




2024-1학기 창의학기제 주간학습보고서 (7주차)

창의과제	세종대학교 집현캠퍼스를 개선시킨 웹서비스 개발				
이름	이지민	학습기간	5월 13일 ~ 5월 17일		
학번	23012127	학습주차	7	학습시간	3
학과(전공)	인공지능	과목명	자기주도 창의전공1	수강학점	3
※ 수강학점에 따른 회차별 학습시간 및 10주차 이상 학습 준수					
금주 학습목표	confusion matrix를 활용한 모델 평가를 통해 얼굴인식에 합당한 모델 학습 및 다른 pre-train 모델 실험				
학습내용	<p>이전 주차에서 confusion matrix를 통해 학습시킨 모델에 FN값이 높다는 것을 알게 되었다. 이후 여러번 모델을 다시 학습시켜 확인해보니 1종 오류가 0일 때 괜찮은 민감도 값을 가질 경우도 있었지만 그러지 않을 경우도 있었다. 먼저 이를 pre-train 모델이 접합하지 않음으로 인한 것이라 판단하여 FaceNet으로 바꾸어 fine-tuning을 진행해 보았다.</p> <p>먼저 FaceNet에 대해 설명하면, 딥러닝을 이용한 얼굴인식모델이다. 이 모델의 특징으로는 이전 보고서에서 언급했던 triplet loss를 활용하여 학습하였다.</p>  <p>그림 1 FaceNet 구조</p> <p>triplet loss를 사용했기 때문에 이 모델은 얼굴을 분류하는 것이 아닌 얼굴은 서로 구분할 수 있도록 한다. 이 점이 이전의 얼굴인식 딥러닝 모델과의 차이점이다.</p> <p>먼저 pre-train한 모델이 잘 분류가 되는지 확인해 보았다. facenet_pytorch에서 제공하는 함수를 가지고 간단하게 거리차를 확인해볼 수 있지만 좀 더 다양하고 빠르게 확인해보고 싶어서 이전에 사용하던 코드를 수정하여 값을 확인해 보았다. 먼저 이전 코드의 mobilenet 대신에 pre-train한 FaceNet모델을 넣은 후에 FC를 삭제하였다. train_model 함수에서 train dataset이 들어올때도 model.eval()로 변경하여 모델이 학습되지 않고 FaceNet의 성능을 확인해보았다.</p> <p><수정한 코드></p> <pre>from torchvision.models import mobilenet_v3_large from facenet_pytorch import InceptionResnetV1 class SiameseNetwork(nn.Module): def __init__(self): super(SiameseNetwork, self).__init__() # self.model=mobilenet_v3_large(pretrained=True)</pre>				



```

self.model=InceptionResnetV1(pretrained='vggface2') #FaceNet으로 변경
...

self.fc = nn.Sequential(
    nn.Linear(960, 512),
    nn.ReLU(inplace=True),
    nn.Linear(512, 128),
    #nn.ReLU(inplace=True),
    #nn.Linear(256, 128),
)
self.model.classifier=self.fc
'''      # 학습하지 않도록 수정
def forward(self, input1, input2):
    output1 = self.model(input1)
    output2 = self.model(input2)
    return output1, output2

...
def train_model(model, dataloader_dict, criterion, optimizer, num_epoch):
    ...
    for phase in ['train', 'val']:
        if phase == 'train':
            model.train()
        else:
            model.eval()

    for inputs1,inputs2, labels in tqdm(dataloader_dict[phase]):
    ...

```

민감도:1.00 특이도:1.00 PPV:1.00 NPV:1.00
 pos_m:0.40 neg_m:1.30
 accuracy:1.00
 1종오류:0

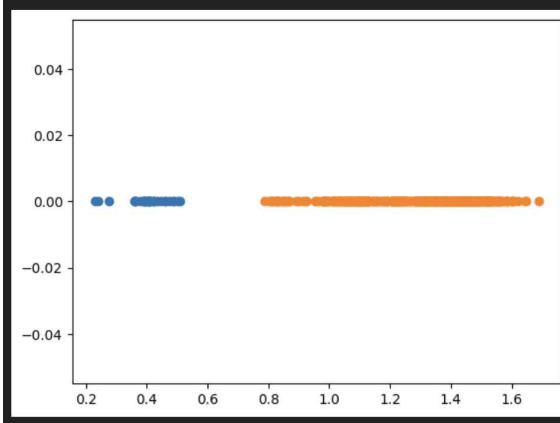


그림 2 pre-train된 FaceNet의 성능결과 학습시켜 보기로 하였다.

현재 모델의 문제를 해결하기 위해 코드를 다시 확인하던 중 contrastiveloss클래스에 오류가 있음을 확인하였다. 코드를 확인해본 결과 label과 1-label이 반대로 적혀있었다.

여러 epoch를 돌려 확인해본 결과 backpropagation의 영향이 있는지 train loss값에는 약간의 변화가 있었지만 test loss값은 일정하여 FaceNet의 성능만 확인할 수 있음을 알 수 있었다. 결과를 확인하면 이미 FaceNet은 사용자의 얼굴을 잘 인식하고 있음을 알 수 있다. 그래프를 보면 y축은 큰 의미를 가지고 있지 않고 x축은 anchor데이터와의 거리를 나타낸다. 그리고 파란색 점은 positive dataset, 주황색 점은 negative dataset이다. 해당 코드를 돌려본 결과 현재 서술한 내용 말고도 다양한 깨달음을 얻을 수 있었다. 하지만 우리의 현재 목표는 사용자의 얼굴만을 인식하는 모델이므로 이어서 modilenet으로



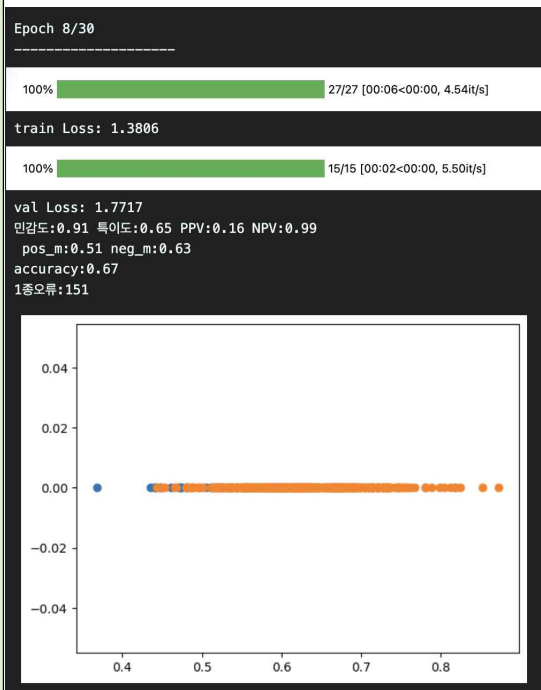
3주차에 서술했던 내용에 더해 설명하면, 현재 코드는 label이 1일 때는 거리가 2만큼 벌어지도록 label이 0일때는 거리가 가까워지도록 학습되고 있었다. 그 결과 학습할수록 positive dataset의 거리가 멀어져 2종오류, FN값이 커졌던 것이다. 반면에 1종 오류가 낮아진 것은 negative data는 한 종류의 데이터가 아니라 다양한 데이터이기 때문인 것 같다.

```
def forward(self, output1, output2, label):
    euclidean_distance = F.pairwise_distance(output1, output2, keepdim = True)
    loss_contrastive = torch.mean((1-label) * torch.pow(euclidean_distance, 2) +
                                   (label) * torch.pow(torch.clamp(self.margin - euclidean_distance, min=0.0), 2))
```

그림 3 잘못된 contrastiveloss클래스

```
def forward(self, output1, output2, label):
    euclidean_distance = F.pairwise_distance(output1, output2, keepdim = True)
    loss_contrastive = torch.mean((label) * torch.pow(euclidean_distance, 2) +
                                   (1-label) * torch.pow(torch.clamp(self.margin - euclidean_distance, min=0.0), 2))
```

그림 4 수정한 contrastiveloss 클래스



하지만 수정한 후 다시 학습시켜본 결과는 좋지 않았다. 그림 5를 확인해보면 정확도 또한 좋지 않고 1종 오류가 쉽게 고쳐지지 않고 positive dataset의 거리가 먼 것을 확인할 수 있다. epoch를 30까지 돌려본 결과도 비슷하였다. 여기서 할 수 있는 추론은 traindata에 오버피팅 됐을 수도 있다는 가정이다.

그림 5 contrastiveloss를 수정한 후 epoch 수행 결과

학습방법	pytorch_facenet github를 통해 이를 활용해 pre-train함수를 다시 구현. 여러 epoch를 돌려가면서 현재 모델의 문제점을 파악
학습성과 및 목표달성도	이번 주차에는 내 학습과정에 다양한 오류를 가지고 진행되었다는 것을 알게 되었다. 첫째 모델을 배포할때는 이미 학습된 모델이어야 한다. 둘째 인공지능은 다양한 또는 새로운 데이터에 유효한 머신이다. 셋째 얼굴인식에 대한 과거 학술에 대해 깊게 알아보지 않고 시작하였다. 하지만 이미 중반을 넘었으므로 처음 정한대로 학습을 계속하고자 한다. 이번 주차에 loss 함수에 중대한 오류가 있음을 발견하고 이를 수정하였다. 그러므로 이론상 모델이 더 정확해졌는데 결과는 그렇지 못했다.
참고자료 및 문헌	https://en.wikipedia.org/wiki/FaceNet#:~:text=FaceNet%20is%20a%20facial%20recognition,Pattern%20Recognition%20held%20in%202015.



	https://github.com/timesler/facenet-pytorch?tab=readme-ov-file#guide-to-mtcnn-in-facenet-pytorch
내주 계획	여러 epoch을 돌려보면서 현재 모델의 문제점에 대해 알아본다.

년 월 일

지도교수

(인)