기초 인공지능 : Assignment04(Neural Network)

전공: 컴퓨터공학 학년: 3학년 학번: 20201635 이름: 전찬

**0. 목차**

1. 구현 목표

2. 3번 문제를 해결하기 위해 시도한 방법

3. 4번 문제 해결을 위해 시도한 방법 및 가장 성능이 높았던 방법 소개, 최종 test 성능

**1. 구현 목표**

이번 과제에서는, python과 torch module을 활용해 직접 인공신경망을 구현해 본다. 기초적인 MLP(Multi Layer Perceptron) 부터 image classification을 수행하는 DNN(Neural Network)을 구현해 성능 평가를 수행한다.

**2. 3번 문제를 해결하기 위해 시도한 방법**

3번 문제는 nn.Module을 상속받는 class의 init 부분을 구현하는 것이다. 특히 input에 따라 적절한 input layer, hidden layer, output layer을 구성해야 하며, activation function도 설정해 주어야 한다.

개인적으로 반복문을 바탕으로 out\_feat\_list의 값에 따라 nn.Linear()의 두 parameter을 설정해 주었고, 이전 값의 output number을 input number 변수에 저장하여 활용해 주었다. 또한 이미 구현되어 있는 self.select\_act() 함수를 활용해 activation function을 각 반복문마다 추가해 주었다. 이를 통해 적절한 CustomMLP의 init 을 구현할 수 있다.

**2. 4번 문제 해결을 위해 시도한 방법 및 가장 성능이 높았던 방법 소개, 최종 test 성능**

4번 문제는 주어진 기본 neural network에서 여러 parameter을 변경해가며 가장 좋은 성능을 찾아내야 한다. 개인적으로 5번의 변경을 시도해 최적의 값을 구했는데 이는 아래와 같다.

1. learning rate 변경 (0.001 -> 0.0005) / 테스트 정확도 42.56%

2. relu ->tanh로의 변경 / 테스트 정확도 38.98%

3. optimizer 변경 (Adam -> Adadelta) / 테스트 정확도 9.93%

4. seed 변경 (100 -> 200) / 테스트 정확도 41.03%

이를 통해 38 ~ 42% 의 정확도를 관찰할 수 있었고, 특이점이 존재한다면 3. optimizer 변경에서 9.93%이다. 이는 개인적으로 local optimal 등에서 벗어나지 못해 저런 결과가 나온 것이라고 생각할 수 있었다.

따라서 최적의 정확도는 learning rate 가 0.0005이며 나머지 parameter은 바꾸지 않았을 때로 42.56%의 정확도의 NN을 구현할 수 있었다.