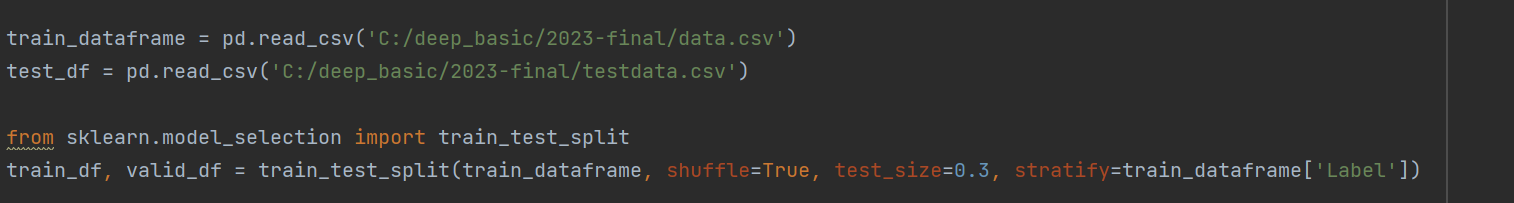
딥러닝 기초 기말 보고서

1. 모델 구조
2. Dropout / batchnorm / weight initialize / transform 종류 별로 정리 / auto augment /optimizier(Adam AdamW)

* Vgg16 , conv층에 따른 실험 , 특성 추가 과정 및 실험 결과

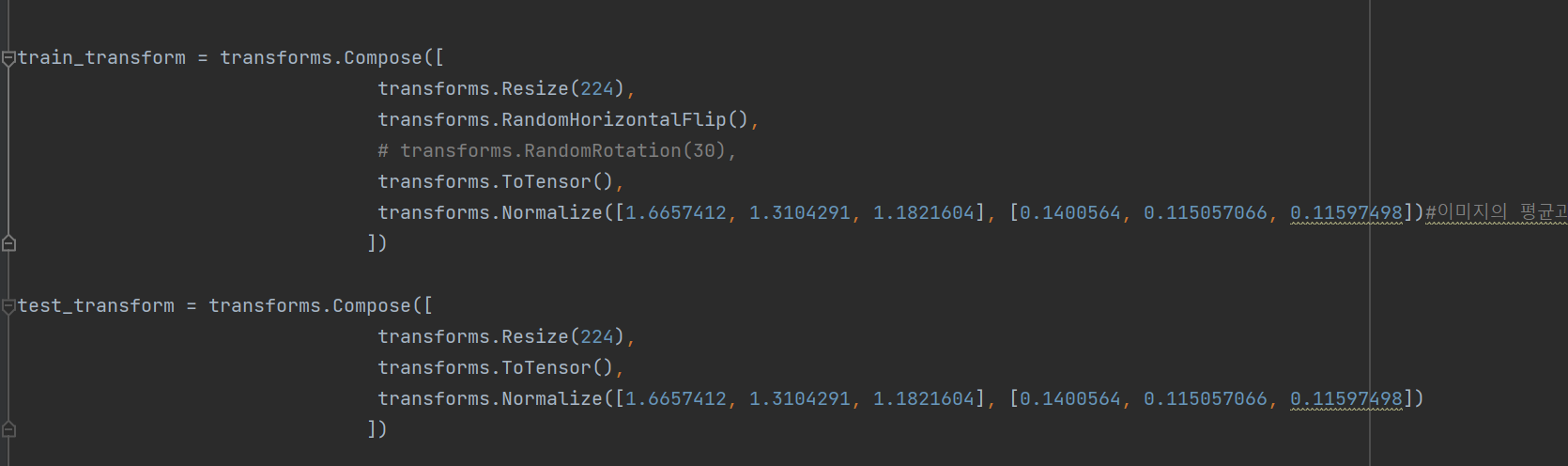
1. Learning rate
2. Weight\_decay - regularization
3. Label smoothing
4. 스케줄러 – 종류 별로 정리
5. K-fold
6. Ensemble – soft hard 정리
7. Grid\_search

먼저 저는 VGG16 모델로 성능이 얼마나 나오는지 확인을 해보기 위해 학습을 시켜보았으나 과적합이 계속 나서 모델의 복잡성을 줄이기 위해 제가 모델을 만들었습니다.



train\_dataframe으로 데이터를 불러오고 train과 validation 데이터를 7:3 비율로 나누어 주었습니다.

무작위로 섞는 shuffle을 주어 모델의 일반화 성능을 올려주었습니다. Stratify



먼저 transform 부분에서 데이터 증강과 데이터 정규화를 하였습니다.

1. Normalization

데이터의

1. 데이터 증강

사람 얼굴이기 때문에 수직 변환(Vertical Flip), 색깔 변환을 주면 안될 것이라고 생각을 하였습니다. 사람의 얼굴은 좌우가 같기 때문에 RandomHoriopf과 resize를 해주었습니다.

모델의 구조는 **Conv** 5층과 **Fully connected** 3층으로 해보았습니다. **Optimizer**는 Adam으로 주었습니다. 가장 기본으로 하였을 때 성능이 42%였습니다. 50epoch중에 30epoch부터 validation loss가 떨어지지 않았고 train loss만 떨어졌습니다.

과적합을 해결하기 위해서 **dropout**과 **batch normalization**을 추가하여 모델을 구성하였고 Adam에 **regularization**을 주기 위하여 weight decay를 추가하였습니다.

1. dropout
2. batch normalization
3. regularization

과적합을 막는 요소들을 적용을 했음에도 성능이 나아지지 않아 저는 learning rate와 batch\_size등 하이퍼파라미터를 바꿔보았습니다.

1)

2)

3)

4)

이 과정에서 처음부터 weight가 초기화가 나쁘게 되었이다고 생각하여 **weight intilialz**를 주었습니다.

1. weight 초기화

그 결과 57프로로 성능이 향상되었습니다. 여기서 성능을 올리기 위하여 위의 하이퍼파라미터 과정을 지속하였지만 오르지 않았습니다. 그래서 스케줄러를 적용하여 보았습니다.

1. 스케줄러
2. cosine
3. step
4. reduce
5. Ex

Optimizer를 바꿔 학습을 시켜보았습니다.

1. AdamW

하지만 스케줄러를 적용하였지만 효과를 보지 못하여 모델 구조를 수정해보았습니다. 과적합이 모델의 구조가 너무 복잡해서 난다고 생각하여 모델을 단순화시켰습니다.

모델의 구조를 설명

모델을 단순화시켜도 효과가 없었습니다. 그래서 저는 과적합에 데이터 증강이 효과적이라고 생각하여 auto augmenation을 적용하여 보았습니다.

1. Auto augmentation

모델을 단순화하고 데이터 증강을 시켜도 과적합에는 아무 효과가 없었습니다.실제로 더 발산하는 경향도 보였습니다.그래서 저는 지금까지 저장한 loss graph를 보고 overfiitng이 아닌 것을 깨달았습니다. Validation loss가 어느 순간부터 올라가지도 내려가지도 않고 그냥 학습이 되지 않고 있었습니다. 그래서 저는 Conv 층에 있는 모든 dropout을 빼주었습니다. Weight\_decay도 낮추어 학습을 해주었습니다. 그 결과 62프로 성능이 뛰었습니다.

Conv층에 있는 모든 dropout을 빼고 Fully 층에만 주는 5층과 6층 모델에 모두 효과가 있었습니다.

그래서 저는 성능이 좋은 모델 3개를 만들고 **ensemble**을 시도하였습니다.

1. ensemble
2. soft voting
3. hard voting

저는 62프로, 58프로,59프로 모델을 ensemble 결과 64 프로 성능이 2프로 향상되었습니다. 저는 성능을 더 올리기 위하여 60프로 모델을 더 만들기 위해 시도를 하였습니다.

**Confusion matrix**는 출력해보면 4번이 항상 낮은 acc를 보였습니다. 그래서 저는 label smoothing을 적용하였습니다.

1. confusion matrix
2. label smoothing

그 결과 4번의 acc가 높아졌습니다.

언더피팅이 30epoch이후로 계속 보여 모델의 구조를 복잡하게 쌓아서 실험을 해보았습니다.

그 결과 ~~~~~~~~~~~~

1. 특성 추가
2. 남자 여자 구분 학습
3. 느낀점