3.4. AdapterBERT

논문 : https://arxiv.org/abs/1902.00751

주의 : 실험을 통한 분석으로 수학적인 해석이 힘듦

3줄요약

1. BERT같은 Transformer model을 fine-tuning하는건 너무 오래걸리고 task끼리 파라미터 공유가 안되서 모델관리가 힘듦
2. Adapter tuning을 제안! Adapter Layer = NonLinear + Linear (Bottleneck 형식)
3. BERT모델은 냅두고 Adapter Layer+Layer Normalization 만 fine-tuning을 했더니 정확도도 유지되고 학습 파라미터수도 줄었다!

문제제기

- Transformer model 류에서 pre-training과 fine-tuning 과정으로 task별 SOTA를 찍었음
- fine-tuning과정에서 Transformer model의 모든 파라미터를 재학습하면 문제가 생김
  - 하드웨어 연산이 부담됨
  - task별로 transformer 모델의 파라미터가 변경됨 (재사용 불가)
- 논문은 여러 다운 스트림 작업에서 큰 텍스트 모델을 조정하기위해 아래 항목 3개를 달성하고싶음
  1. 우수한 성능 달성하기
  2. multi-task를 구현하기위해 모든 데이터를 동시에 접근할 필요가 없도록 만들기
  3. task당 추가되는 파라미터 수를 적게 만들기

제안

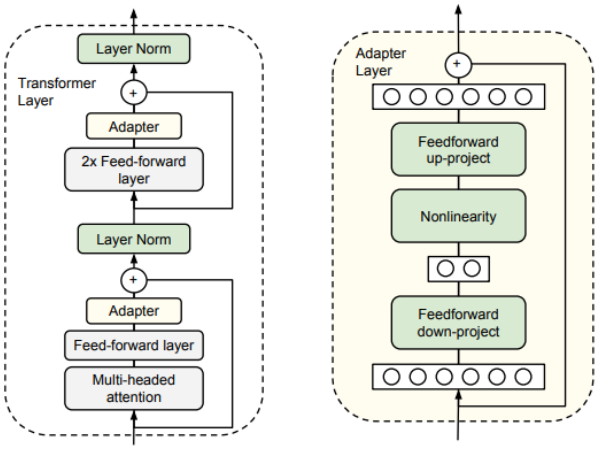

Figure 2. Architecture of the adapter module and its integration with the Transformer. **Left:** We add the adapter module twice to each Transformer layer: after the projection following multi-headed attention and after the two feed-forward layers. **Right:** The adapter consists of a bottleneck which contains few parameters relative to the attention and feedforward layers in the original model. The adapter also contains a skip-connection. During adapter tuning, the green layers are trained on the downstream data, this includes the adapter, the layer normalization parameters, and the final classification layer (not shown in the figure).

그림 1. Transformer의 Encoder Layer에 Adapter Layer를 추가한 모습. fine-tuning 과정에서 BERT의 파라미터는 고정하고 초록색 부분만 학습하는 것을 제안.

- **Adapter Layer**
  - Feed Forward Layer + Feed Forward Layer + Skip-connection으로 구성된 레이어
  - 논문은 3번조건(task당 학습할 파라미터를 적게 만들기)을 달성하기 위해서 Bottleneck 구조의 FFN을 사용
  - BERT의 모든 Encoder Layer(Base 모델 기준 12개)에 Adapter Layer를 추가하고 fine-tuning 과정에서 BERT의 파라미터를 고정시키는 방식을 제안함
  - Adapter Layer만 학습하기 때문에 기존 모델의 파라미터에 0.5~8%정도만 fine-tuning 함
  - **BERT의 파라미터를 고정하기 때문에 Adapter Layer의 파라미터만 Task별로 관리하면 모델을 여러개 만들필요가 없게됨!**

분석과 의견

논문은 실험을 통해 Adapter Layer가 좋다는 것을 증명 (수식적으로 의미를 두지 않음)

	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

Table 1. Results on GLUE test sets scored using the GLUE evaluation server. MRPC and QQP are evaluated using F1 score. STS-B is evaluated using Spearman's correlation coefficient. CoLA is evaluated using Matthew's Correlation. The other tasks are evaluated using accuracy. Adapter tuning achieves comparable overall score (80.0) to full fine-tuning (80.4) using 1.3× parameters in total, compared to 9×. Fixing the adapter size to 64 leads to a slightly decreased overall score of 79.6 and slightly smaller model.

표 1. BERT와 제안 기법의 성능 비교. Adapter Layer만 학습한 방식이 BERT 모델 전체를 학습한 것과 큰 차이가 없음.

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.

표 2. 추가 분류작업에 대한 테스트 정확도. No BERT baseline은 AutoML을 의미. BERT Variable FT는 상위 n개 Layer만 fine-tuning한 것을 의미 ($n \in \{1, 2, 3, 5, 7, 9, 11, 12\}$).

- 표 2의 의미
  - 17개의 task에 대한 모델을 만들려면....
    - BERT Fine-tune : 모든 모델이 파라미터가 다르기때문에 17 task * 1 배
    - BERT Variable FT : task 별로 상위 n개 layer을 학습하기 때문에 9.9배
    - BERT Adapters : Adapter layer만 학습하면 되기 때문에 1배(BERT) + 0.19배(Adapter Layer의 parameter)
  - 사실 BERT는 Task에 따라 상위 n개 Layer만 fine-tuning해도 성능차이가 별로 없다. 단, n을 찾기위해 여러번 튜닝을 해봐야될듯

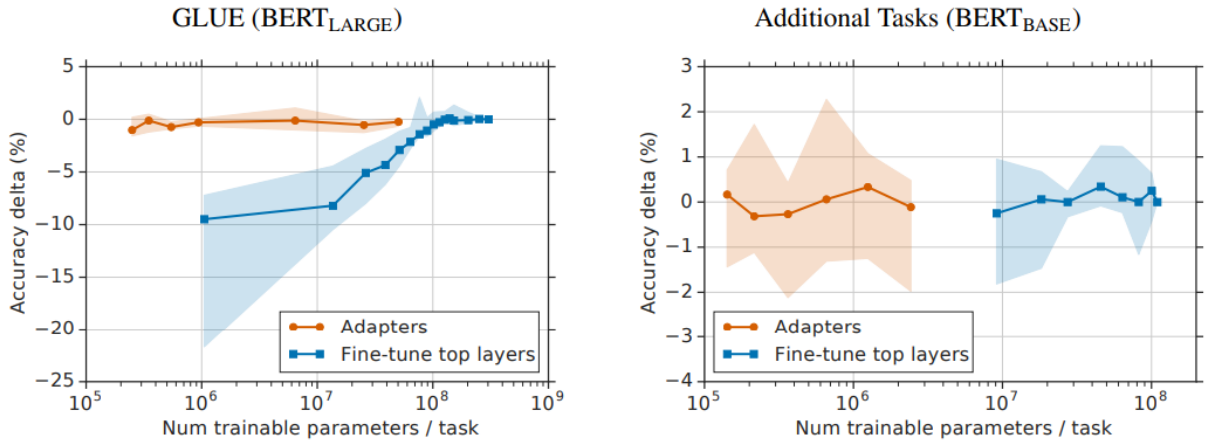

Figure 3. Accuracy versus the number of trained parameters, aggregated across tasks. We compare adapters of different sizes (orange) with fine-tuning the top n layers, for varying n (blue). The lines and shaded areas indicate the 20th, 50th, and 80th percentiles across tasks. For each task and algorithm, 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.

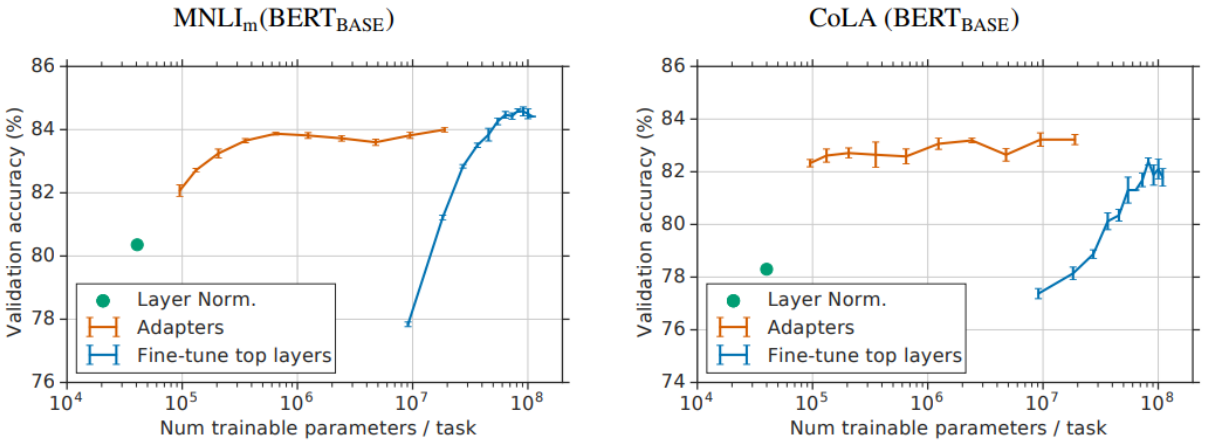

Figure 4. Validation set accuracy versus number of trained parameters for three methods: (i) Adapter tuning with an adapter sizes 2^n for $n = 0 \ldots 9$ (orange). (ii) Fine-tuning the top k layers for $k = 1 \ldots 12$ (blue). (iii) Tuning the layer normalization parameters only (green). Error bars indicate ± 1 s.e.m. across three random seeds.

- 그림 3의 의미
  - 실험 목적 : full fine-tuning한 BERT의 성능을 따라잡으려면 Adapter size(오렌지)를 얼마로 해야되고 Top layer를 얼만큼 학습해야될까?
  - 포인트에 대한 정확한 값 명시가 논문에 없어서 생략
- 그림 4의 의미
  - 특정 Task에 대해 Adapter size(2의 n승, 오렌지)와 fine-tuning top layer 수(1~12, 블루)의 정확도 결과 실험

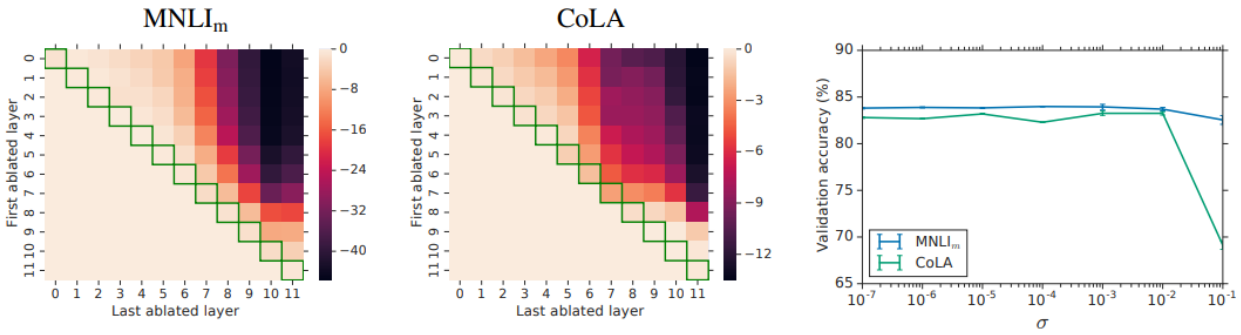

Figure 6. **Left, Center:** Ablation of trained adapters from continuous layer spans. The heatmap shows the relative decrease in validation accuracy to the fully trained adapted model. The y and x axes indicate the first and last layers ablated (inclusive), respectively. The diagonal cells, highlighted in green, indicate ablation of a single layer's adapters. The cell in the top-right indicates ablation of all adapters. Cells in the lower triangle are meaningless, and are set to 0%, the best possible relative performance. **Right:** Performance of BERT$_{\text{BASE}}$ using adapters with different initial weight magnitudes. The x-axis is the standard deviation of the initialization distribution.

- 그림 6에 대한 의미
  - 목적 1 : Adapter Layer가 실제 모델에 미치는 영향을 파악해보자! Adapter tuning이 끝난후 각 Layer의 Adapter Layer를 제거했을 때의 정확도 변화 분석. (왼쪽과 오른쪽 그림)
  - low-level adapter layer를 제거했을 땐 성능이 크게 차이가 없음!(정확도 2% 내외)
  - high-level adapter layer를 제거하면 성능 차이가 크게 발생 => 다른 논문(Universal Language Model Fine-tuning for Text Classification : https://arxiv.org/abs/1801.06146)이 이미 증명함
  - 그러면 low-level adapter layer에서 parameter sharing을 하자! => 새로운 task를 학습할 때 더 적은 파라미터를 학습하면 된다
  - 목적 2 : Adapter Layer의 FFN을 초기화할 때 표준 편차를 얼마로 정해야될까?
  - 결과 : 초기 파라미터 설정에 표준편차를 0.01이상으로 설정하면 성능이 하락되는 것을 확인

- 그외 논문이 실험해봤지만 성능변화가 없는것
  - Adapter Layer에 batch/layer normalization을 추가
  - Adapter 내부에 Layer수를 증가
  - tanh같은 Activation function 변경
  - Attention Layer 뒤에만 Adapter Layer 추가
  - 메인 레이어에 병렬로 Adapter Layer를 추가하고 multiplicative interaction 해보기