

5장. 다층 퍼셉트론 모델 구조: 꽃 이미지 분류 신경망

1. 개요

다층 퍼셉트론을 이용한 꽃 이미지 분류 신경망 실험에서는 은닉 계층 수와 폭 조절 및 하이퍼 파라미터 튜닝을 통해 성능을 개선하는 작업을 수행하였다. 이 실험에서는 4242개의 꽃 이미지가 포함된 데이터셋을 이용하며, 이는 카모마일, 튼リップ, 장미, 해바라기, 민들레의 다섯 가지 클래스로 구분되어 있다. 다층 퍼셉트론은 이미지 처리에 부적합한 모습을 보이는데, 이 실험을 통해 그 이유를 분석해보고자 한다. 더불어 활성화 함수로 relu와 tanh를 사용할 때 나타나는 결과 차이, 파라미터가 많아지게 될 때의 문제점, 데이터를 학습데이터, 검증데이터, 평가데이터로 구분짓는 이유에 대해서도 고민해보고자 한다.

2. 학습 결과

2.1 은닉 계층 개수에 따른 비교

구분	내용			
조건	- 에포크 횟수(epoch_count) : 10 - 배치 크기(batch_size) : 10 - 학습률(learning rate) : 0.001			
	은닉계층개수= 0	은닉계층개수= 1	은닉계층개수= 2	은닉계층개수= 3
데이터 정확도	34.8%	24.7%	38.5%	41.1%
소요 시간	30s	33s	33s	34s

결과 화면

```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [])
fm.exec_all(epoch_count=10, report=2)
```

Model flowers_model_1 train started:
 Epoch 2: cost=14.450, accuracy=0.372/0.400 (6/6 secs)
 Epoch 4: cost=13.709, accuracy=0.405/0.330 (6/12 secs)
 Epoch 6: cost=12.641, accuracy=0.451/0.290 (6/18 secs)
 Epoch 8: cost=12.200, accuracy=0.470/0.400 (6/24 secs)
 Epoch 10: cost=11.734, accuracy=0.490/0.410 (6/30 secs)
 Model flowers_model_1 train ended in 30 secs:
 Model flowers_model_1 test report: accuracy = 0.348, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [0, 0,100, 0, 0] => 추정 rose : 정답 dandelion => X
 추정확률분포 [0,100, 0, 0, 0] => 추정 dandelion : 정답 dandelion => 0
 추정확률분포 [0, 0,100, 0, 0] => 추정 rose : 정답 tulip => X

```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [10])
fm.exec_all(epoch_count=10, report=2)
```

Model flowers_model_1 train started:
 Epoch 2: cost=1.608, accuracy=0.243/0.220 (7/7 secs)
 Epoch 4: cost=1.606, accuracy=0.243/0.240 (6/13 secs)
 Epoch 6: cost=1.604, accuracy=0.243/0.290 (7/20 secs)
 Epoch 8: cost=1.603, accuracy=0.243/0.210 (7/27 secs)
 Epoch 10: cost=1.603, accuracy=0.243/0.260 (6/33 secs)
 Model flowers_model_1 train ended in 33 secs:
 Model flowers_model_1 test report: accuracy = 0.247, (0 secs)

Model flowers_model_1 Visualization



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

```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [10,10])
fm.exec_all(epoch_count=10, report=2)
```

Model flowers_model_1 train started:
 Epoch 2: cost=1.482, accuracy=0.341/0.390 (6/6 secs)
 Epoch 4: cost=1.360, accuracy=0.404/0.470 (7/13 secs)
 Epoch 6: cost=1.305, accuracy=0.411/0.430 (6/19 secs)
 Epoch 8: cost=1.276, accuracy=0.436/0.390 (7/26 secs)
 Epoch 10: cost=1.248, accuracy=0.447/0.460 (7/33 secs)
 Model flowers_model_1 train ended in 33 secs:
 Model flowers_model_1 test report: accuracy = 0.385, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [2,12, 2,76, 8] => 추정 sunflower : 정답 sunflower => 0
 추정확률분포 [26,17,38, 1,19] => 추정 rose : 정답 tulip => X
 추정확률분포 [7,16,26,13,38] => 추정 tulip : 정답 tulip => 0

```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [10,10,10])
fm.exec_all(epoch_count=10, report=2)
```

Model flowers_model_1 train started:
 Epoch 2: cost=1.500, accuracy=0.310/0.300 (6/6 secs)
 Epoch 4: cost=1.327, accuracy=0.405/0.430 (7/13 secs)
 Epoch 6: cost=1.274, accuracy=0.418/0.420 (7/20 secs)
 Epoch 8: cost=1.240, accuracy=0.438/0.470 (7/27 secs)
 Epoch 10: cost=1.212, accuracy=0.451/0.440 (7/34 secs)
 Model flowers_model_1 train ended in 34 secs:
 Model flowers_model_1 test report: accuracy = 0.411, (0 secs)

Model flowers_model_1 Visualization




추정확률분포 [53,34, 6, 2, 6] => 추정 daisy : 정답 daisy => 0
 추정확률분포 [48,28,12, 2,10] => 추정 daisy : 정답 tulip => X
 추정확률분포 [25,58, 2,12, 4] => 추정 dandelion : 정답 dandelion => 0

2.2 은닉 계층 폭에 따른 비교

은닉 계층 간 폭이 동일한 경우, 증가한 경우, 감소한 경우를 비교해보기 위해 은닉 계층의 개수를 2개로 설정하고 실험을 진행하였다.

(1) 은닉 계층 간 폭이 동일한 경우

폭의 크기를 조절해가며 결과를 비교하였다.

구분	내용		
조건	<ul style="list-style-type: none"> - 에포크 횟수(epoch_count) : 10 - 배치 크기(batch_size) : 10 - 학습률(learning rate) : 0.001 - 은닉 계층 개수 : 2 		
	은닉계층 = [4,4]	은닉계층 = [10,10]	은닉계층 = [30,30]
데이터 정확도	38.0%	36.7%	37.9%
소요 시간	25s	34s	45s
결과 화면	<pre>fd = FlowersDataset() fm = MlpModel('flowers_model_1', fd, [4,4]) fm.exec_all(epoch_count=10, report=2)</pre> <p>Model flowers_model_1 train started: Epoch 2: cost=1.583, accuracy=0.270/0.290 (5/5 secs) Epoch 4: cost=1.531, accuracy=0.292/0.270 (5/10 secs) Epoch 6: cost=1.499, accuracy=0.317/0.280 (5/15 secs) Epoch 8: cost=1.481, accuracy=0.356/0.270 (5/20 secs) Epoch 10: cost=1.458, accuracy=0.376/0.320 (5/25 secs) Model flowers_model_1 train ended in 25 secs: Model flowers_model_1 test report: accuracy = 0.380, (0 secs)</p> <p>Model flowers_model_1 Visualization</p>  <p>추정확률분포 [20,22,20,16,22] => 추정 tulip : 정답 dandelion => X 추정확률분포 [20,21,21,16,22] => 추정 tulip : 정답 tulip => 0 추정확률분포 [20,21,21,16,22] => 추정 tulip : 정답 daisy => X</p> <pre>fd = FlowersDataset() fm = MlpModel('flowers_model_1', fd, [10,10]) fm.exec_all(epoch_count=10, report=2)</pre> <p>Model flowers_model_1 train started: Epoch 2: cost=1.491, accuracy=0.301/0.370 (7/7 secs) Epoch 4: cost=1.358, accuracy=0.381/0.380 (7/14 secs) Epoch 6: cost=1.310, accuracy=0.416/0.300 (6/20 secs) Epoch 8: cost=1.277, accuracy=0.419/0.390 (7/27 secs) Epoch 10: cost=1.268, accuracy=0.414/0.360 (7/34 secs) Model flowers_model_1 train ended in 34 secs: Model flowers_model_1 test report: accuracy = 0.367, (0 secs)</p> <p>Model flowers_model_1 Visualization</p>  <p>추정확률분포 [55,33, 6, 1, 7] => 추정 daisy : 정답 dandelion => X 추정확률분포 [46,33, 8, 2,11] => 추정 daisy : 정답 tulip => X 추정확률분포 [19,21,20,18,23] => 추정 tulip : 정답 tulip => 0</p>		

	<pre>fd = FlowersDataset() fm = MlpModel('flowers_model_1', fd, [30,30]) fm.exec_all(epoch_count=10, report=2)</pre> <p>Model flowers_model_1 train started:</p> <p>Epoch 2: cost=1.440, accuracy=0.368/0.400 (9/9 secs)</p> <p>Epoch 4: cost=1.324, accuracy=0.408/0.410 (9/18 secs)</p> <p>Epoch 6: cost=1.289, accuracy=0.425/0.350 (9/27 secs)</p> <p>Epoch 8: cost=1.240, accuracy=0.450/0.370 (9/36 secs)</p> <p>Epoch 10: cost=1.225, accuracy=0.458/0.370 (9/45 secs)</p> <p>Model flowers_model_1 train ended in 45 secs:</p> <p>Model flowers_model_1 test report: accuracy = 0.379, (0 secs)</p> <p>Model flowers_model_1 Visualization</p> <div>    </div> <p>추정확률분포 [3,11,31,10,45] => 추정 tulip : 정답 sunflower => X</p> <p>추정확률분포 [18,20,21,18,23] => 추정 tulip : 정답 sunflower => X</p> <p>추정확률분포 [75,21, 1, 0, 3] => 추정 daisy : 정답 daisy => 0</p>
--	---

(2) 은닉 계층 간 폭이 증가한 경우

증가폭을 달리하며 실험을 진행하였다.

구분	내용		
조건	<ul style="list-style-type: none"> - 에포크 횟수(epoch_count) : 10 - 배치 크기(batch_size) : 10 - 학습률(learning rate) : 0.001 - 은닉 계층 개수 : 2 		
	은닉계층 = [10,20]	은닉계층 = [10,30]	은닉계층 = [10,50]
데이터 정확도	38.3%	42.8%	29.4%
소요 시간	31s	36s	35s

결과 화면

```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [10,20])
fm.exec_all(epoch_count=10, report=2)
```

Model flowers_model_1 train started:
 Epoch 2: cost=1.514, accuracy=0.288/0.270 (6/6 secs)
 Epoch 4: cost=1.416, accuracy=0.389/0.280 (6/12 secs)
 Epoch 6: cost=1.386, accuracy=0.414/0.270 (6/18 secs)
 Epoch 8: cost=1.363, accuracy=0.414/0.400 (7/25 secs)
 Epoch 10: cost=1.336, accuracy=0.427/0.320 (6/31 secs)
 Model flowers_model_1 train ended in 31 secs:
 Model flowers_model_1 test report: accuracy = 0.383, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [40,47, 7, 1, 5] => 추정 dandelion : 정답 daisy => X
 추정확률분포 [17,21,17,24,21] => 추정 sunflower : 정답 sunflower => 0
 추정확률분포 [27,36,13, 9,16] => 추정 dandelion : 정답 dandelion => 0




```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [10,30])
fm.exec_all(epoch_count=10, report=2)
```

Model flowers_model_1 train started:
 Epoch 2: cost=1.563, accuracy=0.305/0.400 (7/7 secs)
 Epoch 4: cost=1.461, accuracy=0.370/0.350 (8/15 secs)
 Epoch 6: cost=1.349, accuracy=0.414/0.450 (7/22 secs)
 Epoch 8: cost=1.294, accuracy=0.416/0.320 (7/29 secs)
 Epoch 10: cost=1.266, accuracy=0.443/0.370 (7/36 secs)
 Model flowers_model_1 train ended in 36 secs:
 Model flowers_model_1 test report: accuracy = 0.428, (1 secs)

Model flowers_model_1 Visualization



추정확률분포 [31,15,16,15,23] => 추정 daisy : 정답 dandelion => X
 추정확률분포 [28,13,23, 5,31] => 추정 tulip : 정답 daisy => X
 추정확률분포 [1, 7, 1,82,10] => 추정 sunflower : 정답 dandelion => X

	<pre>fd = FlowersDataset() fm = MlpModel('flowers_model_1', fd, [10,50]) fm.exec_all(epoch_count=10, report=2)</pre> <p>Model flowers_model_1 train started:</p> <p>Epoch 2: cost=1.598, accuracy=0.237/0.280 (7/7 secs)</p> <p>Epoch 4: cost=1.588, accuracy=0.246/0.220 (7/14 secs)</p> <p>Epoch 6: cost=1.551, accuracy=0.291/0.340 (7/21 secs)</p> <p>Epoch 8: cost=1.525, accuracy=0.304/0.240 (7/28 secs)</p> <p>Epoch 10: cost=1.511, accuracy=0.310/0.260 (7/35 secs)</p> <p>Model flowers_model_1 train ended in 35 secs:</p> <p>Model flowers_model_1 test report: accuracy = 0.294, (0 secs)</p> <p>Model flowers_model_1 Visualization</p> <div>    </div> <p>추정확률분포 [37,43,11, 0, 8] => 추정 dandelion : 정답 rose => X</p> <p>추정확률분포 [17,21,19,20,23] => 추정 tulip : 정답 tulip => 0</p> <p>추정확률분포 [44,50, 4, 0, 2] => 추정 dandelion : 정답 dandelion => 0</p>
--	--

(3) 은닉 계층 간 폭이 감소한 경우

감소폭을 달리하며 실험을 진행하였다.

구분	내용		
조건	- 에포크 횟수(epoch_count) : 10 - 배치 크기(batch_size) : 10 - 학습률(learning rate) : 0.001 - 은닉 계층 개수 : 2		
	은닉계층 = [50,10]	은닉계층 = [30,10]	은닉계층 = [20,10]
데이터 정확도	30.1%	39.1%	39.6%
소요 시간	61s	48s	43s

결과 화면

```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [50,10])
fm.exec_all(epoch_count=10, report=2)
```

Model flowers_model_1 train started:
 Epoch 2: cost=1.607, accuracy=0.242/0.160 (12/12 secs)
 Epoch 4: cost=1.604, accuracy=0.243/0.250 (13/25 secs)
 Epoch 6: cost=1.596, accuracy=0.243/0.240 (12/37 secs)
 Epoch 8: cost=1.556, accuracy=0.250/0.360 (12/49 secs)
 Epoch 10: cost=1.532, accuracy=0.308/0.300 (12/61 secs)
 Model flowers_model_1 train ended in 61 secs:
 Model flowers_model_1 test report: accuracy = 0.301, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [18,21,19,20,22] => 추정 tulip : 정답 rose => X
 추정확률분포 [23,26,22,10,19] => 추정 dandelion : 정답 dandelion => 0
 추정확률분포 [18,21,19,20,22] => 추정 tulip : 정답 dandelion => X


```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [30,10])
fm.exec_all(epoch_count=10, report=2)
```

Model flowers_model_1 train started:
 Epoch 2: cost=1.585, accuracy=0.261/0.330 (10/10 secs)
 Epoch 4: cost=1.480, accuracy=0.329/0.320 (9/19 secs)
 Epoch 6: cost=1.347, accuracy=0.410/0.430 (10/29 secs)
 Epoch 8: cost=1.303, accuracy=0.422/0.370 (9/38 secs)
 Epoch 10: cost=1.259, accuracy=0.444/0.380 (10/48 secs)
 Model flowers_model_1 train ended in 48 secs:
 Model flowers_model_1 test report: accuracy = 0.391, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [0,20, 0,79, 1] => 추정 sunflower : 정답 sunflower => 0
 추정확률분포 [21,21,26, 4,28] => 추정 tulip : 정답 daisy => X
 추정확률분포 [49,19,29, 0, 3] => 추정 daisy : 정답 daisy => 0

	<pre>fd = FlowersDataset() fm = MlpModel('flowers_model_1', fd, [20,10]) fm.exec_all(epoch_count=10, report=2)</pre> <p>Model flowers_model_1 train started:</p> <p>Epoch 2: cost=1.429, accuracy=0.358/0.360 (8/8 secs)</p> <p>Epoch 4: cost=1.344, accuracy=0.396/0.370 (9/17 secs)</p> <p>Epoch 6: cost=1.303, accuracy=0.419/0.450 (9/26 secs)</p> <p>Epoch 8: cost=1.273, accuracy=0.440/0.430 (8/34 secs)</p> <p>Epoch 10: cost=1.241, accuracy=0.448/0.460 (9/43 secs)</p> <p>Model flowers_model_1 train ended in 43 secs:</p> <p>Model flowers_model_1 test report: accuracy = 0.396, (0 secs)</p> <p>Model flowers_model_1 Visualization</p>  <p>추정확률분포 [20,21,19,20,21] => 추정 tulip : 정답 dandelion => X</p> <p>추정확률분포 [13, 7,40, 5,35] => 추정 rose : 정답 sunflower => X</p> <p>추정확률분포 [14,20,26, 7,33] => 추정 tulip : 정답 tulip => 0</p>
--	---

2.3 하이퍼파라미터에 따른 비교

앞선 실험에서 높은 정확도를 보였던 은닉 계층 [10,30] 조건에서 실험을 진행하였다.

(1) 비교 대상 : 에포크 횟수

구분	내용		
조건	<ul style="list-style-type: none"> - 배치 크기(batch_size) : 10 - 학습률(learning rate) : 0.001 - 은닉 계층 개수 : 2 - 은닉 계층 : [10,30] 		
	에포크 횟수 (epoch_count) : 10	에포크 횟수 (epoch_count) : 30	에포크 횟수 (epoch_count) : 50
데이터 정확도	39.7%	42.2%	39.3%
소요 시간	36s	107s	178s

결과 화면

```
fd = FlowersDataset()
fm = MlpModel('flowers_model_1', fd, [10,30])
fm.exec_all(epoch_count=10, report=2)
```

Model flowers_model_1 train started:
 Epoch 2: cost=1.508, accuracy=0.309/0.360 (7/7 secs)
 Epoch 4: cost=1.384, accuracy=0.393/0.390 (8/15 secs)
 Epoch 6: cost=1.324, accuracy=0.396/0.370 (7/22 secs)
 Epoch 8: cost=1.298, accuracy=0.422/0.330 (7/29 secs)
 Epoch 10: cost=1.276, accuracy=0.439/0.450 (7/36 secs)
 Model flowers_model_1 train ended in 36 secs:
 Model flowers_model_1 test report: accuracy = 0.397, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [5,11,21,24,39] => 추정 tulip : 정답 tulip => 0
 추정확률분포 [20,21,19,19,21] => 추정 tulip : 정답 sunflower => X
 추정확률분포 [20,21,19,19,21] => 추정 tulip : 정답 dandelion => X

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

Model flowers_model_1 train started:
 Epoch 10: cost=1.312, accuracy=0.424/0.370 (36/36 secs)
 Epoch 20: cost=1.190, accuracy=0.474/0.380 (35/71 secs)
 Epoch 30: cost=1.166, accuracy=0.483/0.340 (36/107 secs)
 Model flowers_model_1 train ended in 107 secs:
 Model flowers_model_1 test report: accuracy = 0.422, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [14,23,15,26,22] => 추정 sunflower : 정답 sunflower => 0
 추정확률분포 [6, 6,37, 3,47] => 추정 tulip : 정답 rose => X
 추정확률분포 [95, 5, 0, 0, 0] => 추정 daisy : 정답 dandelion => X

	<pre>fd = FlowersDataset() fm = MlpModel('flowers_model_1', fd, [10,30]) fm.exec_all(epoch_count=50, report=10)</pre> <p>Model flowers_model_1 train started: Epoch 10: cost=1.342, accuracy=0.395/0.410 (35/35 secs) Epoch 20: cost=1.246, accuracy=0.431/0.320 (36/71 secs) Epoch 30: cost=1.166, accuracy=0.475/0.450 (35/106 secs) Epoch 40: cost=1.110, accuracy=0.494/0.410 (36/142 secs) Epoch 50: cost=1.036, accuracy=0.519/0.390 (36/178 secs) Model flowers_model_1 train ended in 178 secs: Model flowers_model_1 test report: accuracy = 0.393, (0 secs)</p> <p>Model flowers_model_1 Visualization</p>  <p>추정확률분포 [25,22,34, 1,17] => 추정 rose : 정답 dandelion => X 추정확률분포 [1, 0,48, 0,51] => 추정 tulip : 정답 tulip => 0 추정확률분포 [10,70, 1,15, 5] => 추정 dandelion : 정답 dandelion => 0</p>
--	--

(2) 비교 대상 : 배치 크기

앞선 두 실험에서 높은 정확도를 보인 은닉 계층 [10,30] , 에포크 횟수 30회의 조건에서 실험을 진행하였다.

구분	내용		
조건	<ul style="list-style-type: none"> - 에포크 횟수(epoch_count) : 30 - 학습률(learning rate) : 0.001 - 은닉 계층 개수 : 2 - 은닉 계층 : [10,30] 		
	배치 크기(batch_size) : 10	배치 크기(batch_size) : 50	배치 크기(batch_size) : 100
데이터 정확도	45.4%	39.9%	41.9%
소요 시간	113s	65s	60s

결과 화면

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

Model flowers_model_1 train started:
 Epoch 10: cost=1.265, accuracy=0.436/0.370 (38/38 secs)
 Epoch 20: cost=1.155, accuracy=0.484/0.310 (37/75 secs)
 Epoch 30: cost=1.070, accuracy=0.529/0.490 (38/113 secs)
 Model flowers_model_1 train ended in 113 secs:
 Model flowers_model_1 test report: accuracy = 0.454, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [13,81, 5, 0, 0] => 추정 dandelion : 정답 dandelion => 0
 추정확률분포 [3,17,18,11,51] => 추정 tulip : 정답 tulip => 0
 추정확률분포 [14,15,25,13,33] => 추정 tulip : 정답 sunflower => X

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

Model flowers_model_1 train started:
 Epoch 10: cost=1.316, accuracy=0.411/0.450 (22/22 secs)
 Epoch 20: cost=1.244, accuracy=0.444/0.460 (21/43 secs)
 Epoch 30: cost=1.196, accuracy=0.466/0.460 (22/65 secs)
 Model flowers_model_1 train ended in 65 secs:
 Model flowers_model_1 test report: accuracy = 0.399, (0 secs)

Model flowers_model_1 Visualization



추정확률분포 [1, 5, 1,81,12] => 추정 sunflower : 정답 sunflower => 0
 추정확률분포 [8,28, 3,45,17] => 추정 sunflower : 정답 tulip => X
 추정확률분포 [5,11, 4,61,19] => 추정 sunflower : 정답 rose => X

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

Model flowers_model_1 train started:
 Epoch 10: cost=1.346, accuracy=0.400/0.400 (20/20 secs)
 Epoch 20: cost=1.272, accuracy=0.442/0.420 (20/40 secs)
 Epoch 30: cost=1.224, accuracy=0.453/0.470 (20/60 secs)
 Model flowers_model_1 train ended in 60 secs:
 Model flowers_model_1 test report: accuracy = 0.419, (0 secs)



Model flowers_model_1 Visualization




추정확률분포 [0, 0,60, 0,39] => 추정 rose : 정답 rose => 0
 추정확률분포 [3,36, 0,53, 8] => 추정 sunflower : 정답 dandelion => X
 추정확률분포 [29,57, 7, 2, 5] => 추정 dandelion : 정답 daisy => X

(3) 비교 대상 : 학습률

앞선 두 실험에서 높은 성능을 보인 은닉 계층 [10,30] , 에포크 횟수 30회, 미니배치 크기 50의 조건에서 실험을 진행하였다.

구분	내용		
조건	- 에포크 횟수(epoch_count) : 30 - 미니배치 크기(mb_size) : 50 - 은닉 계층 개수 : 2 - 은닉 계층 : [10,30]		
	학습률(learning rate) : 0.001	학습률(learning rate) : 0.01	학습률(learning rate) : 0.1
데이터 정확도	41.9%	25.2%	26.2%
소요시간	64s	65s	66s
결과 화면	<pre>fd = FlowersDataset() fm = MlpModel('flowers_model_1', fd, [10,30]) fm.exec_all(epoch_count=30, batch_size=50, learning_rate=0.001, report=10)</pre> <p>Model flowers_model_1 train started: Epoch 10: cost=1.280, accuracy=0.433/0.380 (21/21 secs) Epoch 20: cost=1.207, accuracy=0.466/0.400 (22/43 secs) Epoch 30: cost=1.145, accuracy=0.493/0.440 (21/64 secs) Model flowers_model_1 train ended in 64 secs: Model flowers_model_1 test report: accuracy = 0.419, (1 secs)</p> <p>Model flowers_model_1 Visualization</p>  <p>추정확률분포 [22,50, 5,14,10] => 추정 dandelion : 정답 daisy => X 추정확률분포 [6, 4,11, 7,72] => 추정 tulip : 정답 tulip => 0 추정확률분포 [16,11,30, 8,34] => 추정 tulip : 정답 rose => X</p> <pre>fd = FlowersDataset() fm = MlpModel('flowers_model_1', fd, [10,30]) fm.exec_all(epoch_count=30, batch_size=50, learning_rate=0.01, report=10)</pre> <p>Model flowers_model_1 train started: Epoch 10: cost=1.578, accuracy=0.260/0.230 (21/21 secs) Epoch 20: cost=1.594, accuracy=0.249/0.180 (22/43 secs) Epoch 30: cost=1.588, accuracy=0.251/0.260 (22/65 secs) Model flowers_model_1 train ended in 65 secs: Model flowers_model_1 test report: accuracy = 0.252, (0 secs)</p> <p>Model flowers_model_1 Visualization</p>  <p>추정확률분포 [17,24,18,18,22] => 추정 dandelion : 정답 sunflower => X 추정확률분포 [17,24,18,18,22] => 추정 dandelion : 정답 dandelion => 0 추정확률분포 [17,24,18,18,22] => 추정 dandelion : 정답 sunflower => X</p>		

	<pre>fd = FlowersDataset() fm = MlpModel('flowers_model_1', fd, [10,30]) fm.exec_all(epoch_count=30, batch_size=50, learning_rate=0.1, report=10)</pre> <p>Model flowers_model_1 train started: Epoch 10: cost=1.600, accuracy=0.241/0.260 (22/22 secs) Epoch 20: cost=1.600, accuracy=0.241/0.250 (22/44 secs) Epoch 30: cost=1.600, accuracy=0.233/0.300 (22/66 secs) Model flowers_model_1 train ended in 66 secs: Model flowers_model_1 test report: accuracy = 0.262, (0 secs)</p> <p>Model flowers_model_1 Visualization</p>  <p>추정확률분포 [18,24,18,17,23] => 추정 dandelion : 정답 daisy => X 추정확률분포 [18,24,18,17,23] => 추정 dandelion : 정답 tulip => X 추정확률분포 [18,24,18,17,23] => 추정 dandelion : 정답 dandelion => 0</p>
--	---

2.4 하이퍼볼릭 탄젠트 함수 이용


```
def mlp_forward_layer(self, x, hconfig, pm):
    y = np.matmul(x, pm['w']) + pm['b']
    #if hconfig is not None: y = relu(y)
    if hconfig is not None: y = np.tanh(y)
    return y, [x,y]
def mlp_backprop_layer(self, G_y, hconfig, pm, aux):
    x, y = aux

    #if hconfig is not None: G_y = relu_derv(y) * G_y
    if hconfig is not None: G_y = tanh_derv(y) * G_y
def tanh_derv(x):
    return (1+np.tanh(x))*(1-np.tanh(x))
```


위와 같이 relu 함수를 Hyperbolic Tangent 함수로 변경하여 실험을 진행해보았다. 실험 조건은 앞선 실험의 좋은 결과를 보인 조건을 동일하게 적용하였다.

(1) relu 함수 이용

구분	내용
조건	<ul style="list-style-type: none"> - 에포크 횟수(epoch_count) : 30 - 미니배치 크기(mb_size) : 50 - 학습률(learning rate) : 0.001 - 은닉 계층 개수 : 2 - 은닉 계층 : [10,30]
데이터 정확도	43.4%
소요 시간	67s

결과 화면	<pre>fd = FlowersDataset() fm = MlpModel('flowers_model_1', fd, [10,30]) fm.exec_all(epoch_count=30, batch_size=50, learning_rate=0.001, report=10)</pre> <p>Model flowers_model_1 train started: Epoch 10: cost=1.383, accuracy=0.385/0.350 (22/22 secs) Epoch 20: cost=1.248, accuracy=0.445/0.400 (22/44 secs) Epoch 30: cost=1.192, accuracy=0.468/0.430 (23/67 secs) Model flowers_model_1 train ended in 67 secs: Model flowers_model_1 test report: accuracy = 0.434, (0 secs)</p> <p>Model flowers_model_1 Visualization</p>  <p>추정확률분포 [3, 8,22,11,56] => 추정 tulip : 정답 tulip => 0 추정확률분포 [7,62, 0,21, 9] => 추정 dandelion : 정답 dandelion => 0 추정확률분포 [13,63, 0, 5,18] => 추정 dandelion : 정답 dandelion => 0</p>
-------	---

(2) hyperbolic tangent 함수 이용

구분	내용
조건	<ul style="list-style-type: none"> - 에포크 횟수(epoch_count) : 30 - 미니배치 크기(mb_size) : 50 - 학습률(learning rate) : 0.001 - 은닉 계층 개수 : 2 - 은닉 계층 : [10,30]
데이터 정확도	40.5%
소요 시간	68s
결과 화면	<pre>fd = FlowersDataset() fm = MlpModel('flowers_model_1', fd, [10,30]) fm.exec_all(epoch_count=30, batch_size=50, learning_rate=0.001, report=10)</pre> <p>Model flowers_model_1 train started: Epoch 10: cost=1.303, accuracy=0.419/0.420 (22/22 secs) Epoch 20: cost=1.205, accuracy=0.469/0.370 (23/45 secs) Epoch 30: cost=1.143, accuracy=0.502/0.380 (23/68 secs) Model flowers_model_1 train ended in 68 secs: Model flowers_model_1 test report: accuracy = 0.405, (0 secs)</p> <p>Model flowers_model_1 Visualization</p>  <p>추정확률분포 [0, 0,65, 0,35] => 추정 rose : 정답 rose => 0 추정확률분포 [1, 6,47, 1,45] => 추정 rose : 정답 tulip => X 추정확률분포 [25,70, 3, 0, 1] => 추정 dandelion : 정답 daisy => X</p>

3.결론

실험 결과를 표와 함께 정리하면 다음과 같다.

(1)은닉 계층 개수에 따른 비교

은닉 계층 개수	0	1	2	3
정확도(%)	34.8	24.7	38.5	41.4
소요 시간(s)	30	33	33	34

은닉 계층 개수를 달리하며 여러번 실험해보았을 때, 평균적으로 30-40% 사이의 정확도를 보였다. 하지만 은닉 계층의 개수가 1개일 때는 정확도가 25%로 낮지만, 계층 수가 증가할 수록 정확도가 더 높아지는 모습을 볼 수 있었다. 은닉계층이 0개인 다층 퍼셉트론은 단층퍼셉트론과 같기에, 다층 퍼셉트론에 있어서 은닉 계층이 증가할수록 정확도가 상승한다고 유추해 볼 수 있다. 이후 은닉계층 폭에 따른 비교 실험에서는 계층간의 폭을 효과적으로 보여줄 수 있는 은닉 계층 개수 2개의 조건을 가지고 실험을 수행하였다.

(2)은닉 계층 폭에 따른 비교

은닉 계층간 폭	동일			증가			감소		
은닉 계층	[4,4]	[10,10]	[30,30]	[10,20]	[10,30]	[10,50]	[20,10]	[30,10]	[50,10]
증가 폭	0	0	0	+10	+20	+40	-10	-20	-40
소요 시간(s)	25	34	45	31	36	35	43	48	61
정확도(%)	38.0	36.7	37.9	38.3	42.8	29.4	39.6	39.1	30.1

실험 결과, 은닉 계층 2개의 폭이 동일할 때는 36.5-38.0%로 정확도가 비슷하게 나타났다. 따라서 폭을 달리해 실험을 진행해보았는데, 계층 간의 폭이 너무 커질 경우 정확도가 오히려 낮아지는 모습을 볼 수 있었다. 이를 통해 은닉 계층의 경우 증가폭이 커질수록 정확도가 증가할 수는 있으나, 계층 간 차이가 너무 커지면 오히려 성능 저하를 야기함을 유추할 수 있었다. 은닉 계층을 감소하도록 설정했을 때에도, 두 계층 간의 차이가 크면 오히려 정확도가 감소하는 모습을 확인할 수 있었다.

이어서, 하이퍼파라미터를 조절해 정확도 비교 실험을 진행하였다.

(3)하이퍼 파라미터 튜닝 실험

하이퍼파라미터	에포크 횟수			배치 크기			학습률		
	10	30	50	10	50	100	0.001	0.01	0.1
소요 시간(s)	36	107	178	113	65	60	64	65	66
정확도(%)	38.0	36.7	37.9	45.4	39.9	41.9	41.9	25.2	26.2

에포크 횟수, 배치 크기, 학습률 순으로 비교 실험을 진행하였기에, 배치 크기 및 학습률 실험에

서는 에포크 횟수 실험에서 결과가 준수했던 에포크 횟수 30회 조건을 이용하였다. 또한, 학습률 실험에서는 배치 크기 실험의 결과가 좋았던 배치크기 50의 조건을 이용하여 실험하였다.

그 결과, 에포크 횟수 실험에서는 정확도가 횟수에 상관없이 비슷한 양상을 보였으나, 소요 시간에서 큰 차이를 보였다. 이를 통해 에포크 횟수가 적을 때 성능이 더 좋다고 판단하였다. 이후 실험에서는 에포크 횟수가 너무 적지 않은 30회를 기준으로 하여 작업을 진행하였는데, 배치 크기를 다르게 했을때에도 마찬가지로 정확도보다는 소요 시간에서 큰 차이를 보였다. 정확도의 경우 모두 40-45% 정도로 나타났지만 배치크기가 10인 경우는 113s의 많은 소요시간이 필요함을 알 수 있었다. 크기가 50일 때와 100일때의 정확도 및 소요 시간이 비슷하다고 판단하여 이후 실험에는 배치크기 50의 조건을 이용하였다. 학습률의 경우는 정확도에서 많은 차이가 나타났다. 학습률이 0.001인 경우 41.9%의 정확도를 보인 것과 다르게 0.01, 0.1 경우에는 25%정도의 정확도가 나타났다. 이를 통해 학습률이 0.001인 조건이 성능 개선에 적합하다는 판단을 하였다.

(6)Relu 함수와 Hyperbolic Tangent 함수 사용 시 비교

	relu 함수	tanh 함수
소요시간(s)	67	68
정확도(%)	43.4	40.5

Hyperbolic Tangent 함수는 실수 값을 (-1, 1) 범위로 압축하는 활성화 함수로 일종의 Sigmoid 함수라 할 수 있다. 이 함수는 원점을 중심으로 값이 양의 무한대로 갈수록 1에 수렴하고, 음의 무한대로 갈수록 -1에 수렴하기에, 비선형성을 모델에 부여할 수 있다는 특징이 있다.

Relu 함수를 Hyperbolic Tangent 함수로 변경해 본 결과, 정확도 면에서 소폭 하락한 모습을 보였으나, 소요시간 및 정확도에서 비슷한 결과를 나타냈다고 판단할 수 있다. 이를 통해 다층 퍼셉트론 문제를 해결하는데 있어 relu함수를 hyperbolic tangent 함수로 대체하여 사용할 수도 있다는 점을 알 수 있었다.

결론적으로, 위와 같은 실험을 통해 에포크 횟수 30, 미니배치 크기 50, 학습률 0.001, 은닉 계층 [10,30], relu함수 사용의 조건에서 43.4%의 정확도를 얻으며 성능 개선이 가능했다. 추가로 은닉 계층이나 하이퍼파라미터를 더욱 미세하게 조절함으로써 정확도를 더 높일 수 있을 것으로 기대된다.

3-1. 추가 분석

다층 퍼셉트론 실험 결과, 여러 조건을 조절해보아도 정확도가 25-45%에 불과한 모습을 보였다. 이를 통해 다층 퍼셉트론은 이미지 처리에 부적합하다는 것을 알 수 있다. 이는 다층 퍼셉트론이 작은 변화에도 입력층의 변화가 너무 크기 때문이라고 유추해볼 수 있다. 다층 퍼셉트론은 픽셀과 퍼셉트론을 일대일로 대응시켜 이미지를 인식하는데, 이미지내의 물체의 크기가 달라지거나 각도가 바뀔 경우에도 입력층의 노드 값이 바뀐다. 따라서 다층 퍼셉트론은 이미지의 위치, 크

기, 각도가 바뀌었을 때 이미지를 제대로 인식하지 못할 확률이 높다.

그렇다고 해서 파라미터를 너무 많이 사용하여도 좋지 않다. 왜냐하면 튜닝 가능한 파라미터의 수가 많을 경우, 모델이 오버피팅(overfitting)에 더 취약한 경향이 있기 때문이다. 오버피팅은 학습 오류가 테스트 데이터셋에 대한 오류보다 아주 작은 경우를 말하는데, 오버피팅 발생 시 성능에 대한 올바른 평가를 할 수가 없다.

추가로, 오버피팅을 방지하기 위해 데이터를 보통 학습데이터, 검증데이터, 평가데이터로 구분하여 사용하게 된다. 만약 주어진 데이터에만 치중하여 학습할 경우, 조금이라도 다른 패턴을 가진 데이터에 대해 모델의 성능이 떨어지므로 검증 데이터를 이용하고, 학습데이터와 다른 평가데이터를 사용하며 모델의 성능이 잘 나오는지 확인할 수 있다.