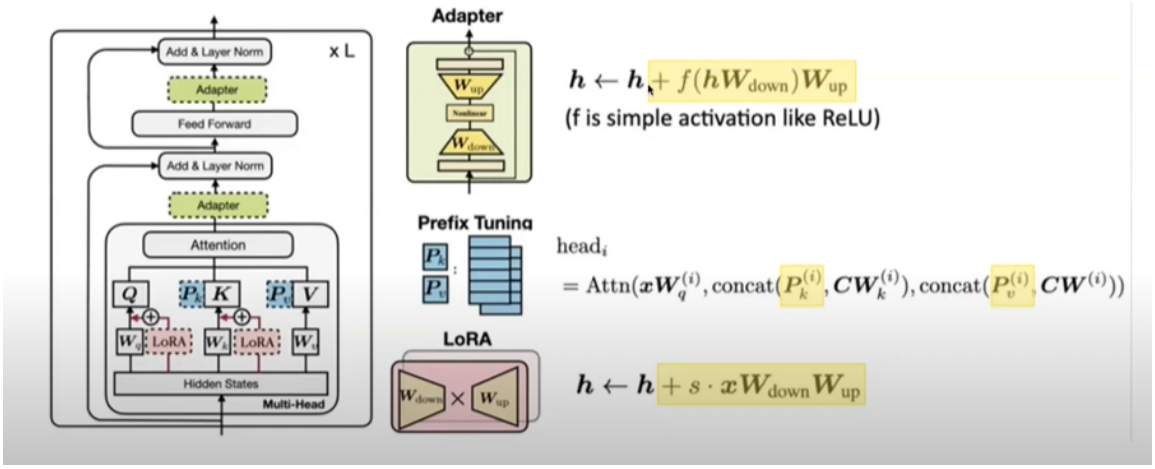
학습 파라미터의 수를 크게 줄이면서도 원래 모델의 성능을 유지할 수 있기 때문입니다.

어댑터의 구조를 보면 **입력 차원을 줄이고, 그 결과를 다시 원래의 차원으로 복원하는데 이 과정에서 모델은 필요한 정보를 압축하여 학습하게 되며 파라미터의 효율성을 높여줍니다.**

skip-connection 사용 이유

- Adapter를 통과하면서 정보가 손실 되는것을 방지하기 위해서

- 만약 파라미터가 parameters of the projection layers에서 거의 0에 가깝게 초기화되었다면 이 모듈은 identity function(항등함수 - 모든 입력을 그대로 출력하는 함수)로 초기화 된다.
- 원래의 모델 구조에 최소한의 영향을 미치면서도 새로운 작업에 대한 학습을 가능하게 해준다.
-



adapter 수행 과정

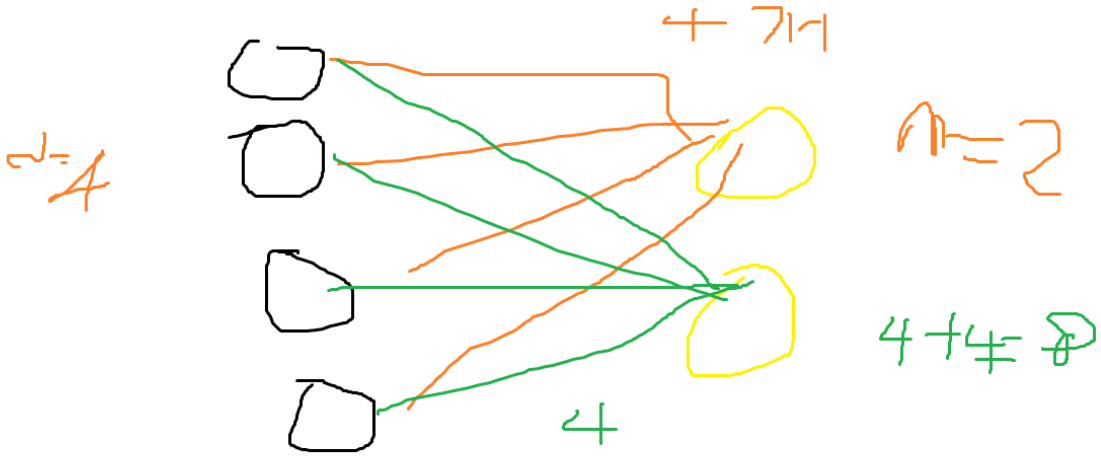
1. feed- forward layer에서 나온 출력값을 입력값으로 사용한다. 이때의 특징 값의 차원이 d차원이라고 가정
2. d차원을 더 작은 m차원으로 투영
3. nonlinearity 적용한다.
4. 다시 m 차원을 d차원으로 투영

매개변수의 총수

2md + d + m 이며 이때 m << d 여야 한다.

md → 병목 계층의 차원 ↔ 원래 차원

d, m의 경우 편향



이때의 매개변수의 경우 original model의 약 0.5 ~ 8퍼센트이다.

장점

- 모든 파라미터를 조정하는것과 비교했을때 어댑터 기반인 튜닝은 3% task-specific parameters 사용해서 100% task-specific parameters 을 사용한것 과 같은 성능을 낸다.

▼ 실험

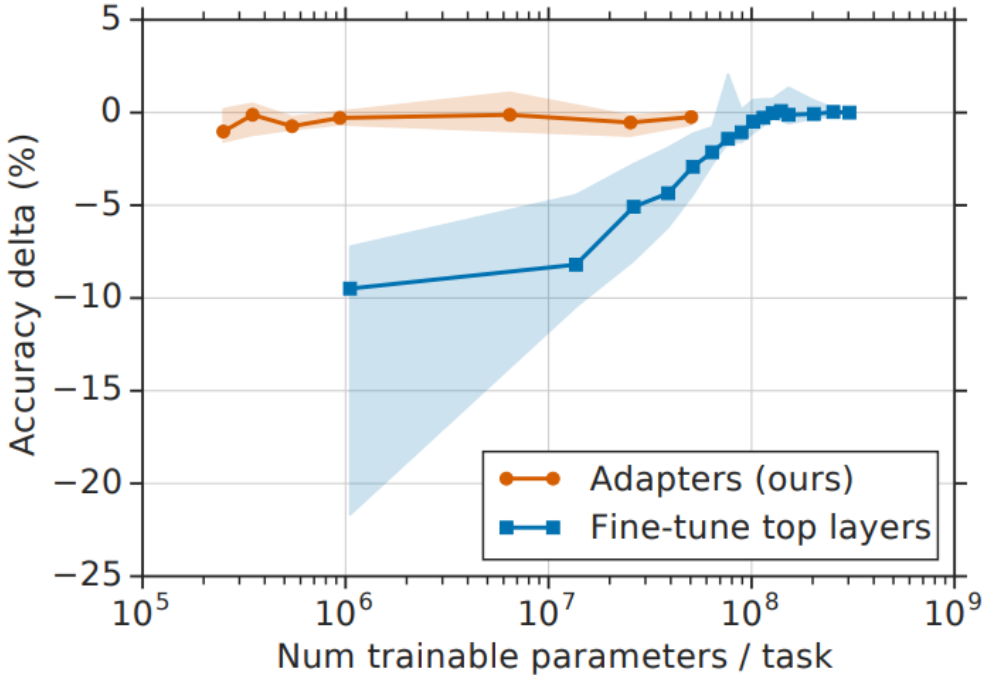


Figure 1. Trade-off between accuracy and number of trained taskspecific parameters, for adapter tuning and fine-tuning

GLUE benchmark

Parameter-Efficient Transfer Learning for NLP

	Total num params	Trained params / task	CoLA	SST	MRPC	STS-B	QQP	MNLI_m	MNLI_mm	QNLI	RTE	Total
BERT_LARGE	9.0×	100%	60.5	94.9	89.3	87.6	72.1	86.7	85.9	91.1	70.1	80.4
Adapters (8-256)	1.3×	3.6%	59.5	94.0	89.5	86.9	71.8	84.9	85.1	90.7	71.5	80.0
Adapters (64)	1.2×	2.1%	56.9	94.2	89.6	87.3	71.8	85.3	84.6	91.4	68.8	79.6

adapter와 fine-tuning 을 비교한 결과를 보여준다.

성능은 비슷하지만 파라미터의 수가 줄어든것을 확인 할 수 있다.

Dataset	No BERT baseline	BERT_BASE Fine-tune	BERT_BASE Variable FT	BERT_BASE Adapters
20 newsgroups	91.1	92.8 ± 0.1	92.8 ± 0.1	91.7 ± 0.2
Crowdflower airline	84.5	83.6 ± 0.3	84.0 ± 0.1	84.5 ± 0.2
Crowdflower corporate messaging	91.9	92.5 ± 0.5	92.4 ± 0.6	92.9 ± 0.3
Crowdflower disasters	84.9	85.3 ± 0.4	85.3 ± 0.4	84.1 ± 0.2
Crowdflower economic news relevance	81.1	82.1 ± 0.0	78.9 ± 2.8	82.5 ± 0.3
Crowdflower emotion	36.3	38.4 ± 0.1	37.6 ± 0.2	38.7 ± 0.1
Crowdflower global warming	82.7	84.2 ± 0.4	81.9 ± 0.2	82.7 ± 0.3
Crowdflower political audience	81.0	80.9 ± 0.3	80.7 ± 0.8	79.0 ± 0.5
Crowdflower political bias	76.8	75.2 ± 0.9	76.5 ± 0.4	75.9 ± 0.3
Crowdflower political message	43.8	38.9 ± 0.6	44.9 ± 0.6	44.1 ± 0.2
Crowdflower primary emotions	33.5	36.9 ± 1.6	38.2 ± 1.0	33.9 ± 1.4
Crowdflower progressive opinion	70.6	71.6 ± 0.5	75.9 ± 1.3	71.7 ± 1.1
Crowdflower progressive stance	54.3	63.8 ± 1.0	61.5 ± 1.3	60.6 ± 1.4
Crowdflower US economic performance	75.6	75.3 ± 0.1	76.5 ± 0.4	77.3 ± 0.1
Customer complaint database	54.5	55.9 ± 0.1	56.4 ± 0.1	55.4 ± 0.1
News aggregator dataset	95.2	96.3 ± 0.0	96.5 ± 0.0	96.2 ± 0.0
SMS spam collection	98.5	99.3 ± 0.2	99.3 ± 0.2	95.1 ± 2.2
Average	72.7	73.7	74.0	73.3
Total number of params	—	17×	9.9×	1.19×
Trained params/task	—	100%	52.9%	1.14%

Table 2. Test accuracy for additional classification tasks. In these experiments we transfer from the BERT_BASE model. For each task and algorithm, the model with the best validation set accuracy is chosen. We report the mean test accuracy and s.e.m. across runs with different random seeds.

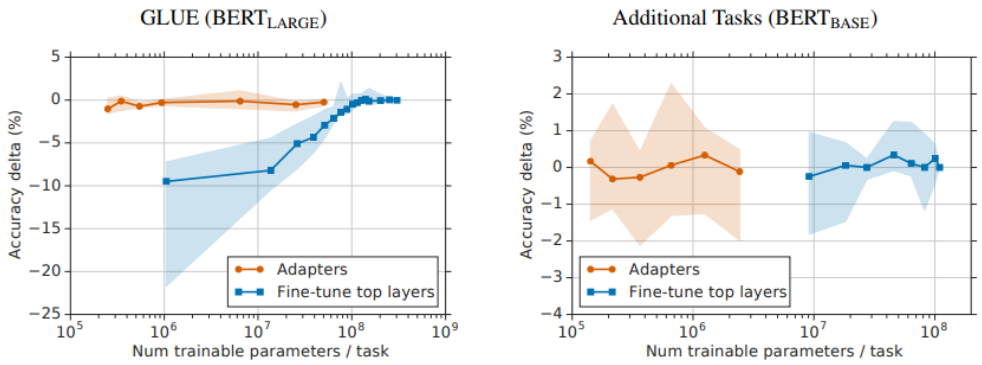


Figure 3. Accuracy versus the number of trained parameters, aggregated across tasks. We compare adapters of different sizes (orange) with fine-tuning the top n layers, for varying n (blue). The lines and shaded areas indicate the 20th, 50th, and 80th percentiles across tasks. For each task and algorithm, the best model is selected for each point along the curve. For GLUE, the validation set accuracy is reported. For the additional tasks, we report the test-set accuracies. To remove the intra-task variance in scores, we normalize the scores for each model and task by subtracting the performance of full fine-tuning on the corresponding task.