

꽃 이미지 분류 신경망

부산대학교 정보컴퓨터공학부

15 학번 이연걸

1. 개요

다층 퍼셉트론을 이용하여 꽃 이미지 분류 신경망을 구성하고 여러가지 파라미터를 건드려보면서 정확도 향상을 바라본다. 다층 퍼셉트론을 사용했을 때 정확도가 높게 나오지 않을 것이다. 이에 따라 다층 퍼셉트론이 이미지 처리에 부적합한 이유를 분석한다. 또한, 이미지의 경우 이 때까지 했던 철판 분류, 전복 고리 수, 천체 펄서 등의 데이터보다 훨씬 많은 파라미터를 가지게 된다. 꽃 이미지 분류 신경망 구성을 통해 파라미터가 많아지면 어떤 문제가 생기는데 대해서도 생각해본다. 한편, 학습 시킬 때 데이터를 학습/검증/평가 셋으로 구분지어서 하게 되는데 이에 대한 이유도 분석해본다.

또한, 활성화 함수로 relu, tanh 함수를 번갈아 사용하며 이 둘의 결과 차이도 알아본다.

2. 구현 과정

2.1 은닉 계층 수에 변화를 주었을 때

```
fd = FlowersDataset()  
fm = MlpModel('flowers_model_1', fd, [10])  
fm.exec_all(epoch_count=20, report=5)
```

```
Model flowers_model_1 train started:  
Epoch 5: cost=1.604, accuracy=0.244/0.190 (16/16 secs)  
Epoch 10: cost=1.600, accuracy=0.244/0.200 (17/33 secs)  
Epoch 15: cost=1.599, accuracy=0.244/0.230 (16/49 secs)  
Epoch 20: cost=1.598, accuracy=0.244/0.180 (16/65 secs)  
Model flowers_model_1 train ended in 65 secs:  
Model flowers_model_1 test report: accuracy = 0.242, (0 secs)
```

Model flowers_model_1 Visualization



```
추정확률분포 [18,23,18,17,22] => 추정 dandelion : 정답 tulip => X  
추정확률분포 [18,23,18,17,22] => 추정 dandelion : 정답 rose => X  
추정확률분포 [18,23,18,17,22] => 추정 dandelion : 정답 tulip => X
```

그림 1 - 은닉 계층 하나인 경우

```
fm2 = MlpModel('flowers_model_2', fd, [30, 10])
fm2.exec_all(epoch_count=20, report=5)
```

Model flowers_model_2 train started:
 Epoch 5: cost=1.380, accuracy=0.385/0.380 (31/31 secs)
 Epoch 10: cost=1.251, accuracy=0.449/0.430 (28/59 secs)
 Epoch 15: cost=1.161, accuracy=0.516/0.400 (30/89 secs)
 Epoch 20: cost=1.078, accuracy=0.548/0.510 (29/118 secs)
 Model flowers_model_2 train ended in 118 secs:
 Model flowers_model_2 test report: accuracy = 0.452, (1 secs)

Model flowers_model_2 Visualization



추정확률분포 [25,37, 9,14,14] => 추정 dandelion : 정답 rose => X
 추정확률분포 [0, 0, 0, 0,100] => 추정 tulip : 정답 tulip => 0
 추정확률분포 [17,18,23,16,27] => 추정 tulip : 정답 rose => X

그림 2 – 은닉 계층 둘인 경우

```
fm2 = MlpModel('flowers_model_2', fd, [80, 40, 10])
fm2.exec_all(epoch_count=20, report=5)
```

Model flowers_model_2 train started:
 Epoch 5: cost=1.247, accuracy=0.448/0.440 (60/60 secs)
 Epoch 10: cost=1.096, accuracy=0.541/0.510 (89/149 secs)
 Epoch 15: cost=0.952, accuracy=0.616/0.480 (67/216 secs)
 Epoch 20: cost=0.814, accuracy=0.672/0.500 (60/276 secs)
 Model flowers_model_2 train ended in 276 secs:
 Model flowers_model_2 test report: accuracy = 0.451, (0 secs)

Model flowers_model_2 Visualization



추정확률분포 [0, 0,16, 0,84] => 추정 tulip : 정답 tulip => 0
 추정확률분포 [0, 1, 0,95, 4] => 추정 sunflower : 정답 sunflower => 0
 추정확률분포 [2,13, 1, 7,77] => 추정 tulip : 정답 rose => X

그림 3 – 은닉 계층 셋인 경우

```
fm2 = MlpModel('flowers_model_2', fd, [100, 80, 40, 10])
fm2.exec_all(epoch_count=20, report=5)
```

Model flowers_model_2 train started:
 Epoch 5: cost=1.266, accuracy=0.446/0.530 (74/74 secs)
 Epoch 10: cost=1.124, accuracy=0.521/0.600 (72/146 secs)
 Epoch 15: cost=0.960, accuracy=0.612/0.530 (77/223 secs)
 Epoch 20: cost=0.812, accuracy=0.681/0.500 (73/296 secs)
 Model flowers_model_2 train ended in 296 secs:
 Model flowers_model_2 test report: accuracy = 0.461, (0 secs)

Model flowers_model_2 Visualization



추정확률분포 [2, 0.95, 0, 3] => 추정 rose : 정답 tulip => X
 추정확률분포 [11,29,36, 1,24] => 추정 rose : 정답 tulip => X
 추정확률분포 [12, 1, 4,71,12] => 추정 sunflower : 정답 tulip => X

그림 4 - 은닉 계층 넷인 경우

다음은 위의 내용을 정리한 표이다.

	그림 1	그림 2	그림 3	그림 4
은닉 계층 수	1	2	3	4
은닉 계층 폭	[10]	[30,10]	[80, 40, 10]	[100, 80, 40, 10]
정확도	24.2%	45.2%	45.1%	46.1%

표 1 - 은닉 계층 수에 따른 결과

확실히 은닉 계층 수가 늘어남에 따라 정확도가 약 20%가량 늘어나는 모습을 확인할 수 있었다. 하지만, 그림 2 ~ 그림 4 에서는 눈에 띄는 변화는 볼 수 없었다. 한편, 그림 4 에서 정확도가 46.1%로 높게 나왔지만 시각화한 부분에서 정답을 맞추지 못하는 모습을 볼 수 있었다. 이를 통해 정확도는 결과를 대표하는 지표로 그다지 좋은 지표임은 아니라는 점을 알 수 있었다.

2.2 은닉 계층 폭에 변화를 주었을 때

2.1 의 은닉 계층에 폭의 변화를 주었다.

```
fd = FlowersDataset()  
fm = MlpModel('flowers_model_1', fd, [20])  
fm.exec_all(epoch_count=20, report=5)
```

Model flowers_model_1 train started:
Epoch 5: cost=1.604, accuracy=0.246/0.180 (23/23 secs)
Epoch 10: cost=1.600, accuracy=0.246/0.200 (23/46 secs)
Epoch 15: cost=1.599, accuracy=0.246/0.170 (24/70 secs)
Epoch 20: cost=1.598, accuracy=0.246/0.220 (23/93 secs)
Model flowers_model_1 train ended in 93 secs:
Model flowers_model_1 test report: accuracy = 0.248, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [18,24,19,17,22] => 추정 dandelion : 정답 daisy => X
추정확률분포 [18,24,19,17,22] => 추정 dandelion : 정답 tulip => X
추정확률분포 [18,24,19,17,22] => 추정 dandelion : 정답 sunflower => X

그림 5 – 은닉 계층이 [20]인 경우

```
fm2 = MlpModel('flowers_model_2', fd, [60, 20])  
fm2.exec_all(epoch_count=20, report=5)
```

Model flowers_model_2 train started:
Epoch 5: cost=1.536, accuracy=0.283/0.230 (52/52 secs)
Epoch 10: cost=1.391, accuracy=0.362/0.350 (51/103 secs)
Epoch 15: cost=1.231, accuracy=0.467/0.340 (50/153 secs)
Epoch 20: cost=1.108, accuracy=0.533/0.350 (49/202 secs)
Model flowers_model_2 train ended in 202 secs:
Model flowers_model_2 test report: accuracy = 0.364, (0 secs)

Model flowers_model_2 Visualization



추정확률분포 [24,19,23, 9,25] => 추정 tulip : 정답 sunflower => X
추정확률분포 [27,56, 8, 3, 6] => 추정 dandelion : 정답 daisy => X
추정확률분포 [16,21,19,20,25] => 추정 tulip : 정답 daisy => X

그림 6 – 은닉 계층이 [60, 20]인 경우

```
fm2 = MlpModel('flowers_model_2', fd, [100, 50, 10])
fm2.exec_all(epoch_count=20, report=5)
```

Model flowers_model_2 train started:
 Epoch 5: cost=1.269, accuracy=0.450/0.320 (79/79 secs)
 Epoch 10: cost=1.098, accuracy=0.544/0.490 (77/156 secs)
 Epoch 15: cost=0.915, accuracy=0.635/0.510 (77/233 secs)
 Epoch 20: cost=0.772, accuracy=0.695/0.430 (79/312 secs)
 Model flowers_model_2 train ended in 312 secs:
 Model flowers_model_2 test report: accuracy = 0.414, (0 secs)

Model flowers_model_2 Visualization



추정확률분포 [20,14,26,11,30] => 추정 tulip : 정답 rose => X
 추정확률분포 [51, 7,13, 1,28] => 추정 daisy : 정답 daisy => 0
 추정확률분포 [81, 9, 1, 0, 8] => 추정 daisy : 정답 dandelion => X

그림 7 – 은닉 계층이 [100, 50, 10]인 경우

```
fm2 = MlpModel('flowers_model_2', fd, [150, 100, 50, 10])
fm2.exec_all(epoch_count=20, report=5)
```

Model flowers_model_2 train started:
 Epoch 5: cost=1.241, accuracy=0.450/0.470 (115/115 secs)
 Epoch 10: cost=1.066, accuracy=0.552/0.480 (110/225 secs)
 Epoch 15: cost=0.874, accuracy=0.642/0.470 (104/329 secs)
 Epoch 20: cost=0.691, accuracy=0.728/0.530 (107/436 secs)
 Model flowers_model_2 train ended in 436 secs:
 Model flowers_model_2 test report: accuracy = 0.432, (0 secs)

Model flowers_model_2 Visualization



추정확률분포 [1, 1, 0.98, 0] => 추정 sunflower : 정답 sunflower => 0
 추정확률분포 [13,34, 2,48, 2] => 추정 sunflower : 정답 tulip => X
 추정확률분포 [54, 9, 0.34, 3] => 추정 daisy : 정답 tulip => X

그림 8 – 은닉 계층이 [150, 100, 50, 10]인 경우

다음은 위의 내용을 정리한 표이다.

	그림 5	그림 6	그림 7	그림 8
은닉 계층 수	1	2	3	4
은닉 계층 폭	[20]	[60,20]	[100, 50, 10]	[100, 80, 40, 10]
정확도	24.8%(+0.6%)	36.4%(-8.8%)	41.4%(-3.7%)	43.2%(-2.9%)

표 2 – 은닉 계층 폭에 따른 결과

그림 5 에서 소폭 증가한 것을 제외하면, 전부 폭이 넓어졌음에도 정확도가 하락하는 현상을 보였다.

2.3 하이퍼 파라미터에 변화를 주었을 때

그림 2 의 은닉 계층이 [30,10]인 모델을 이용하여 하이퍼 파라미터에 변화를 주었다.

2.3.1 epoch 에 변화를 주었을 때

```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=10, report=5)
```

Model flowers_model_1 train started:
 Epoch 5: cost=1.602, accuracy=0.248/0.240 (33/33 secs)
 Epoch 10: cost=1.596, accuracy=0.250/0.280 (37/70 secs)
 Model flowers_model_1 train ended in 70 secs:
 Model flowers_model_1 test report: accuracy = 0.224, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [19,22,19,19,21] => 추정 dandelion : 정답 tulip => X
 추정확률분포 [19,22,19,19,21] => 추정 dandelion : 정답 dandelion => O
 추정확률분포 [19,22,19,19,21] => 추정 dandelion : 정답 rose => X

그림 9 – epoch 이 10 인 경우

```
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=20, report=5)
```

Model flowers_model_1 train started:
 Epoch 5: cost=1.430, accuracy=0.348/0.330 (35/35 secs)
 Epoch 10: cost=1.263, accuracy=0.442/0.360 (36/71 secs)
 Epoch 15: cost=1.172, accuracy=0.491/0.350 (36/107 secs)
 Epoch 20: cost=1.091, accuracy=0.526/0.440 (34/141 secs)
 Model flowers_model_1 train ended in 141 secs:
 Model flowers_model_1 test report: accuracy = 0.394, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [17,26,14,25,18] => 추정 dandelion : 정답 tulip => X
 추정확률분포 [67,30, 1, 1, 1] => 추정 daisy : 정답 rose => X
 추정확률분포 [10, 6,45, 1,39] => 추정 rose : 정답 tulip => X

그림 10 – epoch 이 20 인 경우

```
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=50, report=25)
```

Model flowers_model_1 train started:
 Epoch 25: cost=1.001, accuracy=0.579/0.490 (168/168 secs)
 Epoch 50: cost=0.836, accuracy=0.657/0.490 (181/349 secs)
 Model flowers_model_1 train ended in 349 secs:
 Model flowers_model_1 test report: accuracy = 0.400, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [18,24,29, 5,25] => 추정 rose : 정답 dandelion => X
 추정확률분포 [98, 2, 0, 0, 0] => 추정 daisy : 정답 daisy => 0
 추정확률분포 [9,24,34,16,17] => 추정 rose : 정답 rose => 0

그림 11 – epoch 이 50 인 경우

```
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=100, report=25)
```

Model flowers_model_1 train started:
 Epoch 25: cost=1.016, accuracy=0.572/0.380 (189/189 secs)
 Epoch 50: cost=0.741, accuracy=0.702/0.450 (187/376 secs)
 Epoch 75: cost=0.554, accuracy=0.776/0.460 (183/559 secs)
 Epoch 100: cost=0.492, accuracy=0.807/0.440 (185/744 secs)
 Model flowers_model_1 train ended in 744 secs:
 Model flowers_model_1 test report: accuracy = 0.441, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [6, 7, 48, 2, 37] => 추정 rose : 정답 dandelion => X
 추정확률분포 [29, 45, 2, 18, 6] => 추정 dandelion : 정답 sunflower => X
 추정확률분포 [64, 33, 3, 0, 0] => 추정 daisy : 정답 sunflower => X

그림 12 – epoch 이 100 인 경우

다음은 위의 내용을 정리한 표이다.

	그림 9	그림 10	그림 11	그림 12
epoch	10	20	50	100
은닉 계층 수	2	2	2	2
은닉 계층 폭	[30,10]	[30,10]	[30,10]	[30,10]
정확도	22.4%	39.4%	40.0%	44.1%

표 3 – epoch 의 변화에 따른 결과

확실히 epoch 가 늘어날수록 정확도가 증가하는 경향을 보였지만, 어느정도 진행된 상태에서는 그 변화량이 줄어들었다.

3.3.2 학습률에 변화를 주었을 때

```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=20, learning_rate=0.1, report=5)
```

Model flowers_model_1 train started:
 Epoch 5: cost=1.601, accuracy=0.239/0.250 (29/29 secs)
 Epoch 10: cost=1.600, accuracy=0.244/0.240 (29/58 secs)
 Epoch 15: cost=1.600, accuracy=0.236/0.250 (32/90 secs)
 Epoch 20: cost=1.601, accuracy=0.239/0.260 (29/119 secs)
 Model flowers_model_1 train ended in 119 secs:
 Model flowers_model_1 test report: accuracy = 0.216, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [19,23,18,17,23] => 추정 tulip : 정답 dandelion => X
 추정확률분포 [19,23,18,17,23] => 추정 tulip : 정답 daisy => X
 추정확률분포 [19,23,18,17,23] => 추정 tulip : 정답 rose => X

그림 13 – 학습률이 0.1 인 경우

```
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=20, learning_rate=0.01, report=5)
```

Model flowers_model_1 train started:
 Epoch 5: cost=1.598, accuracy=0.243/0.260 (30/30 secs)
 Epoch 10: cost=1.598, accuracy=0.243/0.290 (29/59 secs)
 Epoch 15: cost=1.598, accuracy=0.243/0.230 (28/87 secs)
 Epoch 20: cost=1.598, accuracy=0.243/0.230 (29/116 secs)
 Model flowers_model_1 train ended in 116 secs:
 Model flowers_model_1 test report: accuracy = 0.245, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [18,24,18,17,23] => 추정 dandelion : 정답 tulip => X
 추정확률분포 [18,24,18,17,23] => 추정 dandelion : 정답 tulip => X
 추정확률분포 [18,24,18,17,23] => 추정 dandelion : 정답 dandelion => 0

그림 14 – 학습률이 0.01 인 경우

```
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=20, learning_rate=0.001, report=5)
```

Model flowers_model_1 train started:
 Epoch 5: cost=1.361, accuracy=0.394/0.390 (30/30 secs)
 Epoch 10: cost=1.251, accuracy=0.449/0.450 (29/59 secs)
 Epoch 15: cost=1.184, accuracy=0.480/0.420 (29/88 secs)
 Epoch 20: cost=1.102, accuracy=0.530/0.470 (29/117 secs)
 Model flowers_model_1 train ended in 117 secs:
 Model flowers_model_1 test report: accuracy = 0.425, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [16,20,18,22,24] => 추정 tulip : 정답 tulip => 0
 추정확률분포 [19,22,19,19,22] => 추정 dandelion : 정답 tulip => X
 추정확률분포 [43,54, 2, 0, 1] => 추정 dandelion : 정답 tulip => X

그림 15 – 학습률이 0.001 인 경우

```
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=20, learning_rate=0.0001, report=5)
```

Model flowers_model_1 train started:
 Epoch 25: cost=1.016, accuracy=0.572/0.380 (189/189 secs)
 Epoch 50: cost=0.741, accuracy=0.702/0.450 (187/376 secs)
 Epoch 75: cost=0.554, accuracy=0.776/0.460 (183/559 secs)
 Epoch 100: cost=0.492, accuracy=0.807/0.440 (185/744 secs)
 Model flowers_model_1 train ended in 744 secs:
 Model flowers_model_1 test report: accuracy = 0.441, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [6, 7,48, 2,37] => 추정 rose : 정답 dandelion => X
 추정확률분포 [29,45, 2,18, 6] => 추정 dandelion : 정답 sunflower => X
 추정확률분포 [64,33, 3, 0, 0] => 추정 daisy : 정답 sunflower => X

그림 16 – 학습률이 0.0001 인 경우

다음은 위의 내용을 정리한 표이다.

	그림 13	그림 14	그림 15	그림 16
학습률	0.1	0.01	0.001	0.0001
은닉 계층 수	2	2	2	2
은닉 계층 폭	[30,10]	[30,10]	[30,10]	[30,10]
정확도	21.6%	24.5%	42.5%	44.1%

표 4 – 학습률의 변화에 따른 결과

예상대로 학습률이 줄어들수록 정확도가 증가하였다.

3.3.3 미니배치 사이즈에 변화를 주었을 때

```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=20, batch_size=1, report=5)
```

Model flowers_model_1 train started:
 Epoch 5: cost=1.597, accuracy=0.251/0.220 (228/228 secs)
 Epoch 10: cost=1.597, accuracy=0.251/0.300 (238/466 secs)
 Epoch 15: cost=1.597, accuracy=0.251/0.250 (218/684 secs)
 Epoch 20: cost=1.597, accuracy=0.251/0.240 (228/912 secs)
 Model flowers_model_1 train ended in 912 secs:
 Model flowers_model_1 test report: accuracy = 0.213, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [18,25,18,17,22] => 추정 dandelion : 정답 rose => X
 추정확률분포 [18,25,18,17,22] => 추정 dandelion : 정답 rose => X
 추정확률분포 [18,25,18,17,22] => 추정 dandelion : 정답 dandelion => 0

그림 17 – 미니배치가 1 인 경우

```
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=20, batch_size=10, report=5)
```

Model flowers_model_1 train started:
 Epoch 5: cost=1.362, accuracy=0.395/0.350 (31/31 secs)
 Epoch 10: cost=1.261, accuracy=0.419/0.360 (30/61 secs)
 Epoch 15: cost=1.177, accuracy=0.470/0.360 (34/95 secs)
 Epoch 20: cost=1.114, accuracy=0.491/0.420 (32/127 secs)
 Model flowers_model_1 train ended in 127 secs:
 Model flowers_model_1 test report: accuracy = 0.414, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [2,20, 0,69, 8] => 추정 sunflower : 정답 dandelion => X
 추정확률분포 [17,11,27,10,35] => 추정 tulip : 정답 tulip => 0
 추정확률분포 [0, 0,14, 0,86] => 추정 tulip : 정답 tulip => 0

그림 18 – 미니배치가 10 인 경우

```
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=20, batch_size=50, report=5)
```

Model flowers_model_1 train started:
 Epoch 5: cost=1.373, accuracy=0.388/0.420 (16/16 secs)
 Epoch 10: cost=1.281, accuracy=0.414/0.380 (16/32 secs)
 Epoch 15: cost=1.236, accuracy=0.440/0.450 (14/46 secs)
 Epoch 20: cost=1.191, accuracy=0.474/0.360 (14/60 secs)
 Model flowers_model_1 train ended in 60 secs:
 Model flowers_model_1 test report: accuracy = 0.402, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [1, 2,35, 1,61] => 추정 tulip : 정답 tulip => 0
 추정확률분포 [34,46,12, 1, 7] => 추정 dandelion : 정답 rose => X
 추정확률분포 [3, 4,38, 2,53] => 추정 tulip : 정답 rose => X

그림 19 – 미니배치가 50 인 경우

```
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=20, batch_size=100, report=5)
```

Model flowers_model_1 train started:
 Epoch 5: cost=1.491, accuracy=0.324/0.180 (14/14 secs)
 Epoch 10: cost=1.360, accuracy=0.393/0.360 (14/28 secs)
 Epoch 15: cost=1.298, accuracy=0.412/0.260 (14/42 secs)
 Epoch 20: cost=1.258, accuracy=0.428/0.460 (13/55 secs)
 Model flowers_model_1 train ended in 55 secs:
 Model flowers_model_1 test report: accuracy = 0.376, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [53,42, 4, 0, 1] => 추정 daisy : 정답 daisy => 0
 추정확률분포 [33,34,15, 7,12] => 추정 dandelion : 정답 rose => X
 추정확률분포 [14,24,10,31,22] => 추정 sunflower : 정답 tulip => X

그림 20 – 미니배치가 100 인 경우

다음은 위의 내용을 정리한 표이다.

	그림 17	그림 18	그림 19	그림 20
미니배치	1	10	50	100
은닉 계층 수	2	2	2	2
은닉 계층 폭	[30,10]	[30,10]	[30,10]	[30,10]
정확도	21.3%	41.4%	40.2%	37.6%

표 5 – 미니배치의 변화에 따른 결과

미니배치가 10 일 때 41.4%로 최대 정확도가 나왔다.

3.4 활성화 함수에 변화를 주었을 때

```
fd = FlowersDataset()  
fm = MlpModel('flowers_model_1', fd, [30,10])  
fm.exec_all(epoch_count=30,report=10)
```

Model flowers_model_1 train started:
Epoch 10: cost=1.238, accuracy=0.451/0.360 (59/59 secs)
Epoch 20: cost=1.074, accuracy=0.538/0.410 (58/117 secs)
Epoch 30: cost=0.961, accuracy=0.593/0.430 (59/176 secs)
Model flowers_model_1 train ended in 176 secs:
Model flowers_model_1 test report: accuracy = 0.419, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [56,37, 3, 0, 3] => 추정 daisy : 정답 dandelion => X
추정확률분포 [1, 0, 1, 1,97] => 추정 tulip : 정답 dandelion => X
추정확률분포 [31,48, 7, 1,12] => 추정 dandelion : 정답 dandelion => 0

그림 21 – 활성화 함수가 relu 인 경우

```
fd = FlowersDataset()  
fm = MlpModel('flowers_model_1', fd, [30,10])  
fm.exec_all(epoch_count=30,report=10)
```

Model flowers_model_1 train started:
Epoch 10: cost=1.594, accuracy=0.246/0.220 (57/57 secs)
Epoch 20: cost=1.583, accuracy=0.255/0.250 (62/119 secs)
Epoch 30: cost=1.583, accuracy=0.261/0.340 (65/184 secs)
Model flowers_model_1 train ended in 184 secs:
Model flowers_model_1 test report: accuracy = 0.250, (1 secs)

Model flowers_model_1 Visualization



추정확률분포 [18,25,18,17,22] => 추정 dandelion : 정답 dandelion => 0
추정확률분포 [18,25,18,17,23] => 추정 dandelion : 정답 dandelion => 0
추정확률분포 [18,25,18,17,23] => 추정 dandelion : 정답 dandelion => 0

그림 22 – 활성화 함수가 tanh 인 경우

다음은 위의 내용을 정리한 표이다.

	그림 21	그림 22
활성화 함수	relu	tanh
은닉 계층 수	2	2
은닉 계층 폭	[30,10]	[30,10]
정확도	42.5%	26.0%

표 6 - 활성화 함수의 변화에 따른 결과

relu 함수가 훨씬 높게 나왔다. 얼핏 시각화를 보면, tanh 가 학습이 잘된 것처럼 보이지만, 확률분포를 보면 학습이 잘 되지않은 것을 확인할 수 있다.

3. 결론

3.1 은닉 계층의 변화에 따른 차이점?

은닉 계층의 수가 증가함에 따라 대체적으로 정확도가 상승하는 점을 확인할 수 있었다. 반면, 예상과는 달리 은닉 계층의 폭을 늘렸을 때는 대체적으로 감소했다. 이는 은닉 계층의 수와 폭이 정확도에 큰 영향을 주며, 충분한 데이터가 있지 않으면 오히려 품질 저하를 초래할 수 있다는 사실을 보여준다.

3.2 하이퍼 파라미터의 변화에 따른 차이점?

epoch 이 증가함에 따라 대체적으로 정확도가 증가함을 알 수 있었다. 또한, 학습률이 적을수록 정확도가 증가하는 경향을 보였다. 그리고 미니배치의 경우에는 미니배치가 1 인 경우를 제외하면, 대체적으로 비슷했다. 미니배치가 1 인 경우는 epoch 당 학습횟수가 많아짐에 따라 학습이 부정확하게 된 것으로 예측된다. 전체적으로 하이퍼 파라미터를 통해 급격한 정확도 향상은 찾아볼 수 없었다.

3.3 활성화 함수로 relu 함수와 tanh 함수를 썼을 때 차이점?

우선 relu 함수의 정확도가 훨씬 높게 나왔다. 이는 tanh 함수의 backpropagation 중 생기는 문제인 Vanishing Gradient 때문이다. 이 때문에 cost function 의 gradient 항이 0 이나 0 에 가까워져 학습이 불가능해지게 된다. 그림 22 의 cost 를 보면 1.583 에서 더 이상 감소하지 않기 때문에 이 현상이 발생하여 학습이 진행되지 않은 모습을 알 수 있다. 이를 해결하기 위해 relu 함수를 쓰는데, relu 는 gradient 가 x 값이 0 이하일 때는, 0 이고, positive input 일 때는 항상 1 이기 때문에 이 현상이 발생하지 않기 때문이다.

3.1 다층 퍼셉트론 신경망이 이미지 처리에 부적합한 이유는?

컬러 이미지는 RGB 색상으로 표현할 수 있다. 이미지는 보통 [행, 열, 채널]의 3 차원 구조의 픽셀 정보로 표현되는데 다층 퍼셉트론에서는 입력 이미지를 1 차원 벡터로 변환해 처리하기 때문에 이미지의 모든 정보를 담을 수 없다. 따라서 이미지 처리에 부적합하며, CNN 을 사용하면 이미지의 3 차원 정보를 담을 수 있기 때문에 이를 사용하게 된다.

3.2 신경망의 파라미터가 많아지면 어떤 문제가 생길까?

이미지를 다룰 때는 이미지 해상도 설정과 은닉 계층 구성에 따라 필요한 파라미터 수가 매우 커질 수 있다. 특히, 표 2 를 보게되면 은닉 계층 구성에 따라 파라미터수가 매우 늘었지만 대체적으로 정확도가 감소하는 경향을 보인다. 이는 많은 파라미터를 제대로 학습시키기 위해서는 데이터도 그만큼 많이 필요함을 뜻한다. 또한, 데이터까지 많아지게 된다면 계산 부담은 이중으로 늘어난다. 따라서 이런 모델은 엄청난 메모리와 계산 시간을 소모하면서도 좋은 결과를 내기가 어려운 문제가 있다.

3.3 데이터를 학습/검증/평가 셋으로 구분지어야 하는 이유가 무엇일까?

우선 정의를 알아보자. 학습 셋은 말 그대로 학습할 데이터이다. 검증 셋은 학습 과정에서 중간 점검을 할 수 있는 데이터이며, 평가 데이터는 학습 후에 학습 품질을 확인하기 위한 데이터이다.

딥러닝 모델에서 중요한 것은 학습 과정에서의 성적보다 실전에서의 성적이다. 검증 데이터나 평가 데이터를 잘 활용해 학습 결과를 검증하고 평가하여 실전 적용에 알맞은 모습으로 학습되고 있는지 확인이 중요하다.

한편, 평가 셋으로 학습해서는 안된다. 평가 측정항목에서 이상할 정도로 좋은 결과가 나온다면 실수로 평가 세트로 학습했다는 증거일 수 있다. 또한, 정확도가 높다면 테스트 데이터가 학습 세트로 유출될 가능성이 있다. 이렇게 정확도에 크게 영향을 줄 수 있기 때문에, 세 데이터는 서로 겹치지 않으면서 골고루 섞이게 준비 되어야 한다. 만약 형편이 여의치 않으면, 학습 데이터나 평가 데이터를 검증 데이터로 이용하기도 하지만, 학습 데이터와 평가 데이터만은 분명히 갈라놓아야 한다.