이미지 분류 competiton개인 회고

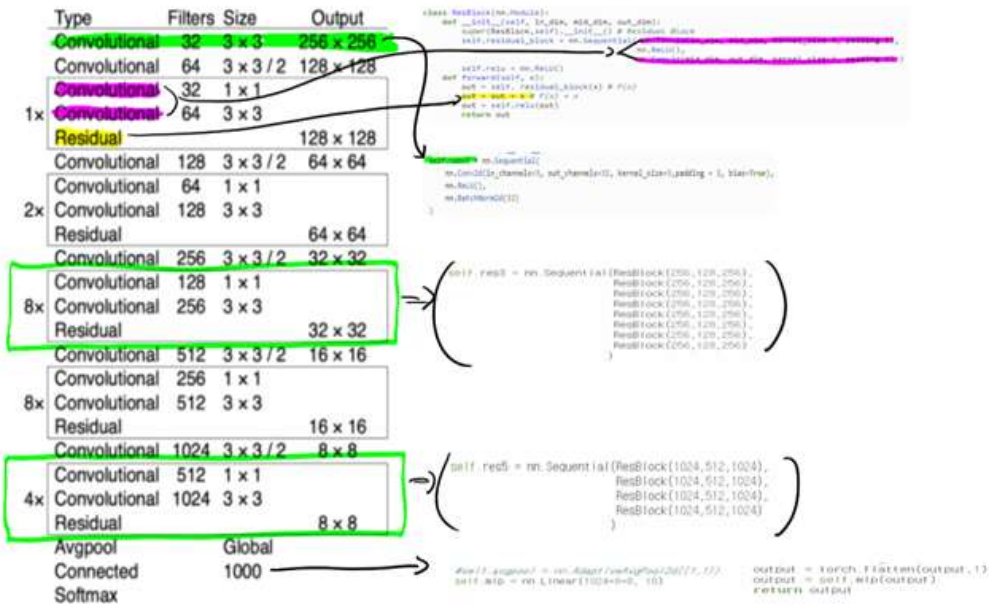
홍현승 T_2250

대회 소개 : 이번 competition은 사람 얼굴 이미지를 통해, 마스크의 착용 여부(착용, 미착용, 불완전 착용), 성별, 나이(30 이하, 30이상 60이하, 60이상)를 구분하는 대회였습니다. 총 18개의 class를 분류하는 대회였습니다.

진행: 대회 1주차: 개인 모델 형성, 개인 모델 성능 평가 시도

대회 2주차: pretrained모델 테스트 및, 앙상블 시도/ hyperparameter tuning시도

대회 1주차: custom model 개발
https://github.com/Hong-Hyun-Seung/upstage-basic-deeplearning/blob/main/Competition1-custom.ipynb



성능:(11epoch기준) Train-> Loss: 0.5562 Acc: 80.6481%

Valid -> Loss: 0.7922 Acc: 74.2328%

대회 2주차:

단계1)여러 pretrained모델 freeze/unfreeze 성능 비교

reference: "Bag of Tricks for Image Classification with Convolutional Neural Networks"
https://github.com/Hong-Hyun-Seung/upstage-basic-deeplearning/blob/main/Competition1_Selecting%20Pretrained.ipynb

모델 설정(resnet34), lossfunction 설정, optimizer 설정

```
model = getModel('resnet34')
loss_func = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr=0.001, momentum = 0.9)
```

Selecting Pretrained Model(resnet34 성능 check)

```
train_and_valid_check(model,'resnet34')
```

```
[Train #9] Loss: 0.0088 Acc: 99.8280%
                                    d
[Validation #9] Loss: 0.0911 Acc: 96.5608%
```

모델 설정(resnet152), lossfunction 설정, optimizer 설정

```
model = getModel('resnet152')
loss_func = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr=0.001, momentum=0.9)
```

Selecting Pretrained Model(resnet152 성능 check)

```
train_and_valid_check(model, 'resnet152')
```

```
[Train #9] Loss: 0.0086 Acc: 99.7950%
[Validation #9] Loss: 0.0782 Acc: 97.4074%
```

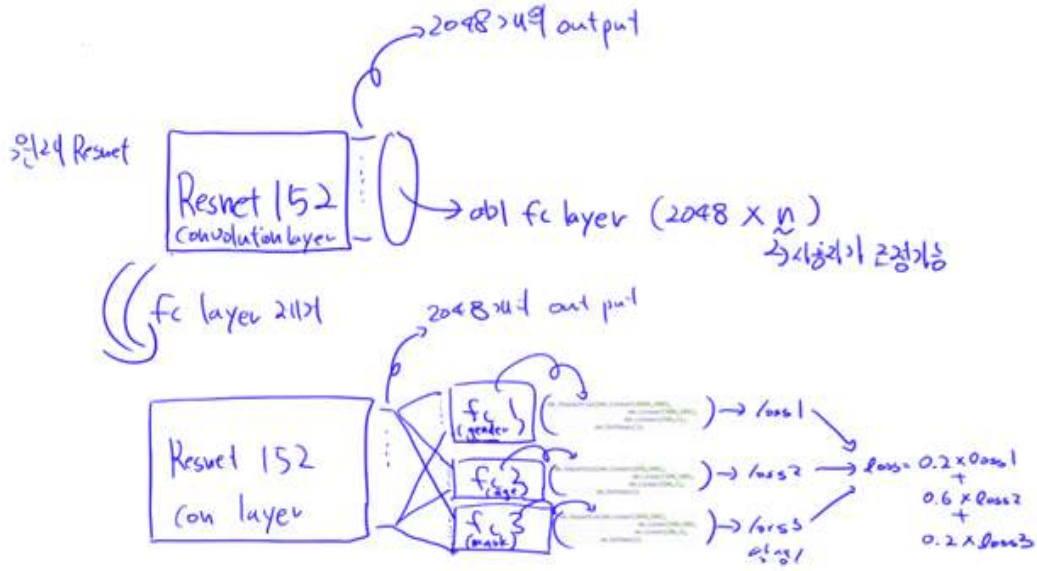
처음에는 training시키는데 걸리는 시간이 너무 길어서 pre trained model을 freeze시켜둔 후, fully connected layer만 training 시키는 방법으로 방향을 잡았으나, 낮은 성능을 보임-> 이는 기존 pretrained domain과 competition이 요구하는 domain의 차이가 많아서 그런 것으로 보임 ->"fine tuning"필요!!!

단계2) resnet 152 여러가지 방식 적용하여 변환 및 성능 평가

방법1: Backbonelayer(resnet152)에 3개의 FC 연결(FC1:성별용,FC2:mask용,FC3: age용)

(https://github.com/JIHOO97/BoostUP/blob/Image-Classification/MultiLayer.ipynb)

판단근거: 각각의 loss를 고려 하여 신경망을 구성하는 것이 더 좋을 것이라는 판단, age에서 가장 많은 오류들이 발생한다는 판단 하에, loss가중치를 age에 더 높게 주는 것이 효율적일 것이라는 판단



성능: (epoch 10 번 기준) Train-> Loss: 0.5613 Acc: 85.0727%

Valid-> Loss: 0.5146 Acc: 85.3439%

방법2: Resnet152에 여러 가지 방식 적용(ensemble, parameter tuning, regularization, etc.)

1.Ensemble: SWA(reference: **"Averaging Weight Leads to Wider Optima and Better Generalization"**)

```
base_opt = optim.SGD(model.parameters(), lr=lr, momentum=config.momentum)
optimizer = SWA(base_opt, swa_start=3, swa_freq=1, swa_lr=0.0005)
```

2.Regularization:L2regularization-> validation accuracy가 계속 상승하는 경향을 보여, overfitting이 발생한다 판단, regularization적용!

```
#model = getModel('resnet152')
loss_func = nn.CrossEntropyLoss()
optimizer = torch.optim.SGD(model.parameters(), lr=0.001, momentum=0.9, weight decay=0.05)
```

3.Hyperparameter Tuning: horizontal flip, p=0.6사용! -> validataion accuracy가 계속 상승하는 경향을 보여, overfitting이 발생한다고 판단!!! Data augmentation사용!

성능: (epoch10번 기준) Train->Loss: 0.1433 Acc: 97.5198%

Valid -> Loss: 0.2162 Acc: 95.2646%

테스트 결론:

결론적으로, 이번 대회에서 필자가 사용한 모델 중 가장 효율적이었던 모델은, resnet152(unfreezed)에 regularization을 사용한 모델이었습니다. 본인의 팀이 리더보드상에서 2주동안 계속 1등을 하였는데, 최종 결과에서 9등이 되어 다소 아쉬웠습니다. 처음에 팀원 지후님께서 18000장의 데이터중 outlier와 noise를 일일이 다 제거해주셨는데, garbage in garbage out과 반대되는, treasure in treasure out(?) 효과를 얻게 되어, 다시한번 EDA와 데이터 전처리의 중요성을 깨달았습니다. 또한 최종적으로 valid에서 95퍼가 넘어가는 모델이 리더보드상에서 90퍼를 못넘는 이유를 알았는데, 이는 나이 예측에 문제 가 있어서입니다. 이 때문에 처음 필자가 사용한 multi task learning 기법에 age에 작은 가중치를 두어 label smoothing을 해봤으면 어떤 결과가 나왔을까 궁금하기도 하면서 아쉽기도 했습니다. 또한 test data에 대한 EDA를 진행했다면, 이 또한 방지 할 수 있지 않았을까 라는 생각이 들어 다시금 EDA의 중요성을 느꼈습니다.