A little bert

THE ELEMENTS OF ALBERT

Given the importance of model size, we ask: Is having better NLP models as easy as having larger models? 已经知道模型大小很重要，怎样建造更好的模型

提出新的方案

暴力增加隐藏层的数量，翻倍隐藏层

效果

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.

缺点

容易产生梯度消失

计算量非常大

想办法保持效果又减少参数数量

分成两部分
By decomposing the large vocabulary embedding matrix into two small matrices, we separate the size of the hidden layers from the size of vocabulary embedding. This separation makes it easier to grow the hidden size without significantly increasing the parameter size of the vocabulary embeddings

factorized embedding parameterization 分解

cross-layer parameter sharing 参数共享

This technique prevents the parameter from growing with the depth of the network

优化损失函数introduce a self-supervised loss for sentence-order prediction (SOP)

SOP primary focuses on inter-sentence coherence and is designed to address the ineffectiveness (Yang et al., 2019; Liu et al., 2019) of the next sentence prediction (NSP) loss proposed in the original BERT.

效果

We establish new state-of-the-art results on the well-known GLUE, SQuAD, and RACE benchmarks for natural language understanding. Specifically, we push the RACE accuracy to 89.4%, the GLUE benchmark to 89.4, and the F1 score of SQuAD 2.0 to 92.2.

The backbone of the ALBERT architecture is similar to BERT in that it uses a transformer en- coder (Vaswani et al., 2017) with GELU nonlinearities (Hendrycks & Gimpel, 2016). We follow the BERT notation conventions and denote the vocabulary embedding size as E, the number of encoder layers as L, and the hidden layer size as H . number of attention heads to be H/64.

改动1：Factorized embedding parameterization

bert不好的地方: embedding size E is tied with the hidden layer size H , i.e., E ≡ H

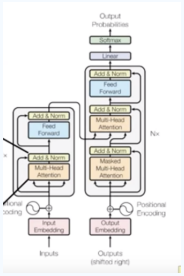
理由：From a modeling perspective, WordPiece embeddings are meant to learn context-independent repre- sentations, whereas hidden-layer embeddings are meant to learn context-dependent representations. 单个embedding更多的提供含义，而隐藏层的embedding更多地包含上下文的信息

untying the WordPiece embedding size E from the hidden layer size H allows us to make a more efficient usage of the total model parameters as informed by modeling needs, which dictate that H >> E.

Instead of projecting the one-hot vectors directly into the hidden space of size H , we first project them into a lower dimensional embedding space of size E, and then project it to the hidden space. By using this decomposition, we reduce the embedding parameters from O(V × H) to O(V × E + E × H)

q: query (to match others)
$$q^i = W^q a^i$$
k: key (to be matched)
$$k^i = W^k a^i$$
v: information to be extracted
$$v^i = W^v a^i$$

self attention



改动2:cross-layer parameter sharing

改动3：改变loss function
BERT uses an additional loss called next-sentence prediction (NSP)

NSP is a binary classification loss for predicting whether two segments appear consecutively in the original text, as follows: positive examples are created by taking consecutive segments from the training corpus; negative examples are created by pairing segments from different documents; positive and negative examples are sampled with equal probability. The NSP objective was designed to improve performance on downstream tasks, such as natural language inference, that require reasoning about the relationship between sentence pairs.

问题：negative sample 不够negative, 两个样本可能都不是同一个topic

改进方式：e maintain that inter-sentence modeling is an important aspect of language understanding, but we propose a loss based primarily on coherence. That is, for ALBERT, we use a sentence-order pre- diction (SOP) loss, which avoids topic prediction and instead focuses on modeling inter-sentence coherence. The SOP loss uses as positive examples the same technique as BERT (two consecu- tive segments from the same document), and as negative examples the same two consecutive seg- ments but with their order swapped.

Model		Parameters	Layers	Hidden	Embedding	Parameter-sharing
BERT	base	108M	12	768	768	False
	large	334M	24	1024	1024	False
	xlarge	1270M	24	2048	2048	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 2: The configurations of the main BERT and ALBERT models analyzed in this paper.

Note that for ALBERT- xxlarge, we mainly report results on a 12-layer network because a 24-layer network (with the same configuration) obtains similar results but is computationally more expensive

对比试验

对比对象

对比结果

dataset

downstream下游的结果：we evaluate our models on three popular bench- marks: The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018), two versions of the Stanford Question Answering Dataset (SQuAD; Rajpurkar et al., 2016; 2018), and the ReAding Comprehension from Examinations (RACE)

F1 and EM

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

Model	E	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
ALBERT base non-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

E 的选择

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	81.2
	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.

most of the performance drop appears to come from sharing the FFN-layer parameters, while sharing the attention parameters results in no drop when E = 128 (+0. 1 on Avg), and a slight drop when E = 768 (-0.7 on Avg).

共享参数

逐个验证

	Intrinsic Tasks			Downstream Tasks				
SP tasks	MLM	NSP	SOP	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
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.

loss SOP

Number of layers	Parameters	SQuAD1.1	SQuAD2.0	MNLI	SST-2	RACE	Avg
1	18M	31.1/22.9	50.1/50.1	66.4	80.8	40.1	52.9
3	18M	79.8/69.7	64.4/61.7	77.7	86.7	54.0	71.2
6	18M	86.4/78.4	73.8/71.1	81.2	88.9	60.9	77.2
12	18M	89.8/83.3	80.7/77.9	83.3	91.7	66.7	81.5
24	18M	**90.3/83.3**	**81.8/79.0**	83.3	91.5	**68.7**	**82.1**
48	18M	90.0/83.1	**81.8/78.9**	**83.4**	**91.9**	66.9	81.8

Table 7: The effect of increasing the number of layers for an ALBERT-large configuration.

网络深度