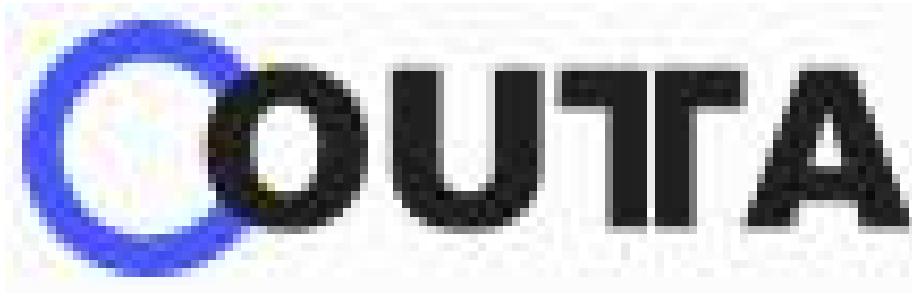


REPORT



OUUTA 감정인식 프로젝트 레포트

팀장 :김동규

팀원 :오도열

왕건우

현시은

우리 팀은 처음에는 CNN의 데이터 셋을 보았을 때 불균형이 심한 것을 인지하였다. 그래서 data augmentation에서 randomaffine으로 회전, 병진 이동, 크기를 변형시켰고 (4)인 augmix를 통해서 데이터를 증감시켰다. augmix 또한 데이터 셋의 분포가 일치하지 않을 때 불확실성 추정을 얻을 수 있다고 하여서 사용했다. CNN 모델은 두 가지를 실험을 해보았습니다. (1)과(2)를 합쳐서 모델을 만들었고 (1)의 layer들에다가 (2)에서 사용한 트릭들인 batch_norm->relu->maxpooling->dropout의 형태로 짜서 만들어 주었다. 그리고 optimizer는 (3)에서의 정보를 가져와서 SGD Nesterov를 사용하였고 learning rate는 1e-2로 학습시켰다 LR Schedule 또한 (3)을 통해서 ReduceLROnPlateau를 사용하여서 학습시켰다. 마지막으로 loss 함수는 facial loss 함수를 사용하였다. 이 loss 함수는 imbalanced dataset에 대해서 좋은 결과를 낸다고 하여서 사용하였다. 그렇게 학습하여 150epochs를 통해서 0.66 정확도를 달성했다. 그리고 두 번째 모델은 vgg16모델을 사용하였으며 사용 시에는 (1,224,224)의 사진으로 학습된 모델이기에 resize를 통해서 (48,48)->(224,224)로 바꾸어서 학습시켜주었고 다른 사항은 모두 같게 실시하였다.

NLP 모델에서 다른 조건 없이 forward에 bert layer와 linear layer만 적용하고 learning rate를 1e-4, 1e-5, 5e-5, 1e-6로 설정하고 epochs를 10회로 학습을 시켜보았을 때 5e-5에서 가장 높은 0.47부근의 정확도를 얻을 수 있었다. 또한 학습을 진행하면서 epochs 6~7회부터 값이 크게 달라지지 않아 epochs는 10회가 적당하고 결론을 내렸다.

옵티마이저를 AdamW에서 SGD로 변경하여 학습시켜보았는데 겨우 0.17 정도의 정확도만 얻을 수 있었다. 확실히 nlp 모델을 학습할 때 SGD보단 AdamW가 좋은 정확도를 얻는 옵티마이저라는 걸 알 수 있었다. 또한 정확도를 높이기 위해 dropout 기법을 forward에 추가시키고 warmup_ratio=0.1, num_epochs=5를 따로 설정해 주고 총 훈련 step을 num_data*epochs에서 num_data*num_epochs로 변경하고 warmup_step을 total_steps*warmup_ratio로 변경해서 스케줄러에 변경된 warmup_step total_steps 값을 추가시켜 주었다. learning rate의 값은 5e-5, epochs의 값은 10으로 고정하고 dropout의 각 p 값을 0.2,0.3,0.4,0.5 정도로 학습을 시도해 보았을 때 0.3~0.4에서 0.49 후 반대의 정확도를 얻을 수 있어 정확도가 향상된 모습을 볼 수 있었다. 또한 옵티마 이전에 weight_decay의 값을 0.01로 설정하여 학습을 진행시켰는데 이것 또한 0.49 정확도를 얻어 결과에 큰 영향을 미치지 않는 것을 확인할 수 있었다.

bert 모델에 linear layer와 dropout(p=0.3,0.2)로 두 번씩 적용시켰는데 오히려 정확도가 0.48로 더 떨어졌고 bert 모델에 LeakyRelu layer를 사용해 봤지만 이것 또한 0.48 근처의 정확도만 나왔다.

leakyRelu를 유지한 채로 CosineAnnealingWarmRestarts 스케줄러를 사용하여 학습을 시켰지만 0.43의 정확도를 얻어 별 효과를 보지 못했다.

또한 loss 함수를 Focalloss라는 새로운 함수로 사용해도 보았지만 원하는 만큼의 정확도를 얻지는 못했다. 최대 0.507정도를 얻었고, 기대에 못미치는 정도였다.

그래서 이번에는 중간미션때 사용했던 BertForSequenceClassification을 사용해보았다. 그러자 확실히 더 보다 더 높은 정확도를 얻을 수 있었다.

BertForSequenceClassification을 사용할 때 조심해야 하는 것은, 이 모델은 결과값이 저장된 텐서값을 튜플 형태로 넣어서 return해준다는 것이다. 따라서 output[0]과 같은 방식으로 튜플 안의 결과값을 꺼내주어야 CrossEntropyLoss 함수에 넣거나 label과 비교하여 accuracy 연산이 가능할 수 있었다. 즉, loss와 accuracy의 계산을 위해 어떤 자료형으로

값이 return되는지 꼼꼼하게 확인이 필요했다.

BertForSequenceClassification와 스케줄러 중 하나인 CosineAnnealingWarmRestarts 스케줄러(get_cosine_schedule_with_warmup라는 것을 통해 import 한다. 그리고 crossentropy 형식으로 loss를 계산하는 것이 가장 좋은 accuracy를 얻을 수 있었다. 그 결과 0.5626의 accuracy를 얻을 수 있었다.

<얼굴 인식 관련 - DNN 모듈>

얼굴 인식의 정확도를 향상시키고자 기존에 주어진 CascadeClassifier에서 OpenCV가 제공하는 DNN Module로 변경하였다. DNN Module은 단순한 object detect 모듈의 한계를 파훼하고자 도입되었다. DNN 모듈은 파이썬 환경뿐만 아니라 c/c++ 환경에서도 동작할 수 있기 때문에, 프로그램 이식성이 높다는 장점이 존재한다. 이 DNN 모듈에 이미지 프레임을 입력하면, 인식된 얼굴의 x, y, w, h를 반환한다. 물론 다른 정보(confidence 등)도 반환하지만, 프로그램을 제작하는데 필수적이라고 판단되지 않아 사용되지 않았다. DNN 모듈이 파라미터로는 기존 파일을 그대로 사용하였다.

(1) <https://arxiv.org/ftp/arxiv/papers/2206/2206.09509.pdf>

(2) https://assets.researchsquare.com/files/rs-511221/v1_covered.pdf?c=1631867304

(3) <https://arxiv.org/ftp/arxiv/papers/2105/2105.03588.pdf>

(4) <https://arxiv.org/pdf/1912.02781.pdf>

(5)

https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertConfig.attention_probs_dropout_prob

(6)

<https://discuss.pytorch.org/t/is-this-a-correct-implementation-for-focal-loss-in-pytorch/43327/21>

<출석 인증>

NLP - Google Drive | nlp_train_final.ipynb - Colaboratory | Meet - qrz-xmdc-zun

meet.google.com/qrz-xmdc-zun

건우
왕건우

동규
김동규

도열
오도열

시은
나

도열
오도열님이 참여했습니다.

Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증하십시오.

11:26 오전 | qrz-xmdc-zun

검색하려면 여기에 입력하십시오.

OUTTA 12조 (6)

환영합니다. 'OUTTA 12조' 오픈채팅방입니다.

감사합니다.

오후 4:30

연호 이현수

이 부분은 다시 한 번 공지드리는것이 좋을거 같아 질문 횟수 제한하지 않도록 하겠습니다.

오후 4:32

김동규

넵 감사합니다-

오후 4:33

김동규

<https://meet.google.com/ocm-hsvz-ow>

오후 10:03

Meet
Google에서 제공하는 실시간 ...
meet.google.com

2022년 8월 19일 금요일

김동규

저희 금일 리포트작성할까요?

오후 9:45

왕건우

nlp쪽은 지향 시은님이 쓰고 vision쪽은 동규님이랑 도열님이 쓰시는게 좋을것 같은데
어떠신가요??

오후 10:24

Windows 정품 인증
[설정]으로 이동하여 Windows를 정품 인증하십시오.

OUTTA 12조 (6)

원영합니다. 'OUTTA 12조' 오픈채팅방입니다.

미니 배지 부분이 이개념같은데 지금 현재 사용은 하는게 맞는지 같습니다

저는 여기서 막혔네요 오후 8:01

chrome-extension://efaidnbmnnnlpcapcgpcjcllndmkaj/https://opscience.iop.org/article/10.1088/1757-899X/1042/1/012027/pdf 오후 8:57

https://meet.google.com/tzq-yjgc-pme

Meet
Google에서 제공하는 실시간 ...
meet.google.com

오후 9:00

```
from torch import nn
from transformers import BertModel, BertConfig

class BertClassifier(nn.Module):

    # BertClassifier는 초대용량 AI 모델인 BertModel을 파인튜닝한 모델이다.
    # 구체적으로 BertModel 위에 linear층 중 하나를 추가해 높은 모델이다.
    # Bert -> Linear -> Softmax의 순서로 연산이 이루어진다.
    # 우리가 Loss를 구하는데 사용되는 CrossEntropyLoss의 경우, softmax를 자동으로 포함하여 연산해준다.
    # 그러, BertClassifier 클래스의 내의 forward 함수에, softmax는 포함시키지 않는다.
    # 단, 학습의 손도가 아니라 테스트나 실제 시연할 때 forward 결과물에 softmax를 적용시켜줘야 한다.

    def __init__(self, num_L_
```

Windows 정품 인증
[실정]으로 이용하여 Windows를 정품 인증합니다.

OUTTA 12조 (6)

원영합니다. 'OUTTA 12조' 오픈채팅방입니다.

Transformers
BERT
여기를 눌러 링크를 확인하세요.
huggingface.co

bertconfig 관련 함수찾아보았는데
dropout을 그냥 0으로 하는거는 어떤가요?
아닌가요?
데이터를 너무 버리는거 아닌가 싶어서 그랬습니다 오후 10:47

이게 진짜 예상 못했어서
해보긴 해야겠네요
 일단 LSTM 찾는거 성공해서
이거 해보고 해보겠습니다
오후 10:48

왕건우
focal loss쓰니까 확실히 loss가 엄청줄어드네요 오후 10:54

왕건우
근데 loss는 줄어드는데 accuracy가 변함이 많지 않아요 ㅠㅠ
오후 10:59

Windows 정품 인증
[실정]으로 이용하여 Windows를 정품 인증합니다.

