ALBERT

A Lite BERT for Self-supervised Learning of Language Representations

이은수

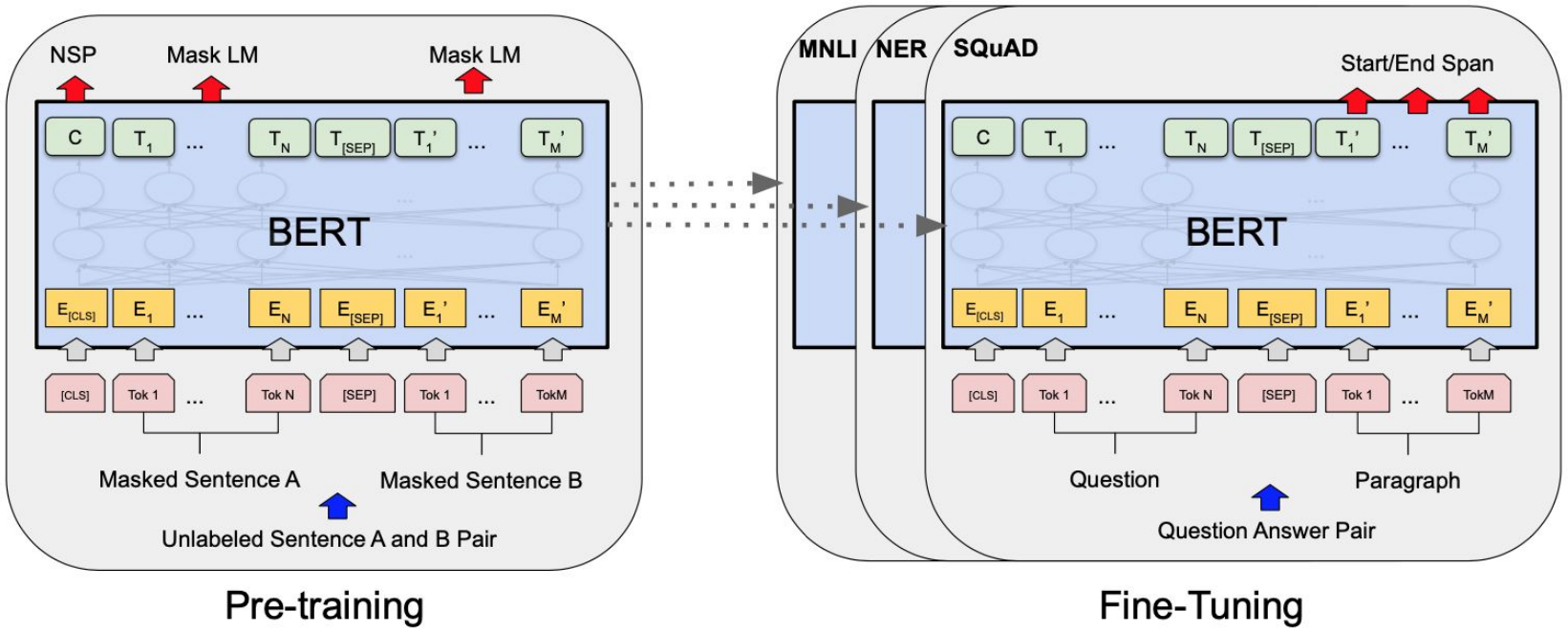
Abstract
요약

ALBERT (A Lite BERT)

Google Research & TTIC (Toyota Technological Institude at Chicago)

- 2019년에 공개되었으며, ICLR 2020에 게재됨
- NLP task에서 pre-training 모델을 학습하기 위해 모델 크기를 키우는 것은 성능 향상을 가져오지만, GPU/TPU 메모리 제한과 매우 긴 학습 시간이 단점으로 꼽힘
- 적은 메모리 사용량과 학습 시간 단축을 위해, 2가지 **parameter reduction**을 제안함
- BERT-large보다 적은 파라미터로 GLUE, RACE, SQuAD 벤치마크에 대해 SOTA를 달성함

Introduction
소개



Devlin et al., (2018)

- Pre-training시킨 네트워크를 사용하는 것은 NLP task에서 획기적인 성능 향상을 불러일으킴
- **그러나 모델의 크기를 계속 키우기만 하면 NLP task 성능을 더 향상시킬 수 있을까?**

Introduction
소개

GPU 1)

네트워크 2)

- BERT-large 기준 3.4억개의 파라미터가 있기 때문에, 모델을 무한정 확장하는 것은 하드웨어 메모리 제한이 발목을 잡게 됨
- 모델 병렬화, 메모리 관리 등으로 해결할 수도 있지만, ALBERT는 파라미터 자체를 훨씬 적게 설계하여 해결함

1) "NVIDIA Quadro P4000 GPU | Dell 대한민국", Dell, 2020.6.8., https://www.dell.com/ko-kr/work/shop/nvidia-quadro-p4000-gpu/apd/490-befg/그래픽-및-비디오-카드
2) "[네트워크]네트워크 입문(1)", 똥선생, 2020.6.8., https://kujung.tistory.com/89

The elements of ALBERT
ALBERT 구성 요소

Parameter Reduction

1. **Factorized embedding parameterization**

- BERT (Devlin et al., 2018) 및 후속 연구 등에서 WordPiece embedding 크기 E는 hidden layer 크기 H와 연결하기 위해 동일한 크기로 설정되어 있지만, 이는 모델링 측면과 실용적 측면에서 최적의 방법이 아님

- NLP 모델들은 단어 크기 V가 매우 큰 값이며, 이 때 E와 H가 동일하다면 **O(V×E)**의 크기는 10억 개 이상의 파라미터를 가지게 되고 속도를 매우 느리게 만듬

The elements of ALBERT
ALBERT 구성 요소

Parameter Reduction

1. **Factorized embedding parameterization**

- ALBERT는 기존에 매우 큰 크기의 WordPiece embedding E로 매핑했던 것을 두 개의 matrix로 나누어 표현함

- 좀 더 작은 WordPiece embedding 크기 E를 설정하여 먼저 매핑하고, 이를 다시 hidden layer 크기 H로 매핑함으로써, 파라미터의 수는 **O(V×E + E×H)**로 줄어듬

- 이처럼 WordPiece embedding 크기 E를 분해하게 되면 E는 작게 가져가면서도 hidden layer 크기 H는 크게 가져갈 수 있게 되면서, BERT가 제공하는 문맥 종속적 표현의 힘을 유지할 수 있음

- SQuAD2.0은 80.4에서 80.3로, RACE는 68.2에서 67.9로 떨어지지만, 프로젝션 블록 파라미터를 **80%** 줄일 수 있음

The elements of ALBERT
ALBERT 구성 요소

Parameter Reduction

2. Cross-layer parameter sharing

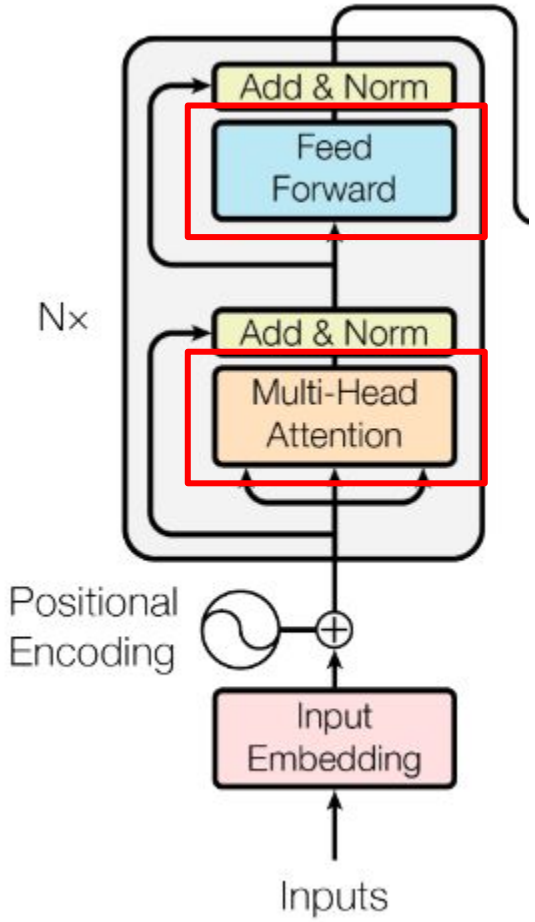
- 유사 작업을 수행하는 특정 layer끼리 파라미터를 공유해, 네트워크의 깊이에 따라 파라미터 수가 늘어나는 것을 방지함

- Attention layer 또는 feed-forward network 등의 파라미터에 적용할 수 있는데, ALBERT에서는 두 블록의 파라미터를 모두 공유함으로써 **90%**의 파라미터 수 감소를 달성할 수 있으며, 이는 전체 **70%**의 감소를 달성함

The elements of ALBERT
ALBERT 구성 요소

Parameter Reduction

2. **Cross-layer parameter sharing**



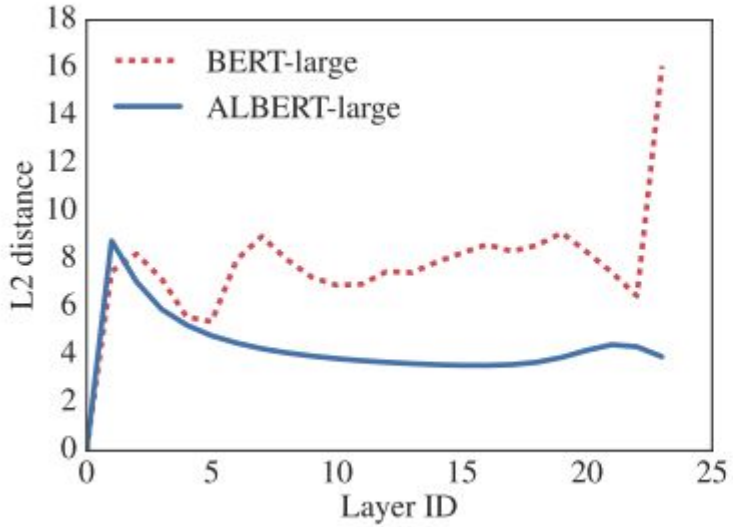
Vaswani et al., (2017)

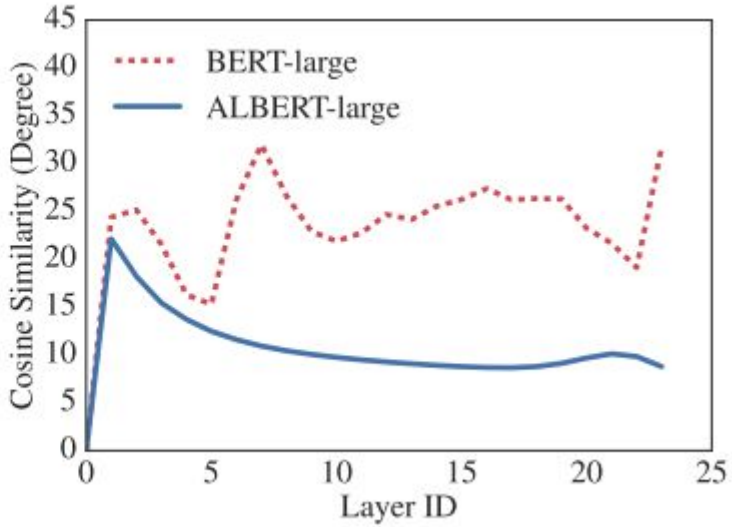
The elements of ALBERT
ALBERT 구성 요소

Parameter Reduction

2. Cross-layer parameter sharing

- 각 Layer의 입력층과 출력층에 대한 L2 거리와 코사인 유사도를 측정해보면 ALBERT가 BERT보다 매끄러우며, 이는 cross-layer parameter sharing이 네트워크 안정화에 영향을 준다는 것을 알 수 있음





Lan et al., (2019)

The elements of ALBERT
ALBERT 구성 요소

Parameter Reduction

Result

- WordPiece Embedding matrix를 분해하고, attention layer 및 feed-forward network 파라미터를 공유할 경우, SQuAD2.0은 80.0에서 79.7로, RACE는 64.0에서 60.1로 성능 저하가 발생함

- 그러나 ALBERT를 BERT-large와 유사하게 구성했을 때, 파라미터 수가 18배 적고, 약 1.7배 빠르게 훈련됨

- 또한 파라미터를 줄이는 데 그치지 않고 다시 확장할 수도 있는데, hidden layer의 크기를 4096으로 구성한 ALBERT-xxlarge의 경우 BERT-large에 비해 파라미터가 30% 감소하고, 또한 SQuAD2.0은 83.9에서 88.1로, RACE는 73.8에서 82.3으로 성능이 향상됨

The elements of ALBERT
ALBERT 구성 요소

Loss

Inter-sentence coherence loss

- BERT는 Masked Language Modeling(MLM)과 Next-Sentence Prediction(NSP) loss를 사용했고, 그 중 NSP는 두 문장이 관계 있는 문장인지 아닌지를 예측함

- 그러나 XLNet(Yang et al., 2019), RoBERTa(Liu et al., 2019) 등의 후속 연구에서 NSP에 의문을 제기하고 이를 제거함

- ALBERT는 NSP loss가 비효율적인 이유를 또 다른 loss인 MLM에 비해 매우 쉽기 때문이라고 예측함. NSP가 수행하는 topic prediction 및 coherence prediction 중 topic prediction은 coherence prediction에 비해 쉽고, MLM과 겹치기 때문

The elements of ALBERT
ALBERT 구성 요소

Loss

Inter-sentence coherence loss

- ALBERT는 다른 후속 연구들과는 달리 문장간 관계를 보는 것이 중요하다고 판단했고, topic 예측을 피하고 coherence만을 판단하는 sentence-order prediction(SOP) loss를 사용함

- NSP는 서로 다른 document에서 가져온 두 문장을 서로 관계 없다고 예측하는 반면, SOP는 같은 document에서 가져온 두 문장을 50% 확률로 순서를 뒤집었을 때, 제대로 된 순서인지 반대로 된 순서인지 예측함

Model Setup
모델 구성

Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
ALBERT	base	12M	12	768	128	True
	large	18M	24	1024	128	True
	xlarge	60M	24	2048	128	True
	xxlarge	235M	12	4096	128	True

Lan et al., (2019)

Experiments
실험

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3	17.7x
	large	334M	92.2/85.5	85.0/82.2	86.6	93.0	73.9	85.2	3.8x
	xlarge	1270M	86.4/78.1	75.5/72.6	81.6	90.7	54.3	76.6	1.0
ALBERT	base	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1	21.1x
	large	18M	90.6/83.9	82.3/79.4	83.5	91.7	68.5	82.4	6.5x
	xlarge	60M	92.5/86.1	86.1/83.1	86.4	92.4	74.8	85.5	2.4x
	xxlarge	235M	**94.1/88.3**	**88.1/85.1**	**88.0**	**95.2**	**82.3**	**88.7**	1.2x

Table 3: Dev set results for models pretrained over BOOKCORPUS and Wikipedia for 125k steps. Here and everywhere else, the Avg column is computed by averaging the scores of the downstream tasks to its left (the two numbers of F1 and EM for each SQuAD are first averaged).

BERT와 ALBERT 벤치마크 비교
Lan et al., (2019)

Experiments
실험

Model	E	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base not-shared	64	87M	89.9/82.9	80.1/77.8	82.9	91.5	66.7	81.3
	128	89M	89.9/82.8	80.3/77.3	83.7	91.5	67.9	81.7
	256	93M	90.2/83.2	80.3/77.4	84.1	91.9	67.3	81.8
	768	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base all-shared	64	10M	88.7/81.4	77.5/74.8	80.8	89.4	63.5	79.0
	128	12M	89.3/82.3	80.0/77.1	81.6	90.3	64.0	80.1
	256	16M	88.8/81.5	79.1/76.3	81.5	90.3	63.4	79.6
	768	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8

Table 4: The effect of vocabulary embedding size on the performance of ALBERT-base.

Factorized embedding parameterization 효과
Lan et al., (2019)

Experiments
실험

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base E=768	all-shared	31M	88.6/81.5	79.2/76.6	82.0	90.6	63.3	79.8
	shared-attention	83M	89.9/82.7	80.0/77.2	84.0	91.4	67.7	81.6
	shared-FFN	57M	89.2/82.1	78.2/75.4	81.5	90.8	62.6	79.5
	not-shared	108M	90.4/83.2	80.4/77.6	84.5	92.8	68.2	82.3
ALBERT base E=128	all-shared	12M	89.3/82.3	80.0/77.1	82.0	90.3	64.0	80.1
	shared-attention	64M	89.9/82.8	80.7/77.9	83.4	91.9	67.6	81.7
	shared-FFN	38M	88.9/81.6	78.6/75.6	82.3	91.7	64.4	80.2
	not-shared	89M	89.9/82.8	80.3/77.3	83.2	91.5	67.9	81.6

Table 5: The effect of cross-layer parameter-sharing strategies, ALBERT-base configuration.

Cross-layer parameter sharing 효과
Lan et al., (2019)

Experiments
실험

	Intrinsic Tasks			Downstream Tasks					
SP tasks	MLM	NSP	SOP	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
None	54.9	52.4	53.3	88.6/81.5	78.1/75.3	81.5	89.9	61.7	79.0
NSP	54.5	90.5	52.0	88.4/81.5	77.2/74.6	81.6	**91.1**	62.3	79.2
SOP	54.0	78.9	86.5	**89.3/82.3**	**80.0/77.1**	**82.0**	90.3	**64.0**	**80.1**

Table 6: The effect of sentence-prediction loss, NSP vs. SOP, on intrinsic and downstream tasks.

Sentence-order Prediction(SOP) 효과
Lan et al., (2019)

References
참고문헌

- Devlin, Jacob, et al. "**Bert: Pre-training of deep bidirectional transformers for language understanding.**" arXiv preprint arXiv:1810.04805 (2018).
- Lan, Zhenzhong, et al. "**Albert: A lite bert for self-supervised learning of language representations.**" arXiv preprint arXiv:1909.11942 (2019).
- “ALBERT:언어 표현의 자율지도학습”, 시나브로의 테크산책, 2019.12.20., https://brunch.co.kr/@synabreu/32
- “ALBERT: A Lite BERT For Self-Supervised Learning of Language Representations”, reniew’s blog, 2020.3.10., https://reniew.github.io/49
- "NVIDIA Quadro P4000 GPU | Dell 대한민국", Dell, 2020.6.8., https://www.dell.com/ko-kr/work/shop/nvidia-quadro-p4000-gpu/apd/490-befg/그래픽-및-비디오-카드
- "[네트워크]네트워크 입문(1)", 똥선생, 2020.6.8., https://kujung.tistory.com/89

End of presentation