

5장. 다층 퍼셉트론 모델 구조: 꽃 이미지 분류 신경망

부산대학교 항공우주공학과

정대현 201527137

1. 개요

이번 5장에서는 다층 퍼셉트론을 사용하여 꽃 이미지를 분류하게 됩니다. 이번 실험의 목적은 우수한 성능을 가진 이미지 분류 모델을 만드는 것 보다 합성곱신경망(CNN, Convolutional Neural Network) 모델 평가를 위한 베이스라인을 만드는 성격을 가지고 있습니다. 이번 실험을 통해 다층 퍼셉트론은 이미지 분류에 얼마나 효율적인지 확인하며 그렇지 않다면 그 이유를 찾아보게 됩니다. 모델은 데이터 학습을 할 때 가지고 있는 모든 데이터를 한 번에 학습하는 것이 아닌 데이터를 특정 목적에 따라서 훈련(train) - 검증(valid) - 평가(test) 3가지로 나누게 됩니다. 데이터를 나누는 이유를 알아보며, 용도에 따라 어떤 데이터는 중복 사용이 가능하지만 특정 데이터는 철저하게 분리되어야 하는 이유를 학습합니다.

2. 다층 퍼셉트론의 이미지 처리

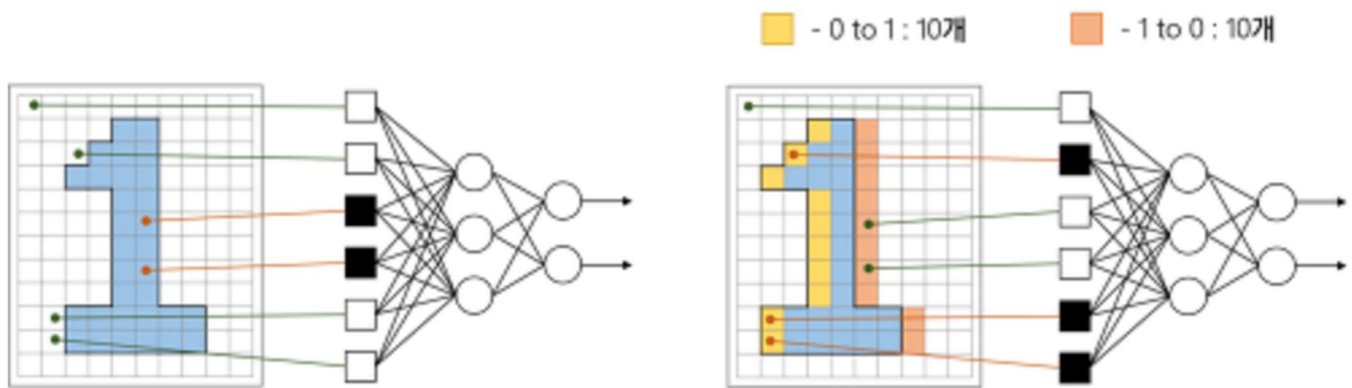


Fig 1. 다층 퍼셉트론의 이미지 처리

위 사진의 경우 1이라는 형태의 이미지는 $N \times N$ 크기의 픽셀로 구성되어 있습니다. 전통적인 머신러닝 기법과 다층 퍼셉트론을 가지고 위 사진을 학습한다면 이미지의 모든 픽셀을 각각의 변수로 설정하고 투명도와 색깔을 기준으로 가중치를 부여합니다. 이를 가지고 기계는 최적의 가중치 값을 찾아내도록 학습합니다. 만일 '1'이라는 이미지가 왼쪽으로 한 칸 이동한다고 가정하면 10개의 픽셀의 값이 0에서 1로 바뀌고 10개의 픽셀의 값이 1에서 0으로 바뀝니다. 또한, '1'이 크기가 달라지거나 각도가 바뀔 경우에도 입력층의 노드 값이 바뀝니다. 우리가 보기에 같은 숫자 '1'이지만, 기계가 받아들이는 입력층의 변화는 굉장히 큼니다. 따라서, 다층 퍼셉트론은 이미지의 위치, 크기 또는 각도가 바뀌었을 때 전혀 다른 이미지로 판단 할 수 있습니다. 이를 극복하기 위해 합성곱신경망을 사용하게 되며 이는 추후에 학습합니다.

3. 데이터의 구분



Fig 2. 데이터의 분류

데이터를 가지고 있다면 이를 모두 학습에 사용하는 것이 아닌 학습-검증-평가로 나누어서 사용하게 됩니다. 만약 모델이 평가 데이터를 가지고 학습하게 된다면 마치 학생이 답안지를 보고 시험을 본 것과 같다고 할 수 있습니다. 당장 시험은 잘 보았다고 할 수 있지만 기존 시험 대신 조금이라도 다른 문제를 만나면 결과는 형편 없을 것으로 예상합니다. 학습모델에서 기존 학습한 데이터의 경우 우수한 정확도를 보이거나 새로운 데이터에 대해서는 잘 작동하지 않을 수 있습니다. 이렇게 특정 데이터에 대해서만 우수한 성능을 보이는 오버피팅(overfitting) 과적합이라 합니다. 이를 해결하기 위해서 모델은 학습 데이터 만을 가지고 학습을 하며 검증 데이터를 통해서 과적합이 일어나지 않도록 일반화 문제를 조정하며 평가 데이터로 모델 성능을 평가하게 됩니다. 즉 평가 데이터는 모델을 학습하는데 전혀 사용하지 않은 데이터로 구성됩니다. 실제 모델을 만들 때 평가 데이터가 학습데이터에 섞이거나 여러가지 이유로 오염되는 것을 데이터누수(data leakage)라고 하며 이를 방지하기 위해 많은 노력을 하고 있습니다.

4. 실험

3.1 은닉계층 1개 폭 [10] epoch 10회

```
Epoch 2: cost=1.606, accuracy=0.247/0.300 (3/3 secs)
Epoch 4: cost=1.604, accuracy=0.247/0.180 (2/5 secs)
Epoch 6: cost=1.602, accuracy=0.247/0.310 (3/8 secs)
Epoch 8: cost=1.601, accuracy=0.247/0.230 (2/10 secs)
Epoch 10: cost=1.600, accuracy=0.247/0.250 (3/13 secs)
Model flowers_model_1 train ended in 13 secs:
Model flowers_model_1 test report: accuracy = 0.232, (0 secs)
추정확률분포 [19,22,19,18,21] => 추정 dandelion : 정답 daisy => X
추정확률분포 [19,22,19,18,21] => 추정 dandelion : 정답 daisy => X
추정확률분포 [19,22,19,18,21] => 추정 dandelion : 정답 rose => X
```

3.2 은닉계층 2개 폭 [30, 10] epoch 10회

```
Epoch 2: cost=1.412, accuracy=0.377/0.320 (4/4 secs)
Epoch 4: cost=1.337, accuracy=0.415/0.380 (5/9 secs)
Epoch 6: cost=1.278, accuracy=0.446/0.470 (4/13 secs)
Epoch 8: cost=1.239, accuracy=0.452/0.410 (5/18 secs)
Epoch 10: cost=1.198, accuracy=0.483/0.360 (4/22 secs)
Model flowers_model_2 train ended in 22 secs:
Model flowers_model_2 test report: accuracy = 0.394, (0 secs)
추정확률분포 [24,20,20,14,22] => 추정 daisy : 정답 daisy => O
추정확률분포 [ 8,15,23,12,43] => 추정 tulip : 정답 dandelion => X
추정확률분포 [22,40, 9,14,16] => 추정 dandelion : 정답 dandelion => O
```

3.3 은닉계층 3개 폭 [50, 25, 10] epoch 10회

```
Epoch 2: cost=1.373, accuracy=0.390/0.500 (6/6 secs)
Epoch 4: cost=1.250, accuracy=0.454/0.400 (7/13 secs)
Epoch 6: cost=1.188, accuracy=0.479/0.370 (6/19 secs)
Epoch 8: cost=1.144, accuracy=0.516/0.460 (6/25 secs)
Epoch 10: cost=1.096, accuracy=0.543/0.370 (7/32 secs)
Model flowers_model_3 train ended in 32 secs:
Model flowers_model_3 test report: accuracy = 0.472, (0 secs)
추정확률분포 [ 7, 8,42, 6,37] => 추정 rose : 정답 tulip => X
추정확률분포 [12,43, 9,25,11] => 추정 dandelion : 정답 tulip => X
추정확률분포 [12, 8, 9,25,47] => 추정 tulip : 정답 tulip => O
```

3.4 은닉계층 3개 폭 [50, 25, 10] epoch 25회

```
Epoch 5: cost=1.239, accuracy=0.453/0.470 (16/16 secs)
Epoch 10: cost=1.115, accuracy=0.527/0.450 (16/32 secs)
Epoch 15: cost=1.002, accuracy=0.595/0.460 (17/49 secs)
Epoch 20: cost=0.880, accuracy=0.651/0.430 (18/67 secs)
Epoch 25: cost=0.771, accuracy=0.704/0.380 (17/84 secs)
Model flowers_model_3 train ended in 84 secs:
Model flowers_model_3 test report: accuracy = 0.419, (0 secs)
추정확률분포 [ 2,14, 1, 0,82] => 추정 tulip : 정답 dandelion => X
추정확률분포 [ 0, 0, 1, 0,99] => 추정 tulip : 정답 tulip => O
추정확률분포 [25,57, 9, 1, 8] => 추정 dandelion : 정답 dandelion => O
```

3.5 은닉계층 3개 폭 [50, 25, 10] epoch 50회

Epoch 10: cost=1.073, accuracy=0.537/0.480 (6/32 secs)
Epoch 20: cost=0.832, accuracy=0.670/0.520 (6/63 secs)
Epoch 30: cost=0.645, accuracy=0.752/0.500 (6/95 secs)
Epoch 40: cost=0.482, accuracy=0.817/0.500 (7/129 secs)
Epoch 50: cost=0.381, accuracy=0.857/0.450 (7/162 secs)
Model flowers_model_3 train ended in 162 secs:
Model flowers_model_3 test report: accuracy = 0.446, (0 secs)
추정확률분포 [0, 0.92, 0, 8] => 추정 rose : 정답 rose => O
추정확률분포 [0, 0, 0, 0.100] => 추정 tulip : 정답 tulip => O
추정확률분포 [45,34, 6, 6, 9] => 추정 daisy : 정답 rose => X

3.6 은닉계층 3개 폭 [100, 50, 25] epoch 10회

Epoch 2: cost=1.346, accuracy=0.422/0.400 (10/10 secs)
Epoch 4: cost=1.233, accuracy=0.461/0.480 (11/21 secs)
Epoch 6: cost=1.145, accuracy=0.518/0.500 (10/31 secs)
Epoch 8: cost=1.065, accuracy=0.558/0.520 (11/42 secs)
Epoch 10: cost=0.998, accuracy=0.589/0.420 (10/52 secs)
Model flowers_model_3 train ended in 52 secs:
Model flowers_model_3 test report: accuracy = 0.440, (1 secs)
추정확률분포 [3, 4.72, 1.21] => 추정 rose : 정답 dandelion => X
추정확률분포 [23,40,18, 5,14] => 추정 dandelion : 정답 dandelion => O
추정확률분포 [2, 1.56, 0.40] => 추정 rose : 정답 tulip => X

3.7 은닉계층 3개 폭 [100, 50, 25] epoch 50회

Epoch 10: cost=1.009, accuracy=0.588/0.440 (54/54 secs)
Epoch 20: cost=0.713, accuracy=0.708/0.360 (54/108 secs)
Epoch 30: cost=0.459, accuracy=0.822/0.470 (57/165 secs)
Epoch 40: cost=0.350, accuracy=0.869/0.420 (59/224 secs)
Epoch 50: cost=0.284, accuracy=0.899/0.430 (62/286 secs)
Model flowers_model_3 train ended in 286 secs:
Model flowers_model_3 test report: accuracy = 0.442, (0 secs)
추정확률분포 [1.99, 0, 0, 0] => 추정 dandelion : 정답 sunflower => X
추정확률분포 [100, 0, 0, 0, 0] => 추정 daisy : 정답 daisy => O
추정확률분포 [19,42,33, 5, 1] => 추정 dandelion : 정답 daisy => X

3.8 은닉계층 4개 폭 [100, 50, 25, 10] epoch 10회

```
Epoch 2: cost=1.516, accuracy=0.331/0.350 (10/10 secs)
Epoch 4: cost=1.320, accuracy=0.425/0.320 (11/21 secs)
Epoch 6: cost=1.250, accuracy=0.460/0.450 (11/32 secs)
Epoch 8: cost=1.196, accuracy=0.482/0.430 (11/43 secs)
Epoch 10: cost=1.154, accuracy=0.514/0.470 (11/54 secs)
Model flowers_model_3 train ended in 54 secs:
Model flowers_model_3 test report: accuracy = 0.423, (0 secs)
추정확률분포 [21,14,10,39,16] => 추정 sunflower : 정답 sunflower => O
추정확률분포 [ 6, 6,42, 2,45] => 추정 tulip : 정답 rose => X
추정확률분포 [34,51, 6, 3, 6] => 추정 dandelion : 정답 dandelion => O
```

실험	은닉층	epoch	시간	정확도
1	[10]	10	13s	0.232
2	[30, 10]	10	22s	0.394
3	[50, 25, 10]	10	32s	0.472
4	[50, 25, 10]	25	84s	0.419
5	[50, 25, 10]	50	162s	0.446
6	[100, 50, 25]	10	52s	0.440
7	[100, 50, 25]	50	286s	0.442
8	[100, 50, 25, 10]	10	55s	0.423

5. 결론

다층 퍼셉트론 신경망은 CNN 보다 이미지 처리에 부적합한 것이 일반적인 사실입니다. 썸네일 사진의 크기를 160x120픽셀이라고 할 경우 일반적으로 사용하는 사진에 비해서 훨씬 저해상도의 사진이지만 이미지의 구조는 [160, 120, 3] 형태로 픽셀 57,600개를 가지게 됩니다. 따라서 사진 한 장의 입력 벡터 크기는 57,600이며 출력 벡터 크기는 분류 대상 꽃의 가지수인 5입니다. 정보 압축면에서 은닉 벡터는 입력벡터 크기와 출력 벡터 크기의 사이에서 몇 단계에 걸쳐 줄여주는 것이 좋습니다. 이를 고려하여 은닉계층의 폭을 [5000, 500, 50]이라고 가정할 경우 출력 계층까지 고려한 모델 구성에 필요한 파라미터 수는 약 3억개에 이르게 됩니다. 이와 같이 엄청난 파라미터를 학습하기 위해서는 데이터 셋도 많이 요구되나 이번 실험에서의 사진 개수는 총 4,242장으로 파라미터를 모두 학습시키기에는 부족합니다. 설정 파라미터를 충분히 학습할 데이터를 확보했다고 하더라도 많은 파라미터와 많은 데이터를 처리하기에는 엄청난 컴퓨터 자원과 시간을 요구하게 됩니다.

이번 실험에서는 모델을 학습하기 위한 하이퍼파라미터로 은닉층 형태와 epoch를 변경하면서 실험했습니다. 은닉층이 1개일 때 보다는 2개 이상 많아지면 정확도는 개선되는 모습을 보였습니다. 하지만 실험 3의 정확도의 0.472를 보였지만 이후의 실험 4~8번을 비교해보면 실험 3에 비해서 epoch 횟수를 증가하거나, 은닉층을 추가하고 폭을 깊게 하여도 정확도는 증가하지 못하고 소폭 감소하였습니다. 파라미터 고려하였을 때 충분히 많은 데이터를 확보하지 못하고 이미지 처리에 부적합한 다층 퍼셉트론으로는 높은 정확도를 기대할 수 없었습니다.