A Lite BERT for Self-supervised Learning of Language Representation

Name
조병웅
NLP
2024/04/30

TRAIN AND TEST

Contents

TRAIN AND TEST

Abstract & Introduction

Is having better NLP models as easy as having larger models?

Existing pre-trained Model ->

- Requires more memory
- Learn more parameters

- Increasing Computational cost

And..

Memory Degradation, OOM, Training Time

TRAIN AND TEST

Abstract & Introduction

Is having better NLP models as easy as having larger models?

Memory Degradation ->

단순한 hidden size 증가는
성능 저하가 발생할 수 있음

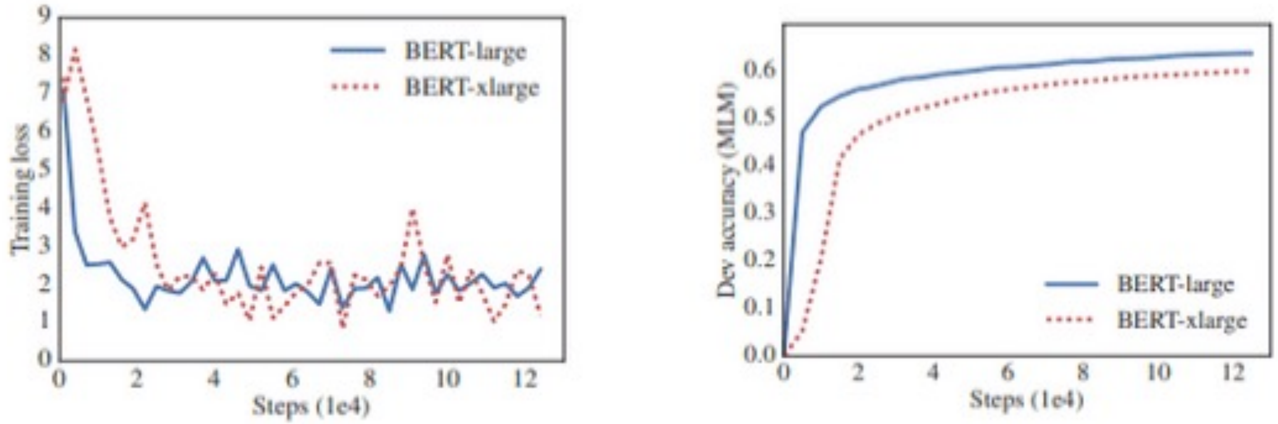


Figure 1: Training loss (left) and dev masked LM accuracy (right) of BERT-large and BERT-xlarge (2x larger than BERT-large in terms of hidden size). The larger model has lower masked LM accuracy while showing no obvious sign of over-fitting.

Model	Hidden Size	Parameters	RACE (Accuracy)
BERT-large (Devlin et al., 2019)	1024	334M	72.0%
BERT-large (ours)	1024	334M	73.9%
BERT-xlarge (ours)	2048	1270M	54.3%

Table 1: Increasing hidden size of BERT-large leads to worse performance on RACE.

TRAIN AND TEST

Abstract & Introduction

Is having better NLP models as easy as having larger models?

Out of Memory(OOM) ->

BERT Large의 경우, 384 length 이상이라면 Inference 불가함.

System	Seq Length	Max Batch Size
BERT-Base	64	64
...	128	32
...	256	16
...	320	14
...	384	12
...	512	6
BERT-Large	64	12
...	128	6
...	256	2
...	320	1
...	384	0
...	512	0

TRAIN AND TEST

Abstract & Introduction

Is having better NLP models as easy as having larger models?

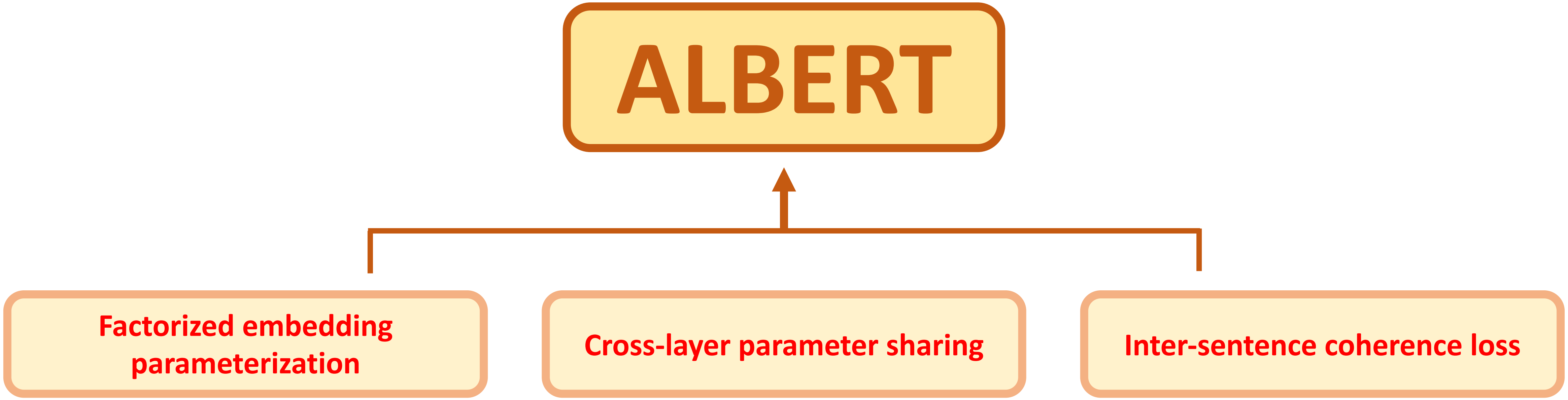
Training Time ->

BERT Large의 경우, 64ro의 TPU로도 4일을 소모함.
GPU의 경우 v100 사용시 약 8일 이상 소모됨.

Abstract & Introduction

Make a Lite & Strong BERT!

Solution -> **Model Downsizing & High Perfomance.**



Model Architecture

Factorized embedding parameterization

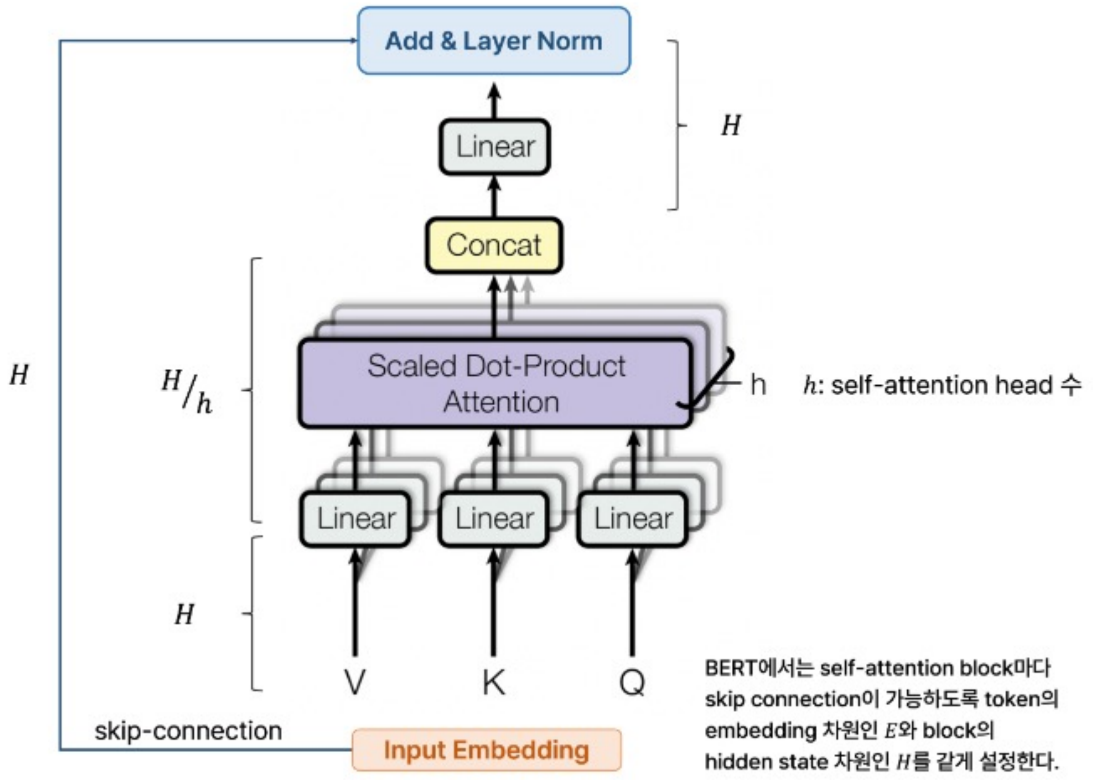
임베딩 사이즈(E) = 히든 사이즈(H)
E가 너무 작으면 정보량이 적음.
반대로 차원의 수가 너무 크면 연산량이 증가

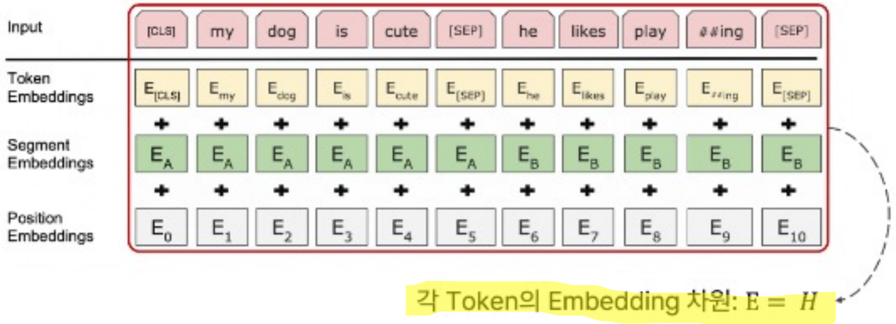
Self-attention block은 쌓아가며 레이어가 깊어질수록
의미론적으로 유의미한 정보를 지님

➔ 앞쪽 임베딩 벡터는 뒤쪽보다 상대적으로 적은 정보만을 필요로 하므로, 적은 차원의 벡터형태를 지니어도 됨.

임베딩 레이어 차원을 줄이는 방법

➔ 선형변환 layer를 추가하여 차원을 축소시킴.





TRAIN AND TEST

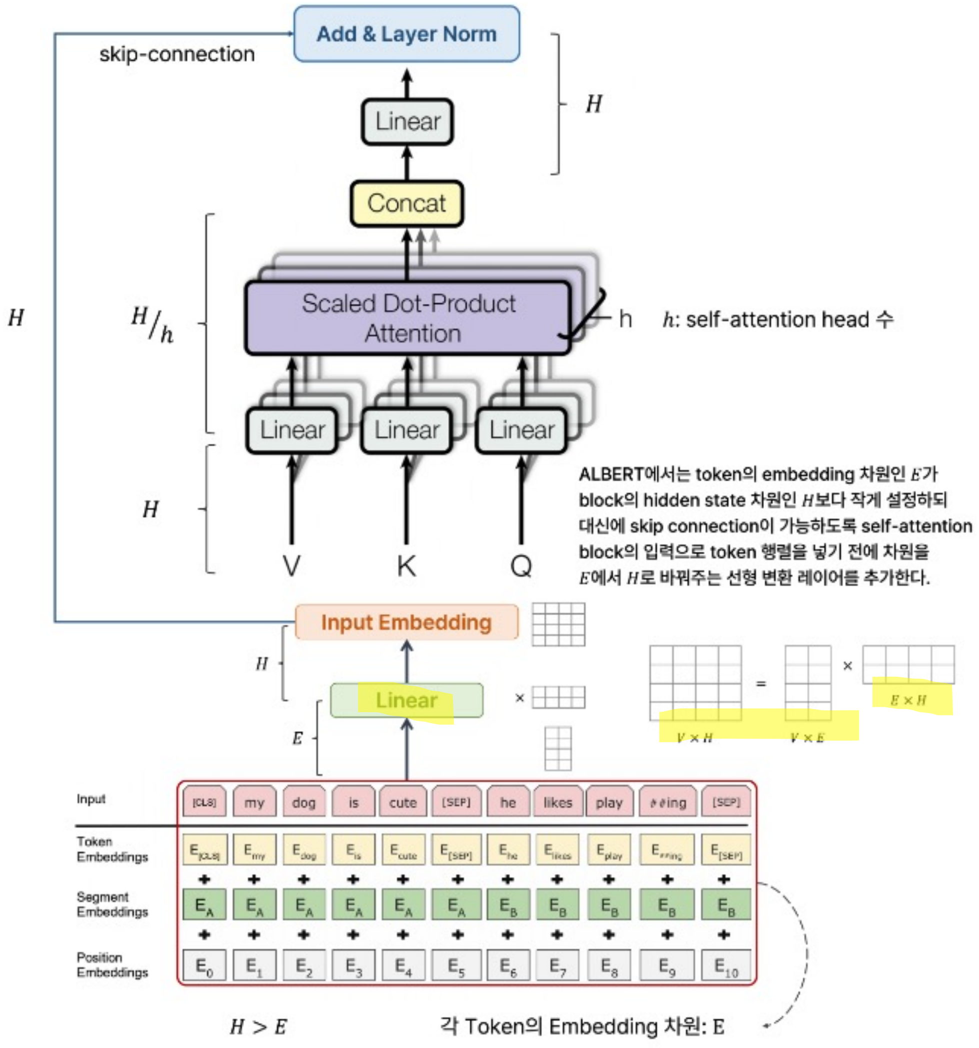
Model Architecture

Factorized embedding parameterization

Vocabulary size(V), Hidden state(H), Embedding vector(E)

Original BERT
➔ V x E

alBERT(low-rank matrix factorization)
➔ V x E + E x H



TRAIN AND TEST

Model Architecture

Factorized embedding parameterization

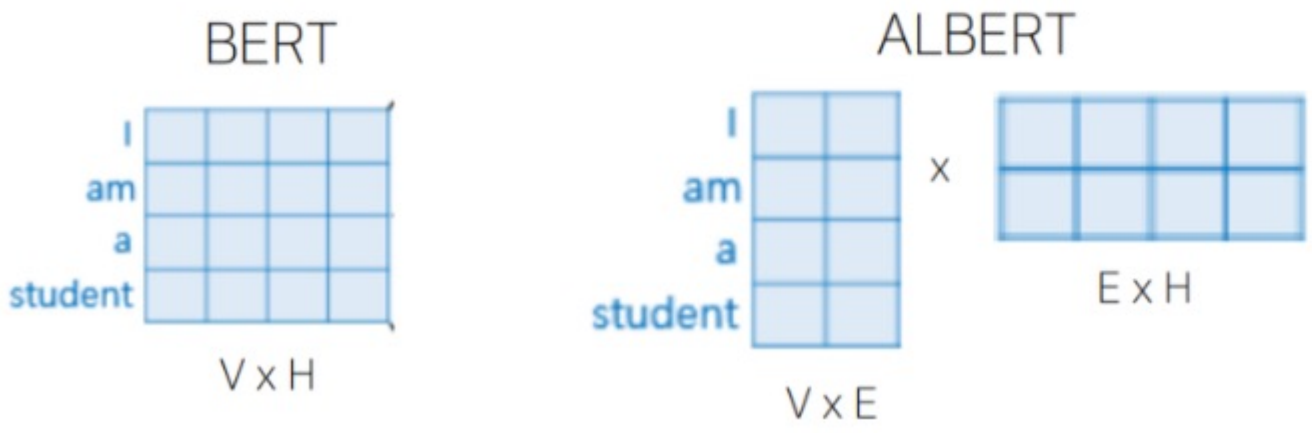
일종의 Decomposition(분해)

임베딩 사이즈 비교

#BERT
➔ V = 30000, H = 768 (가정)
➔ BERT의 파라미터 = 30,000 x 768 = 23,040,000

#ALBERT
➔ V = 30000, E = 128, H = 4096
➔ ALBERT의 파라미터 = 30,000 x 128 + 128 x 4096 = 4,364,288

** 파라미터 개수가 약 80% 감소



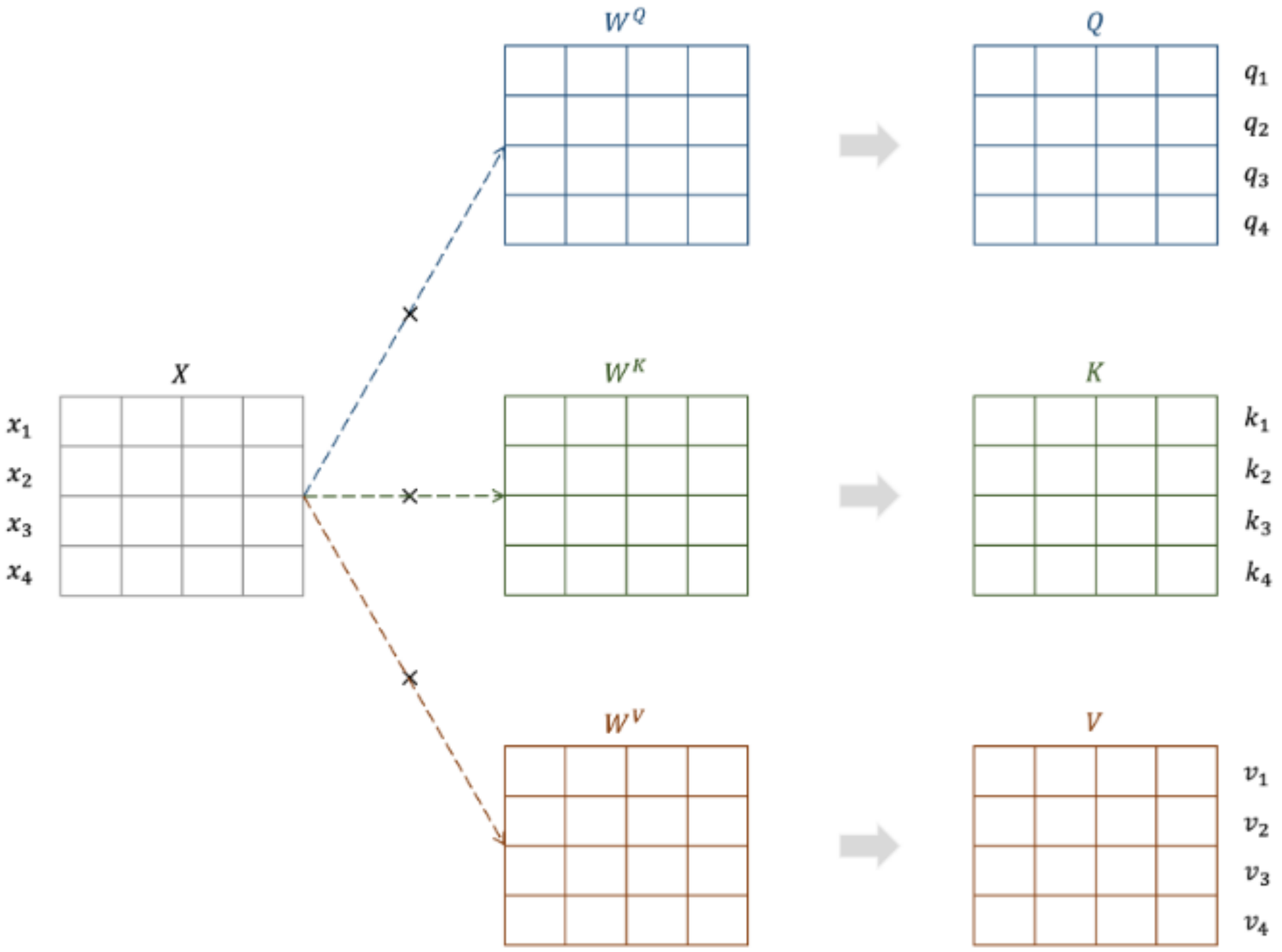
TRAIN AND TEST

Model Architecture

Cross-layer Parameter Sharing

Existing Model(Encoder)

Wq, Wk, Wv가 각각 multi-head attention의 head 수만큼 필요.



Model Architecture

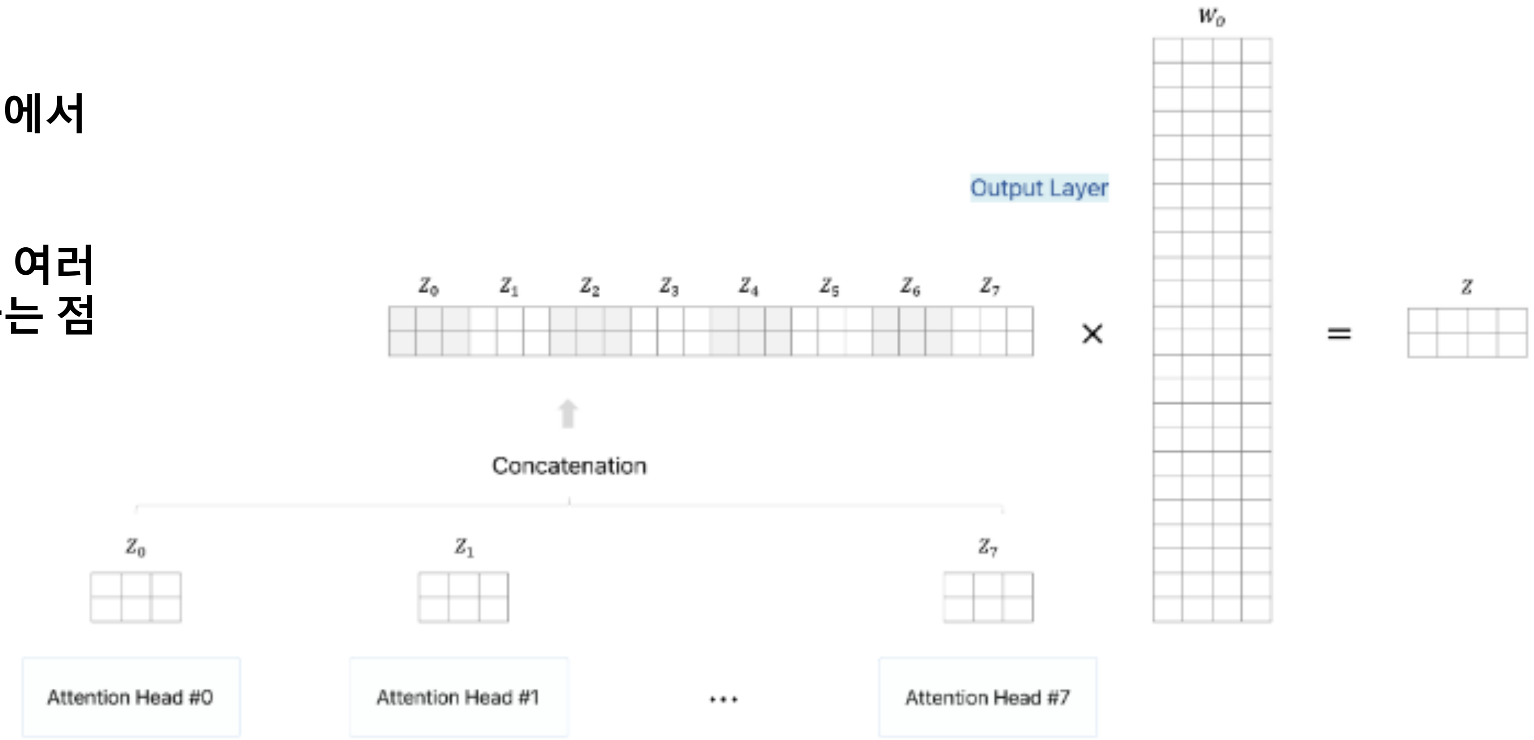
Cross-layer Parameter Sharing

Existing Model(Encoder)

Concat 후 원래 hidden state로 변경하는 과정에서 W0 파라미터 행렬 또한 필요.

**** 문제는 이러한 과정을 인코더가 쌓이면서 여러 번 반복하지만, 파라미터를 공유하지 않는다는 점**

If, Bertlayer를 12번 반복 -> 파라미터 개수가 12배로 증가.



TNT
TRAIN AND TEST

Model Architecture

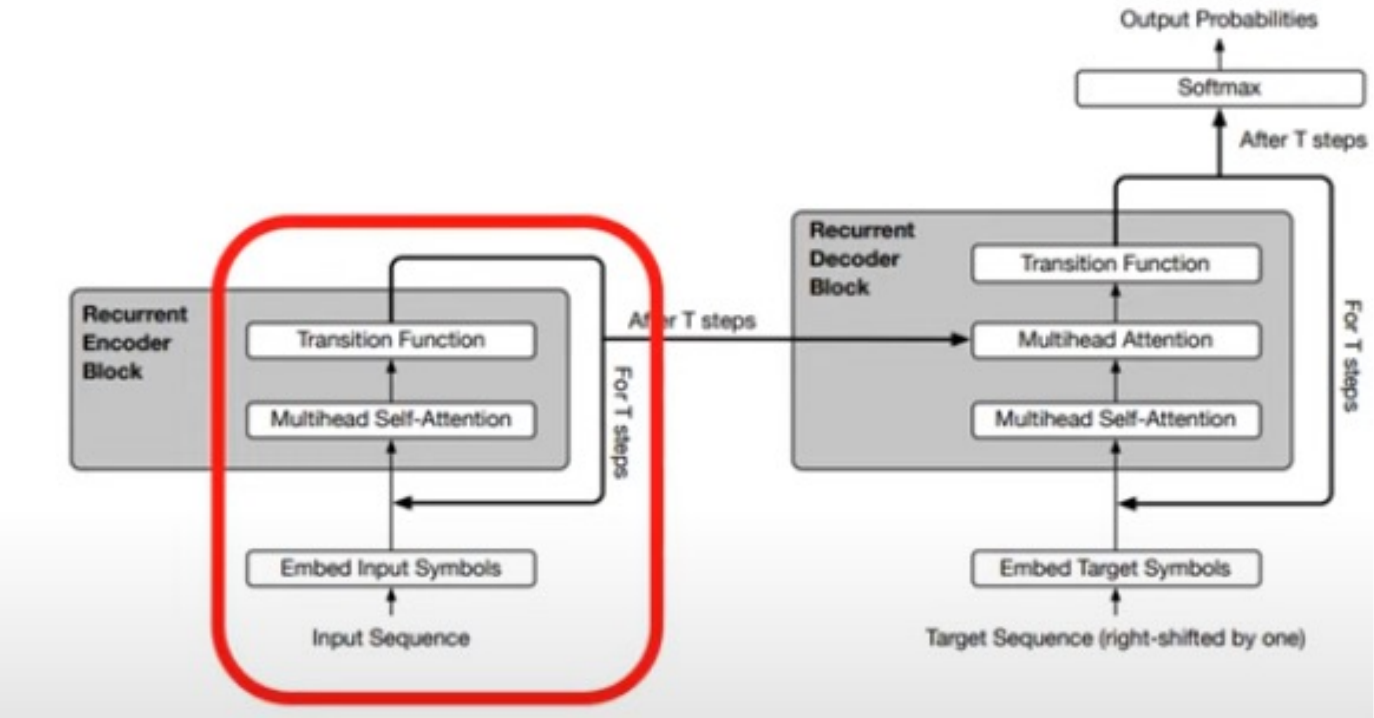
Cross-layer Parameter Sharing

AlBERT Model(Encoder)

파라미터 증가 문제
➔ 파라미터 공유로 해결

파라미터 공유의 구현
➔ Output이 Input으로 들어가는 재귀적 흐름

Layer가 12개라면 기존 BERT보다 1/12로 축소

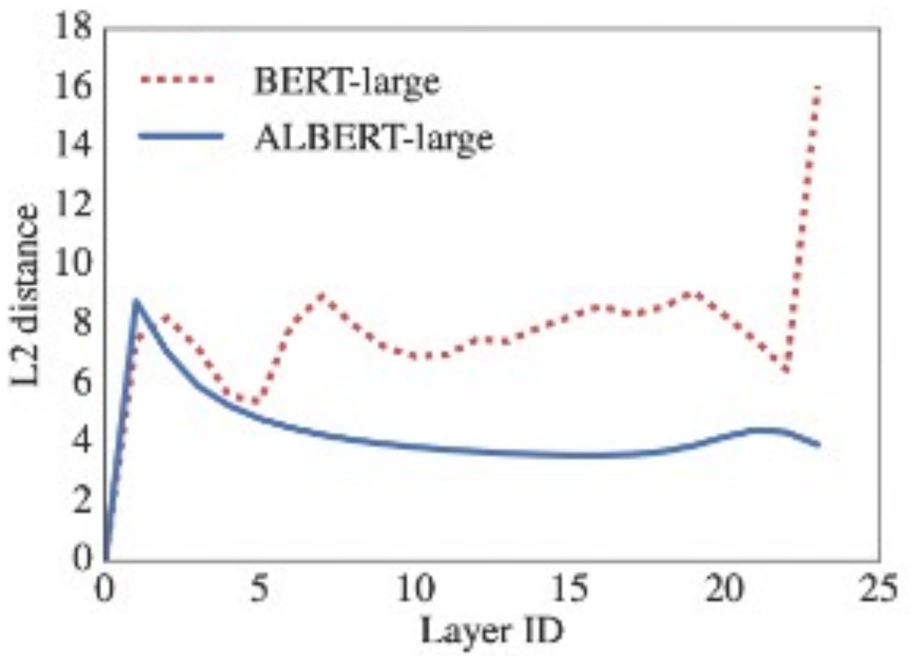


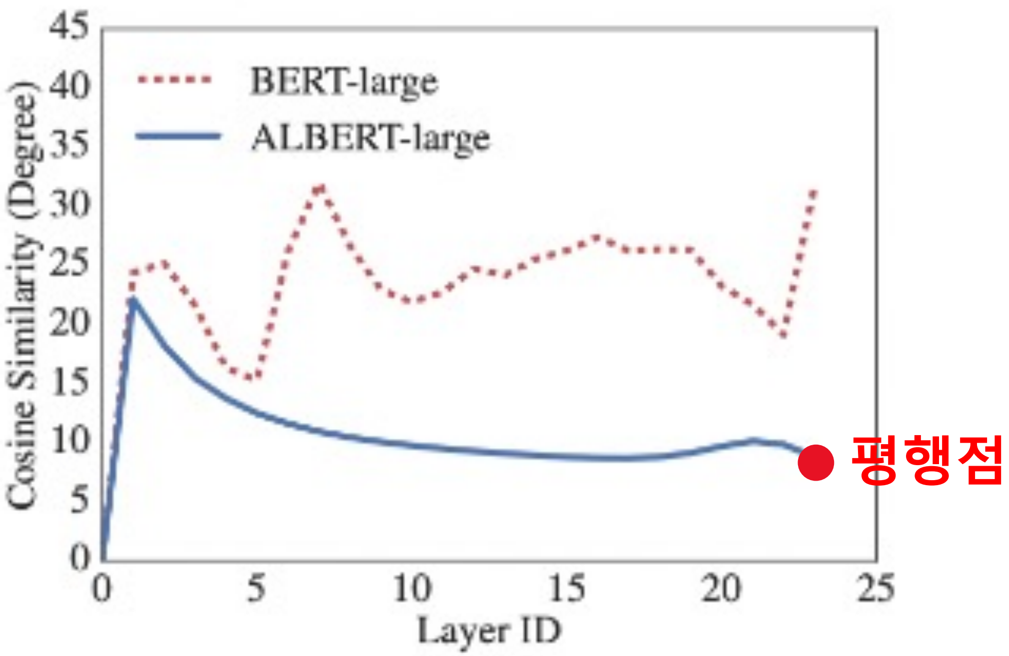
Model Architecture

Cross-layer Parameter Sharing

Is it work?

-> “Transitions from layer to layer are much smoother for ALBERT than for BERT





Model Architecture

Cross-layer Parameter Sharing

Shared-FFN

- W0 파라미터를 공유

Shared-attention

- Wq,k,v를 공유

All-shared

- 모든 파라미터 공유

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

Model Architecture

Cross-layer Parameter Sharing

Shared-FFN

- W0 파라미터를 공유

Shared-attention

- Wq,k,v를 공유

All-shared

- 모든 파라미터 공유

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

TRAIN AND TEST

Model Architecture

Sentence Order Prediction

BERT의 NSP

- Topic prediction + Coherence prediction / But, 후속연구(Yang et al., 2019; Liu et al., 2019)에서 신뢰성 이의 제기
- 서로 다른 문장에서 concat시키는 것은 내용이 겹칠 가능성이 적어 논리적 관계 학습에 큰 의미없을 수도 있음
- 같은 document에서 연속된 두 문장을 학습하는 것은 Overlap 가능성이 있음

ALBERT의 SOP

- Coherence prediction + Inter-sentence coherence / 두 문장 사이의 순서를 바꿔 올바른 순서인지를 예측하는 binary task

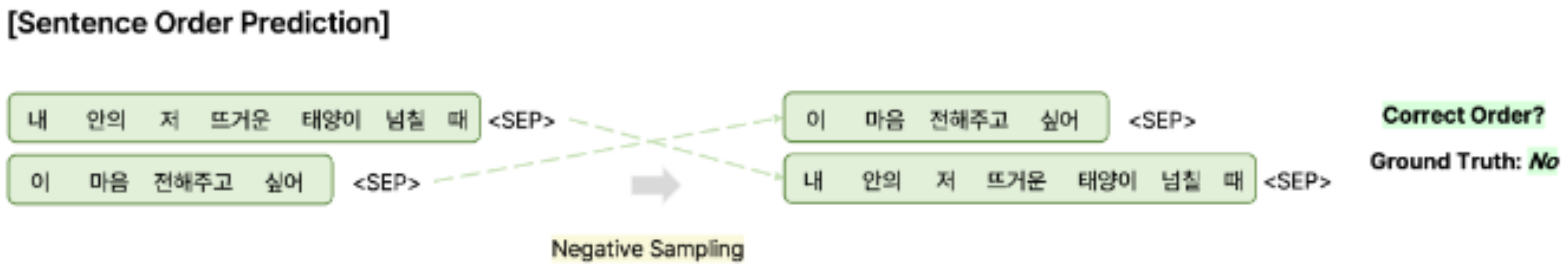
TRAIN AND TEST

Model Architecture

Sentence Order Prediction

ALBERT의 SOP

➔ 문장 순서를 바꾸면 0, 바꾸지 않으면 1



TRAIN AND TEST

Model Setup & Experimental Results.

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

Table 1: The configurations of the main BERT and ALBERT models analyzed in this paper.

ALBERT is 18x fewer parameters than BERT(18M, 334M)

TRAIN AND TEST

Model Setup & Experimental Results.

SETUP

-> pre-train corpora : BookCorpus, Wikipedia (16GB)

-> Maximum input length : 512

-> Vocab size: 30,000

-> Batch size : 4096

-> Lr : 0.00176

-> Optimizer : Lamb

Model Setup & Experimental Results.

Overall Comparison

ALBERT는 BERT보다 적은

Parameter로도 성능이 크게 개선

& 시간 단축 (3배)

Model		Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg	Speedup
BERT	base	108M	90.5/83.3	80.3/77.3	84.1	91.7	68.3	82.1	17.7x
	large	334M	92.4/85.8	83.9/80.8	85.8	92.2	73.8	85.1	3.8x
	xlarge	1270M	86.3/77.9	73.8/70.5	80.5	87.8	39.7	76.7	1.0
ALBERT	base	12M	89.3/82.1	79.1/76.1	81.9	89.4	63.5	80.1	21.1x
	large	18M	90.9/84.1	82.1/79.0	83.8	90.6	68.4	82.4	6.5x
	xlarge	59M	93.0/86.5	85.9/83.1	85.4	91.9	73.9	85.5	2.4x
	xxlarge	233M	**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).

Model Setup & Experimental Results.

Factorized Embedding Parameterization

Non-shared

- Embedding size가 항상 큰 게 좋은 것은 아님.

All-shared

- 128 size일 때 가장 좋음

-> E = 128을 사용

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.

TRAIN AND TEST

Model Setup & Experimental Results.

Cross-layer Parameter Sharing

All-shared

- E = 768, 128 모두 성능 하락.
- 하지만 감소된 연산량에 비해서 유의미한 수치는 아님.

	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.

Model Setup & Experimental Results.

Controlling Training Time

What if model learns at the same time?

- Avg : ALBERT > BERT
- 특히 RACE 같은 벤치마크에서 빠름

Models	Steps	Time	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
BERT-large	400k	34h	93.5/87.4	86.9/84.3	87.8	94.6	77.3	87.2
ALBERT-xxlarge	125k	32h	94.0/88.1	88.3/85.3	87.8	95.4	82.5	88.7

Discussion

➔ Less parameter, more performance.

But, computational cost increasing (Q!)

➔ To improve inference speed, sparse attention and block attention will become important.

➔ Researchers assume that SOP will be a great evaluation indicator. but, do not rule out the emergence of new loss function.

TRAIN AND TEST