기초인공지능 HW4 보고서

20181650 안도현

1. 3번 문제를 해결하기 위해 시도한 방법

nn.Module을 상속받아 CustomMLP 클래스를 만드는 문제였는데, 상속받은 클래스의 생성자 호출과 activation 함수 설정 및 다른 forward 메소드 같은 것들은 이미 다 작성되어 있어 직접 짜야 하는 부분은 layer list 작성뿐이었다.

즉, 진행해야 하는 것은 인자인 out_feat_list에서 각 중간 layer의 output feature 개수를 가져와서 nn.Linear를 호출해 네트워크를 형성한 뒤, activation 함수를 호출하는 것 뿐이다. 따라서 먼저 layers에 nn.Linear(1024,out_feat_list[0]) 및 activation을 layers에 append하면 input layer가 정의되고, 이후 range(1,num_layers)에 대해서 for loop를 시행하여 out_feat_list의 이웃한 값을 인자로 nn.Linear를 호출하고 activation을 호출하는 것을 반복하면 된다.

2. 4번 문제를 해결하기 위해 본인이 시도한 방법. 가장 성능이 높았던 방법에 대해서 소개. 최종적인 test 성능

첫 번째 시도) layer을 줄이고 learning rate를 늘리기

```
##################################### TODO #####################################
SEED = 100 # can be changed
LEARNING_RATE = 0.003 # can be changed
torch.random.manual_seed(SEED)
model = CustomMLP(num_layers=3, out_feat_list=[1024,128,10], act='relu').to(device) #
optimizer = torch.optim.Adam(model.parameters(), lr=LEARNING_RATE) # can be changed
################################################################################
```

이미지라는 train set이 layer를 많이 둬야 하는가 싶어 layer를 줄인 뒤 learning rate를 늘려보았다. 그러자 결과는 아래와 같이 쓸 수 없는 모델이 되었다.

```
TEST average accuracy: 10.00%
```

두 번째 시도) learning rate는 고정한 채 layer 개수를 늘려보기

```
SEED = 100 # can be changed
LEARNING_RATE = 0.001 # can be changed
torch.random.manual_seed(SEED)
model = CustomMLP(num_layers=8, out_feat_list=[1024,512,256,128,64,32,16,10], act='relu').to(device)
optimizer = torch.optim.Adam(model.parameters(), lr=LEARNING_RATE) # can be changed
```

layer 개수를 줄였을 때 정확도가 올라가지 않았으므로 반대로 hidden layer의 중간중간에 적절한 값을 채워넣고 학습하게 하면 어떨까 싶어 그대로 시행해보았다.

```
TEST average accuracy: 41.37%
```

그 결과 목표로 했던 정확도에 도달했다.

세 번째 시도) 기존의 layer에 대하여 learning rate와 random seed값만 바꿔보기

```
SEED = 200 # can be changed
LEARNING_RATE = 0.002 # can be changed
torch.random.manual_seed(SEED)
model = CustomMLP(num_layers=5, out_feat_list=[1024,256,64,16,10], act='relu').to(device)
optimizer = torch.optim.Adam(model.parameters(), lr=LEARNING_RATE) # can be changed
```

random seed는 무슨 의미인지 잘 와닿지가 않지만 변경 가능한 값이라 우선 두 배를 해보았고, learning rate만 두 배로 하여 기존의 네트워크를 그대로 사용한다면 각 epoch 간에 정확도 상승이 눈에 띄지 않을까 하여 실제로 바꿔보았다.

```
TEST average accuracy: 42.48%
```

그 결과 정확도가 조금 더 올라갔다. 그래서 이 모델을 최종 제출하기로 결정했다.