

# 인공지능 챕터6, 7 과제

202055518 김병현

## 챕터6

은닉 계층 하나를 갖는 다층 퍼셉트론 신경망 학습

```
om1 = AdamModel('office31_model_1', od, [10])  
om1.exec_all(epoch_count=20, report=10)
```

Model office31\_model\_1 Visualization



- 일정한 추정 확률 분포에서 벗어나지 못하는 것을 확인할 수 있음

[ 도메인 추정결과 ]

추정확률분포 [63,16,21] => 추정 amazon : 정답 amazon => 0

추정확률분포 [63,16,21] => 추정 amazon : 정답 dslr => X

추정확률분포 [63,16,21] => 추정 amazon : 정답 amazon => 0

[ 상품 추정결과 ]

추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3, 3,...] => 추정 mobile\_phone : 정답 paper\_notebook => X

추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3, 3,...] => 추정 mobile\_phone : 정답 keyboard => X

추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3, 3,...] => 추정 mobile\_phone : 정답 monitor => X

은닉 계층 세 개와 학습률을 조정한 다층 퍼셉트론 신경망 학습

```
om2 = AdamModel('office31_model_2', od, [64,32,10])  
om2.exec_all(epoch_count=50, report=10, learning_rate=0.0001)
```

- 도메인과 품목 모두에서 정확도 향상

[ 도메인 추정결과 ]

추정확률분포 [ 4,23,73] => 추정 webcam : 정답 webcam => 0

추정확률분포 [22,49,29] => 추정 dslr : 정답 amazon => X

추정확률분포 [96, 1, 3] => 추정 amazon : 정답 amazon => 0

[ 상품 추정결과 ]

추정확률분포 [ 0,30, 3, 1, 1, 2, 1, 6,...] => 추정 bike : 정답 stapler => X

추정확률분포 [45, 0, 0, 8, 1,10, 0, 9,...] => 추정 back\_pack : 정답 paper\_notebook => X

추정확률분포 [ 1, 0, 1, 6, 2, 5, 4, 1,...] => 추정 ring\_binder : 정답 ring\_binder => 0

- 추정 확률 분포의 데이터별 차별화 확인 가능

## adam 알고리즘을 이용한 신경망 학습

- use\_adam 플래그 설정으로 adam 알고리즘 활성화

```
om3 = AdamModel('office31_model_3', od, [64,32,10])  
om2.use_adam = True  
om3.exec_all(epoch_count=50, report=10, learning_rate=0.0001)
```

- 전체적인 정확도 약간 향상
  - adam 알고리즘을 이용한다고 항상 성능이 향상되는 것은 아님
  - 하지만 대개의 경우 성능 향상, 크게 향상되는 경우도 많음



### [ 도메인 추정결과 ]

추정확률분포 [99, 0, 1] => 추정 amazon : 정답 amazon => 0

추정확률분포 [ 0,23,77] => 추정 webcam : 정답 ds1r => X

추정확률분포 [99, 0, 1] => 추정 amazon : 정답 amazon => 0

### [ 상품 추정결과 ]

추정확률분포 [ 0, 1, 1, 1, 1, 0, 3, 5,...] => 추정 ruler : 정답 ruler => 0

추정확률분포 [ 1, 0, 2, 0, 0, 0, 1, 0,...] => 추정 monitor : 정답 monitor => 0

추정확률분포 [ 0, 3, 0, 1, 1, 0, 2, 7,...] => 추정 ruler : 정답 ruler => 0

## 챕터7

### cnn\_basic\_test

확장된 모델로 다층 퍼셉트론 신경망 만들어보기

```
fm1 = CnnBasicModel('flowers_model_1', fd, [30, 10])
fm1.exec_all(epoch_count = 10, report = 2)
```

- 은닉 계층 구성 정보 [30, 10]
- 폭 30의 은닉 계층 -> 폭 10의 은닉 계층 -> (폭 5의 출력 계층)
- 일정한 추정 확률 분포에서 벗어나지 못하는 것을 볼 수 있음

추정확률분포 [30,40,14, 4,13] => 추정 dandelion : 정답 daisy => X

추정확률분포 [12,20,20,20,28] => 추정 tulip : 정답 sunflower => X

추정확률분포 [ 1, 1,56, 0,42] => 추정 rose : 정답 tulip => X

Model flowers\_model\_1 train ended in 169 secs:

Model flowers\_model\_1 test report: accuracy = 0.363, (0 secs)

완전 연결 계층 구성 정보의 명시적 표현과 **adam** 알고리즘 해제

```
fm2.use_adam = False
fm2.exec_all(epoch_count = 10, report = 2)
```

- **adam** 알고리즘 해제로 오히려 좋은 결과: 매우 예외적 상황임

추정확률분포 [ 0, 0,68, 0,31] => 추정 rose : 정답 rose => 0

추정확률분포 [ 8,23,14,30,25] => 추정 sunflower : 정답 dandelion => X

추정확률분포 [18,16,29,11,27] => 추정 rose : 정답 sunflower => X

Model flowers\_model\_2 train ended in 48 secs:

Model flowers\_model\_2 test report: accuracy = 0.396, (0 secs)

간단한 합성곱 신경망을 이용한 꽃 이미지 분류 신경망

```
fm3 = CnnBasicModel('flowers_model_3', fd,
                    [['conv', {'ksize':5, 'chn':6}],
                     ['max', {'stride':4}],
                     ['conv', {'ksize':3, 'chn':12}],
                     ['avg', {'stride':2}]],
                    True)
fm3.exec_all(epoch_count = 10, report = 2)
```

- 크게 줄어든 파라미터 개수에도 불구하고 학습시간 크게 증가
- 55.4%의 정확도로 다층 퍼셉트론의 39% 수준을 능가

Model flowers\_model\_3 train ended in 850 secs:

Model flowers\_model\_3 test report: accuracy = 0.554, (17 secs)

추정확률분포 [ 6, 4, 2, 0,88] => 추정 tulip : 정답 tulip => 0

추정확률분포 [ 7,49, 2,19,23] => 추정 dandelion : 정답 dandelion => 0

추정확률분포 [ 4, 1,80, 3,11] => 추정 rose : 정답 daisy => X

계층을 늘린 합성곱 신경망과 꽃 이미지 분류

```
fm4 = CnnBasicModel('flowers_model_4', fd,
                    [['conv', {'ksize':3, 'chn':6}],
                     ['max', {'stride':2}],
                     ['conv', {'ksize':3, 'chn':12}],
                     ['max', {'stride':2}],
                     ['conv', {'ksize':3, 'chn':24}],
                     ['avg', {'stride':3}]])
fm4.exec_all(epoch_count = 10, report = 2)
```

Model flowers\_model\_4 train ended in 751 secs:

Model flowers\_model\_4 test report: accuracy = 0.661, (4 secs)

- 66.1%로 정확도 향상

같은 신경망 이용한 오피스31 다차원 분류 신경망

```
om1 = CnnBasicModel('office31_model_1', od,
                    [['conv', {'ksize':3, 'chn':6}],
                     ['max', {'stride':2}],
                     ['conv', {'ksize':3, 'chn':12}],
                     ['max', {'stride':2}],
                     ['conv', {'ksize':3, 'chn':24}],
                     ['avg', {'stride':3}]])
om1.exec_all(epoch_count = 10, report = 2)
```

- 도메인 정확도 87.5% 상품 정확도 53.9%

Model office31\_model\_1 test report: accuracy = 0.875+0.539, (5 secs)

[ 도메인 추정결과 ]

추정확률분포 [61, 1,38] => 추정 amazon : 정답 amazon => 0

추정확률분포 [100, 0, 0] => 추정 amazon : 정답 amazon => 0

추정확률분포 [100, 0, 0] => 추정 amazon : 정답 amazon => 0

[ 상품 추정결과 ]

추정확률분포 [ 0, 0, 0, 0, 0, 0, 0, 0,...] => 추정 pen : 정답 scissors => X

추정확률분포 [ 0, 0, 0, 0, 0, 0, 0, 0,...] => 추정 laptop\_computer : 정답 phone => X

추정확률분포 [ 0, 0, 0, 1, 0, 0, 0, 0,...] => 추정 monitor : 정답 paper\_notebook => X

오피스 31 문제 위한 합성곱 신경망의 학습 횟수 늘려보기

```
om1.exec_all(epoch_count = 40, show_cnt = 0)
```

Model office31\_model\_1 test report: accuracy = 0.880+0.515, (5 secs)

- 성능 향상 없음

### 시그모이드 함수를 비선형 활성화 함수로 이용해보기

```
om2 = CnnBasicModel('office31_model_2', od,
                    [['conv', {'ksize':3, 'chn':6, 'actfunc':'sigmoid'}],
                     ['max', {'stride':2}],
                     ['conv', {'ksize':3, 'chn':12, 'actfunc':'sigmoid'}],
                     ['max', {'stride':2}],
                     ['conv', {'ksize':3, 'chn':24, 'actfunc':'sigmoid'}],
                     ['avg', {'stride':3}]])
om2.exec_all(epoch_count = 10, report = 2)
```

- **actfunc** 옵션 설정 통해 **relu** 대신 **sigmoid** 함수 이용
- 도메인 정확도: **69.3%** 상품 정확도: **3.8%** -> 학습 실패

Model office31\_model\_2 train ended in 1039 secs:

Model office31\_model\_2 test report: accuracy = 0.693+0.038, (6 secs)

### 쌍곡탄젠트 함수를 비선형 활성화 함수로 이용해보기

```
om3 = CnnBasicModel('office31_model_3', od,
                    [['conv', {'ksize':3, 'chn':6, 'actfunc':'tanh'}],
                     ['max', {'stride':2}],
                     ['conv', {'ksize':3, 'chn':12, 'actfunc':'tanh'}],
                     ['max', {'stride':2}],
                     ['conv', {'ksize':3, 'chn':24, 'actfunc':'tanh'}],
                     ['avg', {'stride':3}]])
om3.exec_all(epoch_count = 10, report = 2)
```

- **actfunc** 옵션 설정 통해 **relu** 대신 **tanh** 함수 이용
- 도메인 정확도: **77.6%** 상품 정확도: **36.6%** -> **relu** 보다 나쁜 결과

Model office31\_model\_3 train ended in 711 secs:

Model office31\_model\_3 test report: accuracy = 0.776+0.366, (5 secs)