# distilBert

## a distilled version of BERT

smaller, faster, cheaper and lighter (Hugging face, 2019.10)

논문 : <https://arxiv.org/abs/1910.01108>

github : <https://github.com/huggingface/transformers/tree/master/examples/distillation>

해당 github에서 제안 방법으로 생성한 다른 distil\* 모델 소개 (DistilGPT2, DistilRoBERTa, DistilmBERT)

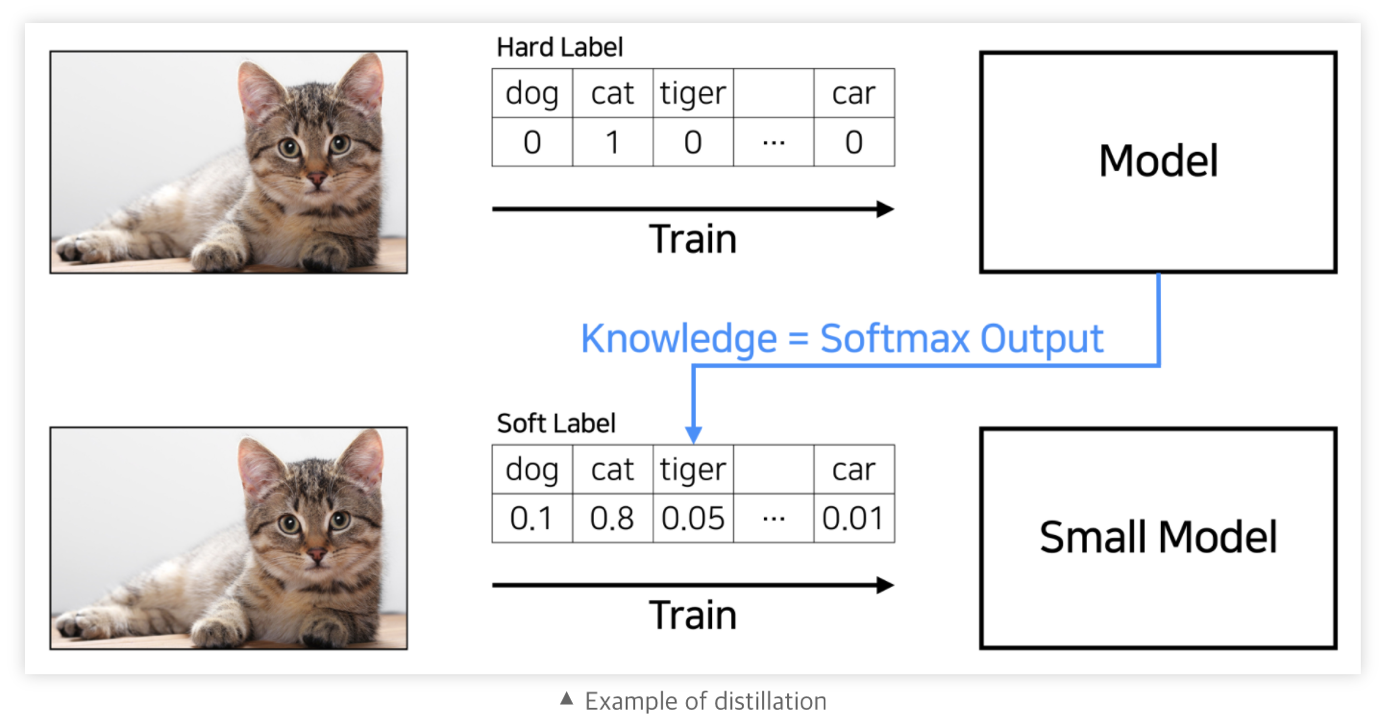
## abstract

* 최근 pretrain -> fine tuning으로 가는 방법이 많아지고 흔해졌지만, 모델 자체가 너무 크기 때문에 제한된 환경에서는 굉장히 사용하기 힘듬
* huggingface에서 DistilBert라는 general purpose language representation model 제안
* BERT를 40% 정도 줄이고 60%나 빠르게 연산하면서 97%의 성능을 유지

## distilation (knowledge extraction)

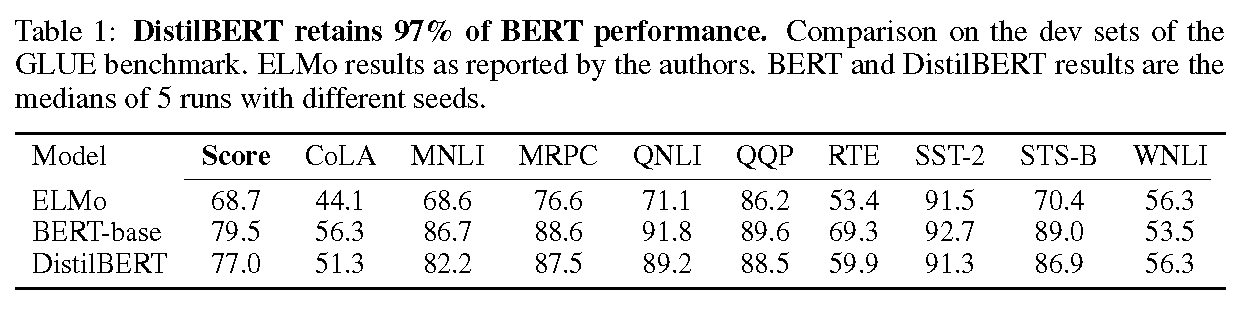
Knowledge Distillation [Bucila et al., 2006, Hinton et al., 2015]은 larger model(teacher model)로부터 compact model(student model)을 만들어내는 방법이다. 이게 작은 모델을 바로 학습시키는 것보다 의미있는 이유는 near-zero인 확률들도 학습할 수 있기 때문이다.

모델이 만들어낸 결과 분포가 실제 이미지를 더 풍부하게 표현한다는 것입니다. 즉, 원래 정답 관점에서는 정답 이외에 대한 정보가 없지만, 한번 모델에서 풀어나온 결과는 정답 외에도 다른 물체에 대한 정보를 담고 있게 됩니다. 이렇게 정보가 묻어 나오는 것이 마치 석유의 부산물들이 증류탑에서 나오는 양상과 같기 때문에 이를 지식 증류라고 일컫습니다. 원래 모델이 생각하는 데이터의 정보가 풀어나온 데이터로 새로 학습시키게 되면 간접적으로 선생 모델이 학습한 바를 반영하게 되므로 더 효율적으로 모델을 학습시킬 수 있으며, 새로 배우는 모델(학생 모델)은 상대적으로 더 적은 규모로 구성될 수 있습니다. 이것이 Hinton et al. 이 제시한 지식 증류의 핵심입니다.



## distilbert

* architecture
  + token-type embedding and the pooler are removed
  + layers is reduced by a factor of 2
    - variations on the last dimension of the tensor(hidden size dimension) have a smaller impact on computation efficiency)
* initialization
  + important to find the right initialization for the sub-network to converge
  + initialize the student from teacher by taking one layer out of two
* distilation
  + large batches (up to 4K)
  + dynamic masking without NSP
* data and computer power
  + same data with BERT
  + on 8 16GB V100 GPUs for approximately 90 hours
* downstream benchmark
  + 97% of performance with 40% fewer parameter comparing with BERT



* distilation 방법 (github)

## loss function

* Loss function



* + Teacher 모델에서 나온 class probabilities를 바로 Student model의 target (soft target) 이용, knowledge를 효과적으로 transfer하는 방식
  + MNIST를 예를 들어 2를 맞출 때 닮은 숫자인 3과 7도 낮은 확률이지만 값을 부여, 이러한 정보들은 data의 structure에 대한 정보가 들어있는 값이기 때문에 중요 정보
  + 굉장히 낮은 값으로 표현되기 때문에 temperature 개념을 도입
* softmax-temperature



* + T는 Temperature이고, T=1이라면 보통의 softmax 식
  + T가 커지면 훨씬 soft한 probability distribution
  + T가 output distribution의 smoothness를 결정. training 동안에만 T를 조정하고 inference 시간에는 1로 설정해서 standard softmax로 사용
* final training loss는 distillation loss Lce와 BERT에서 사용한 Lmlm의 linear combination
* Softmax(소프트맥스)

## distil\*

* distilbert는 bert-base-uncased tokenizer와 같은 tokenizer
* teacher model과 비슷하게 불러오는 방식 (bert, gpt2, roberta, betlm..)
* DistilBERT uncased: model = DistilBertModel.from\_pretrained('distilbert-base-uncased')
* DistilGPT2: model = GPT2Model.from\_pretrained('distilgpt2')
* DistilRoBERTa: model = RobertaModel.from\_pretrained('distilroberta-base')
* DistilmBERT: model = DistilBertModel.from\_pretrained('distilbert-base-multilingual-cased')

## 참고

[Transformer - Harder, Better, Faster, Stronger](https://blog.pingpong.us/transformer-review/#distilbert-knowledge-distillation)

[🏎 Smaller, faster, cheaper, lighter: Introducing DilBERT, a distilled version of BERT](https://medium.com/huggingface/distilbert-8cf3380435b5)

[Distilling the Knowledge in a Neural Network](https://arxiv.org/abs/1503.02531)

[📃 Distilling the Knowledge in a Neural Network 리뷰](https://jeongukjae.github.io/posts/distilling-the-knowledge-in-a-neural-network/)