

# 복합출력

부산대학교 정보컴퓨터공학부

15 학번 이연걸

## 1. 개요

딥러닝에서의 복합 출력 학습법에 대해서 알아본다. 데이터셋은 전이학습 연구용 표준 벤치마크 데이터셋인 오피스 31 데이터셋을 활용하며 이는 사무용품 이미지 4,652 장으로 구성되어 있다. 또한, 데이터별로 도메인 정보와 품목 정보가 함께 레이블링되어있다.

전이학습이란 한 도메인에서 학습시킨 결과를 다른 도메인에 활용하여 학습 효과를 높이는 학습 기법이다. 수집된 데이터에 정답을 표시하는 레이블링 작업에는 많은 경우 데이터 수집 이상의 더 많은 인력과 시간이 든다. 그렇다고 레이블링이 전혀 없이 신경망을 학습시킬 수 있는 방법은 마땅치 않다. 따라서 전이학습을 활용하면 새로운 도메인에서만이라도 다른 도메인에서 학습된 내용을 이용하여 레이블링 작업량을 줄여줄 수 있기 때문에 작업량을 줄일 필요가 있을 때 필요하다.

한편 실험에서는 다층 퍼셉트론 신경망을 사용하면서 은닉 계층과 학습률과 같은 파라미터 조정을 거치면서 차이를 분석한다.

그 후, 아담 알고리즘으로 학습을 진행한 뒤, 어떤 차이가 생기는데 대해서도 알아본다. 아담 알고리즘이란 기존의 경사하강법에서 손실 기울기 대신 모멘텀 정보를 이용해 보정된 값을 이용하여 local minimum 에 빠질 수 있는 경우를 global minimum 으로 벗어날 수 있도록 하는 알고리즘이다.

## 2. 구현 과정

다층 퍼셉트론을 사용한 경우와 아담 알고리즘 적용한 경우를 나누어 진행한다. 두 경우에서 AdamModel 을 사용했고, 아담 알고리즘 적용 여부는 use\_adam 을 True 로 바꾸었을 때이다.

## 2.1 다층 퍼셉트론

폭 [10]의 은닉 계층 하나만 갖는 간단한 다층 퍼셉트론 신경망과 폭 [64,32,10]의 셋을 갖는 신경망을 학습하고 비교한다. 그 후, 학습률과 학습 횟수에 변화를 준다.

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

Model office31\_model\_1 train started:  
Epoch 10: cost=4.263, accuracy=0.685+0.033/0.790+0.010 (39/39 secs)  
Epoch 20: cost=4.259, accuracy=0.685+0.033/0.780+0.020 (39/78 secs)  
Model office31\_model\_1 train ended in 78 secs:  
Model office31\_model\_1 test report: accuracy = 0.661+0.034, (1 secs)

Model office31\_model\_1 Visualization



[ 도메인 추정결과 ]

추정확률분포 [69,12,19] => 추정 amazon : 정답 amazon => 0

추정확률분포 [68,12,19] => 추정 amazon : 정답 amazon => 0

추정확률분포 [69,12,19] => 추정 amazon : 정답 dsir => X

[ 상품 추정결과 ]

추정확률분포 [ 3, 3, 3, 3, 2, 3, 4, 3,... ] => 추정 laptop\_computer : 정답 ring\_binder => X

추정확률분포 [ 4, 3, 3, 3, 3, 3, 4, 3,... ] => 추정 monitor : 정답 pen => X

추정확률분포 [ 3, 3, 3, 3, 2, 3, 4, 3,... ] => 추정 laptop\_computer : 정답 mouse => X

그림 1 - 폭 [10]의 은닉 계층을 가지는 다층 퍼셉트론

```
om1 = AdamModel('office31_model_1', od, [64,32,10])
om1.exec_all(epoch_count=20, report=10)
```

Model office31\_model\_1 train started:  
Epoch 10: cost=4.296, accuracy=0.685+0.040/0.800+0.040 (109/109 secs)  
Epoch 20: cost=4.269, accuracy=0.685+0.028/0.780+0.000 (110/219 secs)  
Model office31\_model\_1 train ended in 219 secs:  
Model office31\_model\_1 test report: accuracy = 0.661+0.040, (0 secs)

Model office31\_model\_1 Visualization



[ 도메인 추정결과 ]

추정확률분포 [65,15,20] => 추정 amazon : 정답 amazon => 0

추정확률분포 [65,15,20] => 추정 amazon : 정답 amazon => 0

추정확률분포 [65,15,20] => 추정 amazon : 정답 amazon => 0

[ 상품 추정결과 ]

추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3,... ] => 추정 monitor : 정답 bike\_helmet => X

추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3,... ] => 추정 monitor : 정답 keyboard => X

추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3,... ] => 추정 monitor : 정답 scissors => X

그림 2 - 폭 [64,32,10]의 은닉 계층을 가지는 다층 퍼셉트론

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

Model office31\_model\_2 train started:  
 Epoch 10: cost=4.482, accuracy=0.685+0.030/0.780+0.020 (100/100 secs)  
 Epoch 20: cost=4.436, accuracy=0.685+0.037/0.750+0.020 (100/200 secs)  
 Epoch 30: cost=4.400, accuracy=0.685+0.041/0.710+0.040 (108/308 secs)  
 Epoch 40: cost=4.371, accuracy=0.685+0.038/0.780+0.020 (101/409 secs)  
 Epoch 50: cost=4.349, accuracy=0.685+0.042/0.800+0.060 (102/511 secs)  
 Model office31\_model\_2 train ended in 511 secs:  
 Model office31\_model\_2 test report: accuracy = 0.661+0.038, (0 secs)

Model office31\_model\_2 Visualization



[ 도메인 추정결과 ]

추정확률분포 [49,24,27] => 추정 amazon : 정답 amazon => 0  
 추정확률분포 [49,24,27] => 추정 amazon : 정답 amazon => 0  
 추정확률분포 [49,24,27] => 추정 amazon : 정답 amazon => 0

[ 상품 추정결과 ]

추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3, ... ] => 추정 laptop\_computer : 정답 calculator => X  
 추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3, ... ] => 추정 speaker : 정답 tape\_dispenser => X  
 추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3, ... ] => 추정 laptop\_computer : 정답 projector => X

그림 3 - 그림 2 에서 학습 횟수를 50 으로 높이고 학습률을 0.0001 으로 변화

## 2.2 Adam 알고리즘을 이용한 경우

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

Model office31\_model\_1 train started:  
 Epoch 10: cost=4.144, accuracy=0.685+0.056/0.740+0.030 (106/106 secs)  
 Epoch 20: cost=4.097, accuracy=0.719+0.053/0.800+0.020 (108/214 secs)  
 Model office31\_model\_1 train ended in 214 secs:  
 Model office31\_model\_1 test report: accuracy = 0.752+0.051, (0 secs)

Model office31\_model\_1 Visualization



[ 도메인 추정결과 ]

추정확률분포 [86, 7, 8] => 추정 amazon : 정답 amazon => 0  
 추정확률분포 [86, 7, 8] => 추정 amazon : 정답 amazon => 0  
 추정확률분포 [86, 7, 8] => 추정 amazon : 정답 amazon => 0

[ 상품 추정결과 ]

추정확률분포 [ 3, 1, 3, 3, 1, 4, 4, 2, ... ] => 추정 monitor : 정답 calculator => X  
 추정확률분포 [ 3, 1, 3, 3, 1, 4, 4, 2, ... ] => 추정 monitor : 정답 desktop\_computer => X  
 추정확률분포 [ 3, 1, 3, 3, 1, 4, 4, 2, ... ] => 추정 monitor : 정답 punchers => X

그림 4 - 폭 [10]의 은닉 계층과 Adam 알고리즘 이용

```
om2 = AdamModel('office31_model_2', od, [64,32,10])
om2.use_adam = True
om2.exec_all(epoch_count=20, report=10)
```

Model office31\_model\_2 train started:

Epoch 10: cost=4.329, accuracy=0.680+0.037/0.720+0.030 (568/568 secs)

Epoch 20: cost=4.127, accuracy=0.706+0.048/0.830+0.040 (497/1065 secs)

Model office31\_model\_2 train ended in 1065 secs:

Model office31\_model\_2 test report: accuracy = 0.718+0.040, (0 secs)

Model office31\_model\_2 Visualization



[ 도메인 추정결과 ]

추정확률분포 [84, 6,10] => 추정 amazon : 정답 amazon => 0

추정확률분포 [73,11,17] => 추정 amazon : 정답 amazon => 0

추정확률분포 [84, 6,10] => 추정 amazon : 정답 amazon => 0

[ 상품 추정결과 ]

추정확률분포 [ 2, 2, 2, 3, 1, 3, 4, 3,... ] => 추정 paper\_notebook : 정답 projector => X

추정확률분포 [ 3, 3, 3, 3, 2, 3, 3, 4,... ] => 추정 pen : 정답 paper\_notebook => X

추정확률분포 [ 2, 2, 2, 3, 1, 3, 4, 3,... ] => 추정 paper\_notebook : 정답 monitor => X

그림 5 - 폭 [64,32,10]의 은닉 계층과 Adam 알고리즘 이용

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

Model office31\_model\_3 train started:

Epoch 10: cost=4.251, accuracy=0.685+0.043/0.780+0.030 (499/499 secs)

Epoch 20: cost=4.202, accuracy=0.685+0.046/0.720+0.030 (494/993 secs)

Epoch 30: cost=4.172, accuracy=0.685+0.053/0.780+0.040 (501/1494 secs)

Epoch 40: cost=4.166, accuracy=0.686+0.055/0.810+0.040 (620/2114 secs)

Epoch 50: cost=4.133, accuracy=0.685+0.053/0.740+0.030 (503/2617 secs)

Model office31\_model\_3 train ended in 2617 secs:

Model office31\_model\_3 test report: accuracy = 0.661+0.043, (1 secs)

Model office31\_model\_3 Visualization



[ 도메인 추정결과 ]

추정확률분포 [85, 6, 9] => 추정 amazon : 정답 amazon => 0

추정확률분포 [73, 9,17] => 추정 amazon : 정답 amazon => 0

추정확률분포 [85, 6, 9] => 추정 amazon : 정답 amazon => 0

[ 상품 추정결과 ]

추정확률분포 [ 1, 2, 2, 3, 1, 3, 4, 2,... ] => 추정 projector : 정답 keyboard => X

추정확률분포 [ 2, 3, 3, 3, 2, 4, 3, 3,... ] => 추정 laptop\_computer : 정답 ruler => X

추정확률분포 [ 1, 2, 2, 3, 1, 3, 4, 2,... ] => 추정 projector : 정답 bookcase => X

그림 6 - 그림 5 에서 학습 횟수를 50 으로 높이고 학습률을 0.0001 으로 변화

### 3. 총평

2 에서 진행한 구현 결과를 표로 정리하면 다음과 같다.

표 - 1 다층 퍼셉트론

	그림 1	그림 2	그림 3
은닉 계층 수	1	3	3
은닉 계층 폭	[10]	[64,32,10]	[64,32,10]
학습 횟수	20	20	50
학습률	0.001	0.001	0.0001
도메인 선택 정확도	66.1%	66.1%	66.1%
품목 선택 정확도	3.4%	4.0%	3.8%

표 - 2 아담 알고리즘 이용

	그림 4	그림 5	그림 6
은닉 계층 수	1	3	3
은닉 계층 폭	[10]	[64,32,10]	[64,32,10]
학습 횟수	20	20	50
학습률	0.001	0.001	0.0001
도메인 선택 정확도	75.2%	71.8%	66.1%
품목 선택 정확도	5.1%	4.0%	4.3%

위 표에서 확인할 수 있는 것처럼 아담 알고리즘을 사용할 때는 기존의 경사 하강법을 사용할 때에 비해 정확도가 대부분 올라갔다. 기존의 경사 하강법이라면 역전파 처리 때 해당 파라미터의 손실 기울기와 학습률을 곱한 값을 각 파라미터에서 빼주는 간단한 방식으로 학습을 수행한다. 그와 다르게, 아담 알고리즘은 모멘텀 개념을 통해 최근의 파라미터값의 변화 추세를 반영하여 정확도 향상을 할 수 있도록 해준다. 하지만, 계산 부담이 커지기 때문에 전체적으로 메모리 소비와 계산량이 증가하고 이에 따라 위 그림들에서 확인 할 수 있듯이, 시간이 오래걸렸다.