DistilBert

Created	@Jul 13, 2020 9:50 PM
Materials	https://arxiv.org/pdf/1910.01108.pdf
github	
참고 url	https://eagle705.github.io/articles/2019-11/DistilBERT https://jeongukjae.github.io/posts/2-distillbert-review/

Abstract

- NLP에서 large-scale pre-trained models에서 transfer learning 하는 것이 매우 보편화
- 그에 따라 제한된 환경에서 모델을 돌리기에 빡셈
- pre-training phase 할 때 knowledge distillation 하는 것으로 집중
- 모델 크기는 BERT 대비 40% 감소, 속도는 60% 빠르며, 97%의 성능을 유지함
- pre-training할 때, inductibe biases를 larger model로 부터 학습하기 위해 아래 3가지 loss를 제안
  - language modeling
  - distillation
  - cosine-distance losses

Introduction

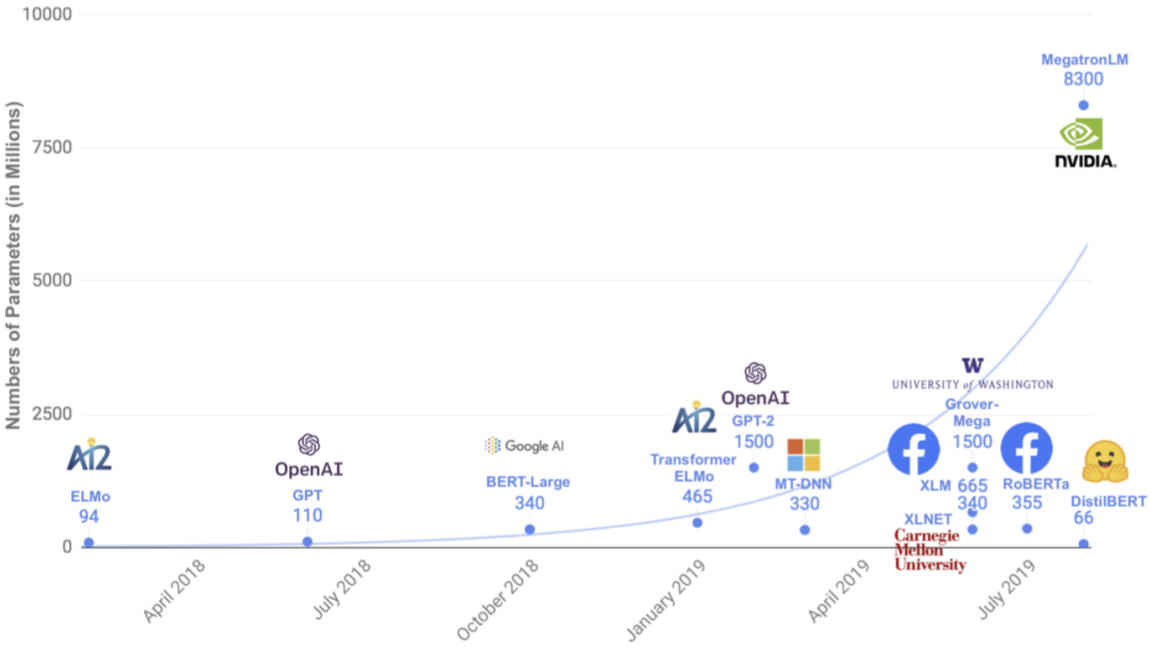

Figure 1: **Parameter counts of several recently released pretrained language models.**

- Transfer Learning이 아주 핫하고 있으며, large-scale pre-trained LM을 쓰는 것이 유행함
- 현재는 모델크기를 키울수록 downstream task에서 다 높은 성능이 나오고 있음
- 단점으로는
  - 환경적 비용이 문제
  - 계산 비용과 메모리 사용량이 높아서 다양한 곳에 적용하기 힘듬
- 논문에서 knowledge distillation으로 경량화된 LM 모델도 비슷한 성능을 낼수 있음을 증명

Knowledge distillation

knowledge distillation

모델을 압축하는 기술

- The student model(compact model)은 The teacher model (larger model)을 재생산하기 위해 학습
- 지도학습에서 분류모델은 instance class를 예측하기 위해 학습되는데, gold label의 estimated probability를 최대로 하게 끔 학습

- 보통의 traning objective는 모델이 예측한 확률 분포와 on-hot empirical distribution of traning labels의 cross-entropy를 최소화 함
- training-set에서 잘 동작하는 모델은 correct class에 대해서 output ditribution을 높은 값을 갖는 확률로 예측, 나머지는 거의 near-zero에 가까운 확률로 예측함
- near-zero에 가까운 확률 중 일부는 다른 것보다 더 확률이 높은데 model의 generalization capbilities를 반영하고 test-set에서 얼마나 잘 동작하는지 보여줌

Traning loss

- The student는 soft target probabilities of the teacher에 대한 distillation loss로 학습
- t_i는 teacher, s_i는 student

  $L_{ce} = \sum_i t_i * \log(s_i)$
- 이러한 모적함수는 the full teacher distribution을 활용할 수 있어서 충분한 training signal이 됨
- softmax-temperature를 사용

  $p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$
- T는 output distribution의 smoothness를 조절하는 term이고, z_i는 class i에 대한 model score를 뜻함
- 학습시에는 same temperature T를 student & teacher에게 적용하고 inference할 때는 T값을 1로 셋팅해서 standard softmax를 사용함
- the final training objective는 distillation loss L_{ce}를 the supervised training loss와 linear combination한 것임

DistillBert: a distilled version of BERT

Student architecture

- the student - DistilBERT는 BERT의 general architecture과 같음
- token type embedding 삭제
- pooler layer 삭제
- transformer block / 2

Student Initialization

- teacher model과 student model의 dim이 같기에 teacher model에서 레이어 두 개당 하나를 취해줌

Distillation

- RoBERTa 논문에서 나온 BERT를 학습 시키기 위한 best practices를 적용
  - gradient accumulation (up to 4K examples per batch)
  - dynamic masking
  - without NSP objective

Data and compute power

- DistilBERT는 original BERT와 같은 학습 corpus(English Wikipedia and Toronto Book Corpus)를 사용함
- 8개의 16GB V100 GPU로 90시간(약 3.5일) 정도 학습함 (RoBERTa의 경우 1024개의 32GB V100으로 하루 학습)

Experiments

Table 1: **DistilBERT retains 97% of BERT performance.** Comparison on the dev sets of the GLUE benchmark. ELMo results as reported by the authors. BERT and DistilBERT results are the medians of 5 runs with different seeds.

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	77.6	48.9	84.3	88.6	89.3	89.5	71.3	91.7	91.2	43.7
DistilBERT	76.8	49.1	81.8	90.2	90.2	89.2	62.9	92.7	90.7	44.4

Table 2: **DistilBERT yields to comparable performance on downstream tasks.** Comparison on downstream tasks: IMDb (test accuracy) and SQuAD 1.1 (EM/F1 on dev set). D: with a second step of distillation during fine-tuning.

Model	IMDb (acc.)	SQuAD (EM/F1)
BERT-base	93.46	81.2/88.5
DistilBERT	92.82	77.7/85.8
DistilBERT (D)	-	79.1/86.9

Table 3: **DistilBERT is significantly smaller while being constantly faster.** Inference time of a full pass of GLUE task STS-B (sentiment analysis) on CPU with a batch size of 1.

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

General Language Understanding

- DistilBERT의 language understanding과 generalization capabilities 를 확인하기 위해 General Language Understanding Evaluation (GLUE) benchmark 로 평가함
- 표 1에 9개의 task에 대해 정리했고 macro-scroe (average of individual score) 도 함께 표시함
- ELMo보다 항상 좋고 기존 BERT의 97% 성능을 냄 (40% 적은 파라미터로)

Downstream tasks

- 표2이며 fine-tuning도 fine-tune된 teacher를 이용한 것.

Size and Inference speed

- 표3이며 40% 적은 파라미터이며 60% 빠름

On device computation

- QA 모바일앱을 만듬
- 총 용량 207MB까지 낮췄으며 quantization이용하면 더 줄일 수 있다고 함