

[P-Stage level 1] Mask Image Classification 개인 회고

- 2021.09.03 (금)

네이버 부스캠프 P-Stage level 1 Mask Image Classification 대회 마무리 회고 글입니다.

1. 시도했던 방법 들

1.1. Data Preprocessing

- 학습용 데이터에서 남자인데 여자로 분류되어 있는 데이터(6명), 여자인데 남자로 분류되어 있는 데이터(2명), 마스크를 잘 못 쓰고 있는데 정상이라고 되어있는 데이터(3명) 등 총 11명의 id 를 가진 사람들의 라벨을 고쳐주었다.
- facenet 을 사용해서 학습용 데이터와 테스트용 데이터에서 얼굴 부분만 추려냈다. 이 때, 얼굴 부분을 못찾는 경우에는 수동으로 적당한 크기를 잡아서 Crop 을 해서 새로운 경로에 Crop 한 이미지를 저장했다.
- augmentation 을 위해서 pytorch Transform 활용
 - train set 적용

```
Resize(resize, Image.BILINEAR),
RandomChoice([ColorJitter(brightness=(0.2, 3)),
               ColorJitter(contrast=(0.2, 3)),
               ColorJitter(saturation=(0.2, 3)),
               ColorJitter(hue=(-0.3, 0.3))]),
RandomHorizontalFlip(p=0.5),
ToTensor(),
Normalize(mean=mean, std=std),
```

- valid & train set 적용

```
Resize(resize, Image.BILINEAR),
ToTensor(),
Normalize(mean=mean, std=std),
```

- 학습용으로 주어진 train data image 를 train / valid 로 잘 구분해야 하는데 이 때, `train_test_split` or `Stratified K-Fold` 를 적용해서 성별과 나이의 비율에 맞게 train / valid set 으로 나눴다.
 - 이 때, valid set 에 train 에서 학습한 사람의 얼굴이 들어가지 않게 되서 valid score 가 LB score 와 비슷해짐

1.2. Model

- timm 라이브러리를 이용해서 pretrained 되어 있는 모델들을 주로 불러와서 사용했다.
 - resnet18, resnet50, resnet101
 - efficientnet_b1_pruned, efficientnet_b2_pruned, efficientnet_b3_pruned
 - efficientnet_b0, efficientnet_b1, efficientnet_b2, efficientnet_b3, efficientnet_b4
 - seresnext26d_32x4d, seresnext26t_32x4d, seresnext50_32x4d
 - nfnet_l0
 - 등

1.3. Loss (criterion)

- Baseline 으로 주어지는 4개의 loss 와 추가적으로 하나 더 사용했다.
 - F1Loss
 - CrossEntropyLoss
 - LabelSmoothingLoss
 - FocalLoss
 - SymmetricCrossEntropyLoss

1.4. Optimizer

- 알고 있는 Optimizer 중에 가장 잘 나오는 Adam 을 주로 사용하다가 구글링을 통해 알게 된 MADGRAD 를 적용했다.
 - SGD
 - Adam
 - AdamW
 - MADGRAD

1.5. Scheduler

- Baseline 에 주어진 StepLR 을 주로 사용하다가 멘토님에게서 Scheduler 가 중요하다는 얘기를 듣고 멘토님이 추천해준 CosineAnnealingLR 을 사용해보다가 이번 대회의 경우에는 보통 긴 epoch 을 돌지 않아도 되기 때문에 lr 을 점차 줄여주는 CyclicLR 을 사용했다.
 - StepLR
 - CosineAnnealingLR
 - CyclicLR

1.6. Tensorboard

- Baseline 코드를 거의 건드리지 않았고 학습을 진행하면서 학습이 잘 진행되는지 확인했다.
 - Train
 - acc, f1_score, loss, learning_rate
 - Valid
 - acc, f1_score, loss

1.7. Cutmix

- 5주차에 나온 special mission 에서 공유해주신 [Cutmix 사용 예제](#) 를 보고 참고해서 Train 함수에 적용

```
if np.random.random() <= args.cutmix:
    W = inputs.shape[2]
    mix_ratio = np.random.beta(1, 1)
    cut_W = np.int(W * mix_ratio)
    bbx1 = np.random.randint(W - cut_W)
    bbx2 = bbx1 + cut_W

    rand_index = torch.randperm(len(inputs))
    target_a = labels
    target_b = labels[rand_index]

    inputs[:, :, :, bbx1:bbx2] = inputs[rand_index, :, :, bbx1:bbx2]
    outs = model(inputs)
    loss = criterion(outs, target_a) * mix_ratio + criterion(outs, target_b) * (1. - mix_ratio)
else:
    outs = model(inputs)
    loss = criterion(outs, labels)
```

1.8 Cross Validation

- 5 Fold 를 valid set 으로 학습에 사용되지 않는 20% 데이터들도 사용할 수 있게끔 했다.
- 학습시간이 5배로 늘어나지만 성능이 올라가기에 쓸 수 밖에 없는 방법이었다.

1.9 Ensemble

- hard voting & soft voting 방법을 적용해서 inference 를 사용했다.

2. 학습과정에서의 교훈

2.1. 실험 관리 정리

- Tensorboard 를 잘 활용하면 실험에 사용한 하이퍼파라미터의 대한 정보를 config.json 파일에 잘 정리할 수 있다는 방법을 알게 되서 따로 수기로 적지 않아도 어떤 실험을 했는지 알 수 있어서 좋았다.
- 그렇지만, 그렇더라도 다른 변수들을 고정하고 한 가지 하이퍼파라미터만 변경해보면서 실험을 해야 하는데 그 부분은 학습시간이 오래 걸린다는 생각때문인지 중심을 잡고 하지 못한점에 대해서 아쉽다.
- 다음 번에는 확실하게 실험할 수 있도록 Tensorboard 도 사용하면서 수기 관리도 해야 할 것 같다.

2.2. 데이터의 중요성!

- 다른 어떤 방법들 보다 데이터를 얼마나 정확하게 전처리 하느냐가 중요했던 대회였습니다.
- 데이터를 그냥 랜덤으로 train / valid set 으로 나누는 것 보다 사람을 기준으로 나눴을 때, train data 에서 봤던 사람을 valid data 에서 보지 않을 때의 성능이 LB 와 비슷할 수 있음을 알게 되자 한 단계 성장하고 우리의 valid score 가 LB score 와 비슷하다는 생각이 들게 되었고 모델의 성능을 비교할 최적의 키가 되었던 것 같습니다.
- 그리고 얼굴 부분에서 label 에 필요한 정보가 많이 있으므로 얼굴 부분만 Crop 한 데이터를 사용한 것도 성능 향상에 큰 도움이 되었던 것 같습니다.

2.3. 공유 문화의 중요성!

- 저는 실험하고 테스트하기 바쁘는데 이것저것 공유까지 해주시는 분들 덕분에 좋은 성적을 낼 수 있었습니다.
- 토론 게시판의 글들이 상당히 많은 도움이 되었고 저 또한 다음 P-Stage 에서는 공유할 수 있는 사람이 되고 싶습니다.

2.4. 팀의 중요성!

- 혼자서는 엄청 오래걸렸을 시도들을 팀원들이 나눠서 진행하다 보니 짧은 시간안에 고를 수 있었습니다.
- 혼자서 잘하는 것도 중요하지만 협업의 중요성을 다시 느낍니다.

3. 마주한 한계와 도전 속제

3.1. 아쉬운 점

- f1_score의 점수가 1등과 진짜 0.007 밖에 차이 안나는데 이 부분을 올릴 수 있는 방법을 찾지 못한것에 대해 너무 아쉽습니다.
- 18개의 label 이 하나로 뭉쳐져 있지만 서로 다른 label 3개가 적절히 조합된거라서 age, gender, mask 의 3개의 label 을 잘 예측하는 모델을 만들어서 하는 방법을 먼저 시도할 수 있었더라면 더 좋은 결과가 있을 수도 있었을 거라고 생각하는데 너무 뒤늦게 생각하고 적용해 본 것이 아쉽습니다.
- 코딩 실력이 부족한 탓인지 함수를 간단하게 모듈화 하는 것이 쉽지 않았습니다.
 - 일단 구현을 우선순위로 생각하다보니 코드가 길어지고 재사용이 쉽지 않았습니다.

3.2. 한계/교훈을 바탕으로 다음 스테이지에서 새롭게 시도해볼 것

- 나만의 실험 관리하는 방법을 만들기
 - 엑셀을 활용하든지 표를 활용해서 적절한 테스트를 진행해 볼 수 있는 form 을 만들어서 학습 테스트 진행
- 토론 게시판에 공유할 내용 하나 써보기
- 코드 refactoring 을 해서 모듈화를 끝내고 submission 제출하기
- 이 문제를 해결하기 위한 다양한 방법을 더 찾아보고 검색하기