###BERT 관련 NLP 관련자료 정리

#1106

#BERT 오리지날 GIT HUB

https://github.com/google-research/bert

#https://mc.ai/a-guide-to-simple-text-classification-with-bert/

#버트 간단 소개

How bert works: a brief overview

#MAX\_LEN이 큰 경우

If you get an out of memory error, you may need to run this on a machine with a GPU that has more on-board RAM or a TPU (see instructions for TPUs in the bert github repo). You can try to fix this issue by reducing the training\_batch\_size, though the training will run slower as a result. If your typical text is longer than 128 words, you can increase max\_seq\_length up to a max of 512, though the model will run slower if you do this and you may get an out of memory error.

--> TPU 필요성

#구글 공식 깃헙에서는 문서 분류 테스크가 너무 불친절

#괜찮은 코랩 가이드 발견 하지만 너무 자세함. 불-편

https://colab.research.google.com/github/google-research/bert/blob/master/predicting\_movie\_reviews\_with\_bert\_on\_tf\_hub.ipynb

#좀 더 간단하게 API화해서 이용가능한 코랩 노트북

블로그:https://towardsdatascience.com/how-to-do-text-binary-classification-with-bert-f1348a25d905

코랩: https://github.com/wshuyi/demo-text-binary-classification-with-bert/blob/master/bert\_text\_classification.ipynb

#다만 중국인인듯 한자 잔뜩.. 불-편

#DocBERT

Finetuning the pre-trained BERT models for Document Classification tasks.

#이 논문에 따르면 그냥 bert는 문서 분류에 최적화되어있지 않음

(그 이유는 bert의 학습 방향 자체가 분류랑 큰 연관이 없다는 것)

따라서 doc classification을 위해 최적화해줘야 한다.

(그 방법은 아직 완벽히 파악이 안됨.. 대략적으로 파라미터만 건든 거 같은데

그걸로 성능이 향상된다는게 잘 납득 되지는 않음)

블로그:https://www.groundai.com/project/docbert-bert-for-document-classification/1

논문:https://arxiv.org/abs/1904.08398v1

깃헙:https://github.com/castorini/hedwig/tree/master/models/bert

#kobert를 활용KoBERT와 CRF로 만든 한국어 객체명인식기 (BERT+CRF based Named Entity Recognition model for Korean)

https://github.com/eagle705/pytorch-bert-crf-ner

####colab에서 tpu로 bert 학습 시키기

https://blog.nerdfactory.ai/2019/04/25/learn-bert-with-colab.html

2. TPU로 학습하기

주의 사항 : Colab의 TPU와 Google Cloud Platform의 TPU는 다릅니다. 이 글에서는 오로지 무료인 Colab의 TPU만 사용하기 때문에

절대 Google Cloud Platform의 TPU 노드를 절대 생성하지 마세요!

(TPU 노드 생성 후 인스턴스 활성화 상태만으로 크레딧이 지불됩니다.)

#다만 데이터들을gcp storage에 넣어야 한다.

실험 parameter는 이 글에 작성된 기본 설정으로 동일하게 맞춰져 있습니다.

batch\_size = 16 learning\_rate = 3e-5

num\_train\_epochs = 2.0 조건에서

총 학습 시간(분) gpu: 172분 tpu: 약 28분

##그 외 tk hub 방식의 tpu 사용 가이드 코랩 노트북 찾음

https://colab.research.google.com/github/tensorflow/tpu/blob/master/tools/colab/bert\_finetuning\_with\_cloud\_tpus.ipynb

#####bert 학습시킬때 keras generatior api처럼 셔플 옵션은 없는가?

####찾아본 결과 많은 케이스에서 셔플을 하지 않았을떄

bert을 통한 문서 분류가 성능이 크게 저하되고

특정 클래스만 예측하는(특히 이진분류에서) 문제가 발생했다고 한다.

관련 레딧:https://github.com/google-research/bert/issues/154

case1)

INFO:tensorflow: eval\_accuracy = 0.64309764

The model performs perfectly after shuffling the dataset. Thanks! I am closing it.

NFO:tensorflow: eval\_accuracy = 0.97306395

case2:)

Thanks for your help @yanfan0531 ..Shuffling solved my issue and bert is now giving correct predictions. I read a bit more and shuffling is usually a pre requisite for training a model in neural networks.

##gluon api에서 발견한 셔플 관련 코드

이걸 내 코드에 넣을 수 있는가?

https://gluon-nlp.mxnet.io/examples/sentence\_embedding/bert.html

# The FixedBucketSampler and the DataLoader for making the mini-batches

train\_sampler = nlp.data.FixedBucketSampler(lengths=[int(item[1]) for item in data\_train],

batch\_size=batch\_size,

shuffle=True)

bert\_dataloader = mx.gluon.data.DataLoader(data\_train, batch\_sampler=train\_sampler)

내 코드에서 해당 부분을 반영할 수 있는 부분을 찾아서 수정해줬음!

과연 학습에 영향이 갈 것인가? (현재 학습이 안되는 상황)

batch\_size = 8

lr = 5e-5

train\_dataloader = mx.gluon.data.DataLoader(data\_train, batch\_size=batch\_size, num\_workers=5, shuffle=True)

test\_dataloader = mx.gluon.data.DataLoader(data\_test, batch\_size=int(batch\_size/2), num\_workers=5)

###챗봇과 직접적인 연결

Sequence Generation with Sampling and Beam SearchDOWNLOAD THIS TUTORIAL

#간단하게 gluon api에서 lm을 갖고 문장 생성 하는 부분인데

bert도 사용가능한지 확인해야함

https://gluon-nlp.mxnet.io/examples/sequence\_sampling/sequence\_sampling.html

##########bert 개념

#BERT 논문정리

##bert관련 논문 정리하면서 자세한 설명해주는 블로그

https://mino-park7.github.io/nlp/2018/12/12/bert-%EB%85%BC%EB%AC%B8%EC%A0%95%EB%A6%AC/?fbclid=IwAR3S-8iLWEVG6FGUVxoYdwQyA-zG0GpOUzVEsFBd0ARFg4eFXqCyGLznu7w

내용:

BERT의 아키텍처는 Attention is all you need에서 소개된 Transformer를 사용하지만, pre-training과 fine-tuning시의 아키텍처를 조금 다르게하여 Transfer Learning을 용이하게 만드는 것이 핵심입니다.

BERT는 transformer 중에서도 encoder 부분만을 사용합니다

BERT\_base 모델의 경우, OpenAI GPT모델과 hyper parameter가 동일합니다. 여기서 BERT의 저자가 의도한 바는, 모델의 하이퍼 파라미터가 동일하더라도, pre-training concept를 바꾸어 주는 것만으로 훨씬 높은 성능을 낼 수 있다는 것을 보여주고자 하는 것 같습니다.

OpenAI GPT모델의 경우 그림1에서 볼 수 있듯, next token 만을 맞추어내는 기본적인 language model 방식을 사용하였고, 그를 위해 transformer decoder를 사용했습니다. 하지만 BERT는 MLM과 NSP를 위해 self-attention을 수행하는 transformer encoder구조를 사용했음을 알 수 있습니다.

실제로 대부분의 NLP task SOTA는 BERT\_large모델로 이루어 냈습니다.

#bert를 tf hub로 사용하는 코랩 코드 (나중에 한번 뜯어봐야 확실한 이해 가능할듯)

https://colab.research.google.com/github/google-research/bert/blob/master/predicting\_movie\_reviews\_with\_bert\_on\_tf\_hub.ipynb#scrollTo=FnH-AnOQ9KKW

##################흥미로워보이지만 탐색할 시간이 없는 것들

#rasa\_nlu\_bert

https://github.com/winstarwang/rasa\_nlu\_bert

#Distilling the Knowledge in a Neural Network

#https://jamiekang.github.io/2017/05/21/distilling-the-knowledge-in-a-neural-network/

대략 내용:

이 논문은 크게 두 부분으로 나뉘어집니다.

Model Compression: 1번 ensemble 모델의 지식을 2번 모델로 전달하는 방법

Specialist Networks: 작은 문제에 특화된 모델들을 training시켜 ensemble의 training 시간을 단축하는 방법

저자들은 그런 형태의 결과를 hard target이라고 부르면서, 1과 0으로 변환되기 전의 실수 값(soft target)을 사용하는 것이 2번 모델로 지식을 전달하는데 효과적이라고 주장합니다.

#다른 블로그

https://blog.lunit.io/2018/03/22/distilling-the-knowledge-in-a-neural-network-nips-2014-workshop/

내용:

Distillation (증류) 는 불순물이 섞여 있는 혼합물에서 원하는 특정 성분을 분리시키는 방법을 말합니다. 본 논문에서 말하는 neural network에서의 knowledge distillation은 불필요하게 많은 parameter가 사용되는 (불순물이 많이 섞인) ensemble model로부터, generalization 성능을 기존 대비 향상시킬 수 있는 knowledge들을 분리해 내는 방법을 말합니다.

Conclusion

거대한 ensembled model이 가지고 있는 좋은 knowledge를 비교적 간단하게 해당 모델의 softmax output을 활용해서 small single network에 전달해 줄 수 있는 knowledge distillation 방법은 꽤나 좋은 아이디어이고, 활용도도 높을 거라고 예상됩니다.

하지만 이러한 방법이 MNIST라는 작은 데이터에서 실험을 진행해서 검증을 하였고, 추후에 나온 여러 논문들에서 더욱 복잡한 문제와 deeper network 환경에서는 softmax output만으로 knowledge를 전달하기에는 무리가 있다는 의견도 있습니다.

최근에는 본 논문처럼 단순히 bigger model의 정보를 smaller model로 전달하는 방법 뿐만 아니라, 약간의 변형으로 기존에 학습된 정보를 그대로 유지하면서도 추가적인 데이터로부터 학습을 진행하기 위한 learning without forgetting [1], model ensemble 대신 data transformation 으로부터 knowledge를 추출해 전달하는 data distillation [2] 과 같은 본 논문에 기반한 여러 후속 연구들이 나오고 있으므로 이러한 후속 연구들을 같이 살펴보는 것도 괜찮을 것 같습니다.

#####pretrain된 elmo를 이용해 text의 features 추출 (features based) 방식인듯?

https://gluon-nlp.mxnet.io/examples/sentence\_embedding/elmo\_sentence\_representation.html