

## <프로젝트의 방향성>

Medium을 통해 towardsdatascience채널을 주로 보는 중에 fastai에 대한 글들을 종종 보았습니다.

AI공부를 tensorflow를 통해 시작했었고, 부스트캠프를 진행하면서 pytorch, pytorch lightning을 써 보았습니다.

부스트캠프를 진행중 오픈소스들을 참고하게 되었을 때, tensorflow로만 구현이 되어지거나 pytorch에 비해 더 많은 자료들이 있는 경우들도 보게 되었습니다.

이전에 tensorflow를 조금은 경험해봤기에 위의 상황에 대해 많은 도움이 되었습니다.

뒤의 2,3,4 stage에서보다 이번 stage 1에서 접해보는 것이 조금은 더 쉬울 것 같은 생각으로 이번 프로젝트의 방향성은 fastai를 사용해보자는 목적으로 진행되었습니다.

## <기술적인 도전>

### ● 검증(Validation) 전략

1. Train data들은 rand augmentation이 적용되어 loss가 높게 나오므로 validation으로 변형 전의 data들을 이용해 loss를 확인하는데에 사용.

### ● 사용한 모델 아키텍처 및 하이퍼 파라미터

1. 아키텍처 및 전략 : resnet34, multi task learning
  - a. LB 점수 : 0.68(f1) 151등
  - b. random augmentation : 개수(3), 강도(6)
  - c. img\_size = 224 \* 224
    - i. center crop(380, 256)
    - ii. Resize(224,224)
  - d. 기타
    - i. 첫날 가볍게 진행했던 모델이라 코드 구성이 잘 기억이 나질 않지만, 이때 나왔던 결과가 최고 성능이 되었다.
    - ii. 두번으로 학습을 나누어 진행했다. 처음에는 상위 6개 레이어와, fc 레이어만 열고 학습한 후, 두번째 학습에서 모두 열고 상대적으로 작은 lr로 학습을 진행했다.
2. 아키텍처 및 전략 : efficientnet b2, each single task learning

- a. LB 점수 : 0.52(f1)
- b. random augmentation : 개수(3), 강도(6)
- c. img\_size = 260 \* 260
  - i. center crop(380, 256)
  - ii. Resize(260,260)
- d. optimizer : ranger (RAdam + Lookahead)
- e. 기타
  - i. 아래의 앙상블 방법을 추가해서 진행했다. 같은 환경에서 앙상블 추가전 보다 accuracy는 10%, f1score는 0.5가 증가했다.

#### ● 앙상블 방법

1. Train data가 불균형이 너무 심해서 데이터를 여러 set로 분리를 했습니다.
  - i. Age : 60이상의 데이터 수에 맞추어 나머지 2개의 데이터들을 3분할을 해서 9개의 set을 만들어 hard voting 진행
  - ii. Mask : 1:5:1의 비율을 가진 데이터 셋이여서, 2:2:2가 될수있도록 비율이 1인 데이터들을 2배 over sampling을 진행하고, 나머지 5의 비율데이터는 조합(5C2)을 뽑아내어 10개의 set을 만들어 hard voting 진행
  - iii. Gender : male의 수에 맞추어 female을 2개로 분할하여 만든 2set과 원래 그대로의 데이터 1set, 총 3set의 dataset으로 학습하고 hard voting 진행

#### ● 시도했으나 잘 되지 않았던 것들

1. Fastai가 익숙하지 않았기에 모델의 변형을 크게 다루기에는 무리가 있었다.
2. 처음 fastai v1을 사용해서 진행하다가 Timm 모델을 fastai에 적용하려면 fastai v2를 사용해야 했으며 version이 업그레이드되면서 바뀐 부분이 너무많아 이해하는데에 시간 소요가 매우 컸다.
3. Attention을 주는 모델을 사용하면 좋을 것 같다는 생각으로 ECA(Efficient Channel Attention)이 사용된 eca-resnet을 사용하여 며칠동안 실험해 보았지만 성능이 좋지 못했다.
4. Multi task learning에서 hard sharing외에 다른 방법들을 사용해보려고 찾아보았지만, 아직 이해하기에는 실력이 부족하여 적용하는데 실패했다.
5. Multi task learning을 아래의 링크와 같이 age는 직접 예측을 시도해 보았다. sigmoid를

통해 0~1로 학습을 진행했고 최대가 60이되어서 60은 1이 되도록 시도해 보았다. loss 계산시에 age는 mse를 통해 진행하고 data imbalance를 통제하기위해 학습 진행중 loss에 적은 데이터들은 많은 가중치를 받도록 조정해 주었지만, 60대가 넘는 분들을 40대 중후반으로 예측하는 현상이 발생했다. (참조 링크 : <https://towardsdatascience.com/multi-task-learning-with-pytorch-and-fastai-6d10dc7ce855>)

## <학습과정에서의 교훈>

1. 조금더 겸손해져야겠다는 생각이 제일 먼저 들었습니다. 주어진 baseline에 맞추어 진행했다면 더 좋았을 것 같습니다.
2. fastai에서 lr\_find는 learning rate을 찾아주는 ( Leslie Smith in Cyclical Learning Rates for Training Neural Networks : <https://arxiv.org/abs/1803.09820>) 기술로서, 추후에 다른 ML을 진행할 때에 큰 도움이 될 수 있을 것 같습니다.
3. Opt\_func로 ranger를 사용했는데, 다시한번 되짚어주는 모습이 학습중에 보였고, 이또한 추후에 사용하기에 좋을 것 같습니다.
4. 앞으로도 정말 많은 공부가 필요하다고 느꼈다.