

# 인공지능 챕터 5 과제

202055518 김병현

## 꽃 이미지 분류 신경망

```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [10])
fm.exec_all(epoch_count=10, report=2)
Model flowers_model_1 test report: accuracy = 0.242, (0 secs)
추정확률분포 [19,22,19,19,21] => 추정 dandelion : 정답 dandelion => 0
추정확률분포 [19,22,19,19,21] => 추정 dandelion : 정답 rose => X
추정확률분포 [19,22,19,19,21] => 추정 dandelion : 정답 rose => X
```

```
fm2 = MlpModel('flowers_model_2', fd, [30, 10])
fm2.exec_all(epoch_count=10, report=2)
Model flowers_model_2 test report: accuracy = 0.330, (0 secs)
추정확률분포 [12, 9,37, 9,33] => 추정 rose : 정답 rose => 0
추정확률분포 [36,35,14, 4,11] => 추정 daisy : 정답 dandelion => X
추정확률분포 [18,22,19,20,22] => 추정 tulip : 정답 sunflower => X
```

fm2 가 fm 보다 정확도가 높게 나오는데, 이는 모델의 구조 차이 때문이다.  
fm2 에서 은닉층이 추가된 점과 노드 수가 증가한 점이 성능 향상에 기여한 것으로 보인다.

## 은닉층의 추가로 인한 표현력 증가

- fm은 [10]이라는 단일 레이어만을 가지고 있는 반면, fm2는 [30, 10]으로 두 개의 은닉층을 포함한다. 이때문에 fm2는 더 복잡하고 다양한 특징을 학습할 수 있는 표현력을 가지게 된다.
- 은닉층이 하나 더 추가되면서, 데이터의 패턴과 구조를 더 잘 학습하게 된다.

## 노드 수 증가

- fm2의 첫 번째 은닉층에는 30개의 노드가 있어, fm보다 더 많은 뉴런을 사용하여 데이터를 처리할 수 있다.
- 일반적으로 모델에 더 많은 노드가 있을수록 더 복잡한 관계를 학습할 수 있어, 정확도 향상에 도움이 된다. 특히 이미지 데이터는 많은 특성을 포함하기 때문에 더 많은 뉴런을 통해 더 좋은 성능을 발휘할 수 있다.

```

fm2 = MlpModel('flowers_model_2', fd, [30, 10])
fm2.exec_all(epoch_count=30, report=2)
Model flowers_model_2 test report: accuracy = 0.411, (0 secs)
추정확률분포 [ 1,27, 0,62,10] => 추정 sunflower : 정답 sunflower => 0
추정확률분포 [12,28,11,24,25] => 추정 dandelion : 정답 tulip => X
추정확률분포 [18,82, 0, 0, 0] => 추정 dandelion : 정답 dandelion => 0

fm2 = MlpModel('flowers_model_2', fd, [30, 10])
fm2.exec_all(epoch_count=50, report=2)
Model flowers_model_2 test report: accuracy = 0.456, (0 secs)
추정확률분포 [ 4, 1,71, 7,18] => 추정 rose : 정답 rose => 0
추정확률분포 [50,50, 0, 0, 0] => 추정 daisy : 정답 dandelion => X
추정확률분포 [39,35,17, 0, 9] => 추정 daisy : 정답 tulip => X

fm2 = MlpModel('flowers_model_2', fd, [30, 10])
fm2.exec_all(epoch_count=60, report=2)
Model flowers_model_2 test report: accuracy = 0.425, (0 secs)
추정확률분포 [ 2, 2,11,14,72] => 추정 tulip : 정답 tulip => 0
추정확률분포 [ 0, 2, 0,98, 0] => 추정 sunflower : 정답 sunflower => 0
추정확률분포 [ 0, 0, 2, 5,93] => 추정 tulip : 정답 tulip => 0

```

지금 모델 구조에서 에포크를 늘려서 올리는 성능은 이 정도인 것 같다.

위 구조에서 은닉층을 추가해보았다.

```

fm2 = MlpModel('flowers_model_2', fd, [30, 20, 10])
fm2.exec_all(epoch_count=10, report=2)
Model flowers_model_2 test report: accuracy = 0.454, (0 secs)
추정확률분포 [37,48,10, 0, 5] => 추정 dandelion : 정답 rose => X
추정확률분포 [ 4,87, 0, 8, 1] => 추정 dandelion : 정답 dandelion => 0
추정확률분포 [24,59, 5, 4, 8] => 추정 dandelion : 정답 dandelion => 0

```

에포크가 낮더라도 정확도가 조금 더 증가한 것을 확인할 수 있다.

위 구조에서 노드를 증가해보았다.

```
fm2 = MlpModel('flowers_model_2', fd, [64, 32, 16])
fm2.exec_all(epoch_count=10, report=2)
Model flowers_model_2 test report: accuracy = 0.465, (0 secs)
추정확률분포 [17,11,31, 6,36] => 추정 tulip : 정답 dandelion => X
추정확률분포 [ 2, 6, 1,87, 5] => 추정 sunflower : 정답 sunflower => 0
추정확률분포 [23,36,17,10,14] => 추정 dandelion : 정답 daisy => X
```

노드 수가 많아지면 모델이 데이터를 표현하는 능력이 높아져 더 나은 성능을 얻을 수 있지만, 과적합 가능성도 증가할 수 있다.

위 모델 구조에서 에포크를 늘리며 실험을 진행해 보았다.

```
fm2 = MlpModel('flowers_model_2', fd, [64, 32, 16])
fm2.exec_all(epoch_count=30, report=2)
Model flowers_model_2 test report: accuracy = 0.474, (0 secs)
추정확률분포 [18, 1, 0,70,11] => 추정 sunflower : 정답 dandelion => X
추정확률분포 [93, 3, 2, 0, 2] => 추정 daisy : 정답 dandelion => X
추정확률분포 [ 4,93, 2, 0, 1] => 추정 dandelion : 정답 dandelion => 0
```

```
fm2 = MlpModel('flowers_model_2', fd, [64, 32, 16])
fm2.exec_all(epoch_count=40, report=2)
Model flowers_model_2 test report: accuracy = 0.429, (0 secs)
추정확률분포 [98, 0, 2, 0, 0] => 추정 daisy : 정답 daisy => 0
추정확률분포 [ 0,87, 0, 2,11] => 추정 dandelion : 정답 sunflower => X
추정확률분포 [ 0, 0,99, 0, 1] => 추정 rose : 정답 rose => 0
```

에포크를 늘리는 한계에 도달한 것으로 보인다.