

6장. 복합 출력의 처리 방법: 오피스31 다차원 분류 신경망

부산대학교 항공우주공학과

정대현 201527137

1. 개요

지난번 실험의 분류에 차원을 사용해서 표현하면 한 번만 선택 판단을 하는 1차원 문제였습니다. 이번에는 두 가지 차원에서 동시에 분류를 수행하는 다차원 분류 신경망을 학습하게 됩니다. 물론 다차원 분류 문제 역시 차원 축소 기법을 통해서 보다 낮은 차원에서 진행할 수 있으나 차원이 늘어날수록 출력이 과도하게 커지며 도메인 특성이나 품목 특성 또한 따로 포착하기 어려워져 학습 성과가 좋게 나오지 않을 수 있습니다. 다차원 학습에 적합한 전이 학습방법에 대해 실험하며, 더해서 아담(ADAM, Adaptive Moments) 알고리즘을 적용해 신경망 모델의 학습 품질을 향상시키는 학습을 합니다.

2. 아담(ADAM, Adaptive Moments) vs SGD(Stochastic Gradient Descent)

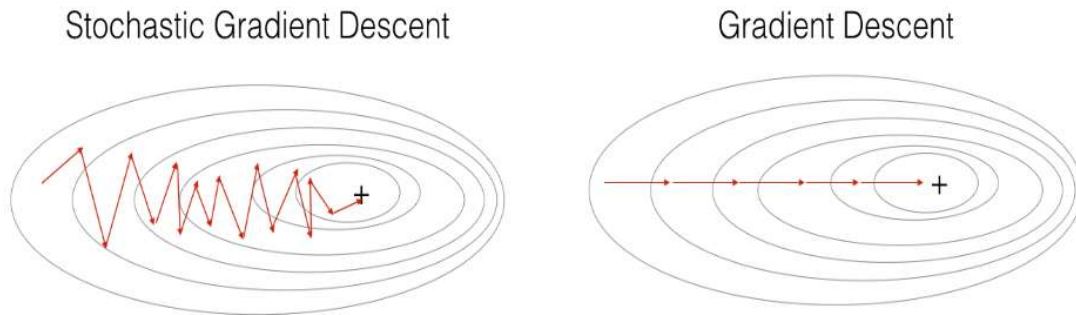


Figure 1: SGD vs GD

"+" denotes a minimum of the cost. SGD leads to many oscillations to reach convergence. But each step is a lot faster to compute for SGD than for GD, as it uses only one training example (vs. the whole batch for GD).

확률적 경사하강법(Stochastic Gradient Descent)은 추출된 데이터 각각에 대해 기울기를 계산하고, 경사 하강 알고리즘을 적용하는 방법입니다. 전체 데이터를 사용하는 것이 아닌 랜덤하게 추출한 일부 데이터를 사용하는 것으로 학습 중간 과정에서 결과의 진폭이 크고 불안정하지만 속도가 매우 빠른 특징이 있습니다. 데이터를 개별적으로 처리하기 때문에 오차율이 크고 GPU의 성능을 모두 활용하지 못하는 단점을 가지고 있습니다.

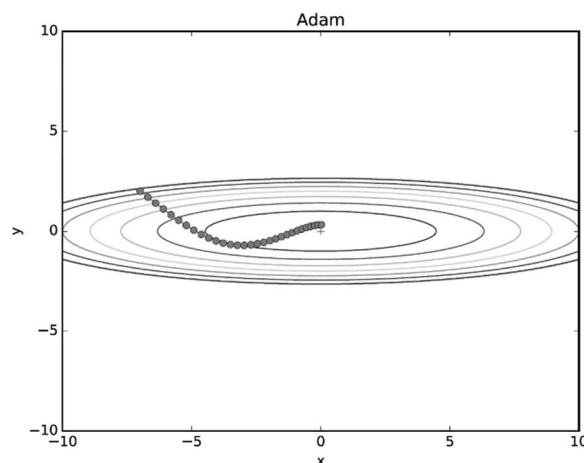


Figure 2. Adam 모멘텀 갱신 경로 시각화

이번 실험에 사용할 Optimizer은 아담 알고리즘입니다. 파라미터에 적용되는 실질적인 학습률을 개별 파라미터 별로 동적으로 조절해 경사하강법의 동작을 보완하고 학습 품질을 높여주는 방법입니다. 기존 경사하강법에서 역전파 처리 때 해당 파라미터의 손실 기울기와 학습률을 곱한 값을 각 파라미터에서 빼 주는 간단한 방식으로 학습을 수행하나 아담 알고리즘에서는 이를 모멘텀 개념을 도입해 처리 과정을 보완합니다. 하지만 단점으로는 모멘텀 정보는 개별 파라미터 수준에서 따로 계산되고 관리됩니다. 학습이 이루어 질때마다 파라미터 값과 동시에 모멘텀 정보도 처음부터 계산하게 되며 메모리 소비량이 3배정도 늘어나게 됩니다.

3. 전이 학습(Transfer Learning)

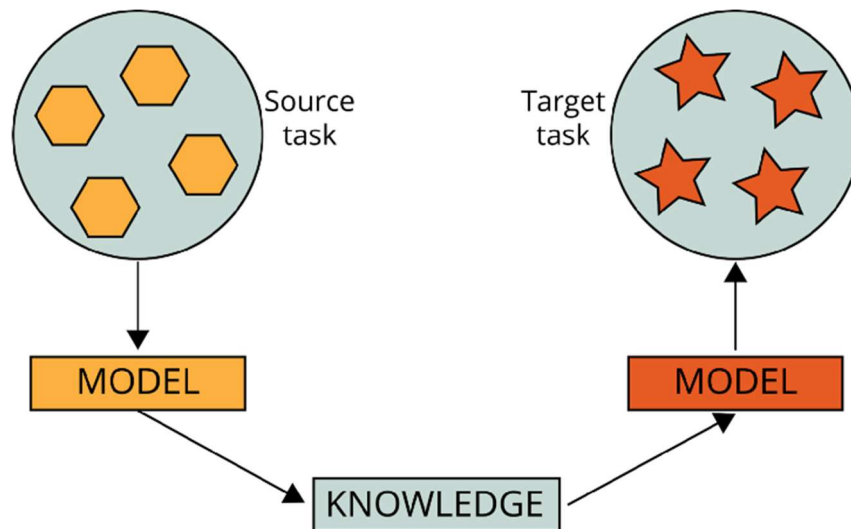


Fig 2. 전이학습 개요

전이 학습(Transfer Learning)은 특정 분야에서 학습된 신경망의 일부 능력을 유사하거나 전혀 새로운 분야에서 사용되는 신경망의 학습에 이용하는 것을 의미합니다. 전통적인 머신러닝 기법은 단일 과제에 대해서 학습을 진행했습니다. 즉 데이터셋과 학습시스템은 1대1 관계였습니다. 하지만 전이 학습은 한 도메인에서 학습시킨 결과를 다른 도메인에 활용하여 학습 효과를 높이는 기법입니다.

이번 실험의 목적은 '레이블링을 어떻게 하면 효과적으로 할 수 있는가?' 입니다. 지도 학습을 위해서는 완벽하게 레이블링이 되어 있는 방대한 데이터셋을 필요로 합니다. 하지만 매 애플리케이션마다 방대한 양의 포괄적인 데이터셋을 구축하기란 쉽지 않을 뿐만 아니라 현실성도 떨어집니다. 이럴 때 전이 학습은 데이터셋을 사용해서 학습된 모델(사전에 학습된 모델)의 입력 층과 중간 층을 재사용하는 방법으로서, 구체적이고 포괄적인 데이터셋이 부족한 문제를 극복합니다. 새로운 작업을 하기 위해서 최종 층만 다시 학습을 시키면 됩니다. 사전에 학습된 모델의 파라미터들을 사용하고 그리고 학습 단계에서 최대의 정확도를 달성하도록 조절합니다. 매 특정한 애플리케이션마다 모든 층들을 처음부터 학습시킬 필요가 없으므로 데이터의 수가 적을 때도 효과적이며, 학습 속도도 빠릅니다. 하지만 단점으로 왜곡을 방지하기 위해서 천천히 진행해야 하고, 사전에 학습된 데이터셋에 이미 존재하는 파라미터들에 의해서 제약을 받게 됩니다.

4. 실험

4.1 아담알고리즘 False, 은닉계층 [10] epoch 20회, learning_rate=0.001(default)

```
Epoch 10: cost=4.296, accuracy=0.685+0.035/0.790+0.040 (16/16 secs)
Epoch 20: cost=4.268, accuracy=0.685+0.037/0.780+0.020 (15/31 secs)
Model office31_model_1 train ended in 31 secs:
Model office31_model_1 test report: accuracy = 0.661+0.048, (0 secs)
[ 도메인 추정결과 ]
추정확률분포 [65,15,20] => 추정 amazon : 정답 amazon => O
추정확률분포 [65,15,20] => 추정 amazon : 정답 amazon => O
추정확률분포 [65,15,20] => 추정 amazon : 정답 dslr => X
[ 상품 추정결과 ]
추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3,...] => 추정 monitor : 정답 ring_binder => X
추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3,...] => 추정 monitor : 정답 pen => X
추정확률분포 [ 3, 3, 3, 3, 3, 3, 3, 3,...] => 추정 monitor : 정답 mouse => X
```

4.2 아담알고리즘 False, 은닉계층 [64,32,10] epoch 50회, learning_rate=0.0001

```
Epoch 10: cost=3.745, accuracy=0.805+0.098/0.860+0.080 (41/41 secs)
Epoch 20: cost=3.471, accuracy=0.843+0.146/0.830+0.090 (43/84 secs)
Epoch 30: cost=3.263, accuracy=0.866+0.198/0.840+0.100 (43/127 secs)
Epoch 40: cost=3.106, accuracy=0.880+0.223/0.880+0.140 (43/170 secs)
Epoch 50: cost=2.964, accuracy=0.889+0.252/0.880+0.160 (42/212 secs)
Model office31_model_2 train ended in 212 secs:
Model office31_model_2 test report: accuracy = 0.867+0.197, (0 secs)
[ 도메인 추정결과 ]
추정확률분포 [100, 0, 0] => 추정 amazon : 정답 amazon => O
추정확률분포 [100, 0, 0] => 추정 amazon : 정답 amazon => O
추정확률분포 [96, 1, 3] => 추정 amazon : 정답 amazon => O
[ 상품 추정결과 ]
추정확률분포 [ 1, 0, 0,23, 3, 9, 6,16,...] => 추정 bookcase : 정답 tape_dispenser => X
추정확률분포 [ 0, 1, 4, 0, 0, 0, 0, 0,...] => 추정 keyboard : 정답 letter_tray => X
추정확률분포 [ 0, 1, 0, 1, 1, 3, 1, 8,...] => 추정 pen : 정답 ruler => X
```

4.3 아담알고리즘 True, 은닉계층 [64,32,10] epoch 50회, learning_rate=0.0001

```
Epoch 10: cost=3.711, accuracy=0.819+0.084/0.860+0.060 (43/43 secs)
Epoch 20: cost=3.311, accuracy=0.869+0.163/0.860+0.150 (44/87 secs)
Epoch 30: cost=3.103, accuracy=0.883+0.204/0.870+0.160 (44/131 secs)
Epoch 40: cost=2.935, accuracy=0.894+0.241/0.860+0.230 (43/174 secs)
Epoch 50: cost=2.799, accuracy=0.904+0.264/0.850+0.180 (43/217 secs)
Model office31_model_3 train ended in 217 secs:
Model office31_model_3 test report: accuracy = 0.890+0.234, (0 secs)
```

[도메인 추정결과]

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

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

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

[상품 추정결과]

추정확률분포 [0, 1, 1, 2, 5, 2, 1,12,...] => 추정 desk_lamp : 정답 scissors => X

추정확률분포 [3, 1, 1,10, 6, 2, 0,11,...] => 추정 paper_notebook : 정답 printer => X

추정확률분포 [0, 0, 2, 2, 0, 1, 2, 1,...] => 추정 stapler : 정답 calculator => X

4.4 아담알고리즘 True, 은닉계층 [100,50,25] epoch 50회, learning_rate=0.0001

Epoch 10: cost=3.377, accuracy=0.838+0.168/0.840+0.140 (58/58 secs)

Epoch 20: cost=3.024, accuracy=0.873+0.246/0.890+0.210 (64/122 secs)

Epoch 30: cost=2.783, accuracy=0.880+0.300/0.790+0.240 (63/185 secs)

Epoch 40: cost=2.558, accuracy=0.900+0.339/0.830+0.280 (67/252 secs)

Epoch 50: cost=2.367, accuracy=0.909+0.398/0.870+0.200 (69/321 secs)

Model office31_model_3 train ended in 321 secs:

Model office31_model_3 test report: accuracy = 0.851+0.296, (0 secs)

[도메인 추정결과]

추정확률분포 [21,11,68] => 추정 webcam : 정답 amazon => X

추정확률분포 [87, 1,11] => 추정 amazon : 정답 amazon => O

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

[상품 추정결과]

추정확률분포 [0, 0, 0, 0, 0, 4, 0, 0,...] => 추정 speaker : 정답 tape_dispenser => X

추정확률분포 [0, 1, 0, 0, 0, 1, 0, 0,...] => 추정 keyboard : 정답 mug => X

추정확률분포 [0, 0, 0, 1,55, 4, 0, 3,...] => 추정 bottle : 정답 keyboard => X

4.5 아담알고리즘 True, 은닉계층 [100,50,25] epoch 100회, learning_rate=0.0001

Epoch 10: cost=3.583, accuracy=0.816+0.119/0.840+0.090 (58/58 secs)

Epoch 30: cost=2.874, accuracy=0.880+0.284/0.850+0.140 (63/181 secs)

Epoch 50: cost=2.451, accuracy=0.909+0.367/0.900+0.280 (66/310 secs)

Epoch 90: cost=1.817, accuracy=0.933+0.512/0.880+0.300 (66/575 secs)

Epoch 100: cost=1.659, accuracy=0.946+0.556/0.850+0.270 (67/642 secs)

Model office31_model_3 train ended in 642 secs:

Model office31_model_3 test report: accuracy = 0.874+0.317, (0 secs)

[도메인 추정결과]

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

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

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

[상품 추정결과]

추정확률분포 [0, 8, 0, 2, 0, 4, 0, 1,...] => 추정 scissors : 정답 tape_dispenser => X

추정확률분포 [0, 0, 1, 0, 0, 0, 0, 0,...] => 추정 printer : 정답 projector => X

추정확률분포 [0, 0, 0, 0, 0, 0, 0, 0,...] => 추정 projector : 정답 keyboard => X

4.6 아담알고리즘 True, 은닉계층 [120,60,30,10] epoch 50회, learning_rate=0.0001

```
Epoch 10: cost=3.853, accuracy=0.788+0.061/0.790+0.020 (66/66 secs)
Epoch 20: cost=3.585, accuracy=0.822+0.108/0.820+0.090 (72/138 secs)
Epoch 30: cost=3.361, accuracy=0.839+0.149/0.850+0.160 (73/211 secs)
Epoch 40: cost=3.234, accuracy=0.852+0.171/0.850+0.130 (77/288 secs)
Epoch 50: cost=3.137, accuracy=0.859+0.201/0.850+0.150 (78/366 secs)
Model office31_model_3 train ended in 366 secs:
Model office31_model_3 test report: accuracy = 0.830+0.190, (0 secs)
[ 도메인 추정결과 ]
추정확률분포 [100, 0, 0] => 추정 amazon : 정답 amazon => O
추정확률분포 [99, 1, 0] => 추정 amazon : 정답 amazon => O
추정확률분포 [90, 8, 2] => 추정 amazon : 정답 amazon => O
[ 상품 추정결과 ]
추정확률분포 [32, 0, 3, 7, 0, 2, 2, 1,...] => 추정 back_pack : 정답 speaker => X
추정확률분포 [ 0, 0, 6, 1, 0, 0, 0, 0,...] => 추정 monitor : 정답 speaker => X
추정확률분포 [ 0, 0, 5, 0, 0, 0, 0, 0,...] => 추정 monitor : 정답 monitor => O
```

실험	epoch	은닉계층	Learning_rate	Adam	도메인 정확도	상품 정확도	시간
1	10	[10]	0.001	False	0.661	0.048	31s
2	50	[64,32,10]	0.0001	False	0.867	0.197	212s
3	50	[64,32,10]	0.0001	True	0.890	0.234	217s
4	50	[100,50,25]	0.0001	True	0.851	0.296	321s
5	100	[100,50,25]	0.0001	True	0.874	0.317	642s
6	50	[120,60,30,10]	0.0001	True	0.830	0.190	366s

5. 결론

첫번째 실험에서 도메인 선택 추정 분포가 [65,15,20]로 일정하게 나오며 도메인 정확도는 0.661으로 나옵니다. 이는 66%의 정확도를 지녔다고 하기보다 모든 문제에 대해서 정답을 아마존이라고 응답했을 가능성이 높습니다. 상품추정결과 역시 [3, 3, 3, 3, 3, 3, 3, 3,...]으로 거의 똑같은 가중치를 주었으며 정확도 역시 0.048로 나왔습니다. 데이터셋 가운데 가장 많은 품목인 모니터의 비율이 4.8%인데 역시 모든 도메인 문제를 아마존이라 답한 것과 마찬가지로 모든 상품 추정에서도 모니터라고 답했을 가능성이 높습니다. 첫번째 학습 모델에서는 학습이 이루어졌다고 보기 힘듭니다.

두번째 실험에서는 epoch횟수와 은닉계층의 수를 늘렸으며 학습률은 더 작게 조정했습니다. 도메인 추정 분포와 상품 추정 분포 중 하나는 [96, 1, 3], [1, 0, 0,23, 3, 9, 6,16,...] 으로 첫번째 실험에서 샘플의 크기에 따라 분포가 결정된 것과 달리 상품 추정 결과 모두 데이터에 따라 서로 다른 추정 확률 분포가 구해지고 선택되는 답 역시 달라졌습니다. 높은 정확도를 지니지는 않았지만 학습이 어느정도 이루어진 것을 알 수 있습니다.

세번째 실험에서는 두번째 실험에 더해 아담알고리즘을 적용했습니다. 도메인 추정 분포와 상품 추정 분포는 데이터에 따라 서로 다른 추정 확률 분포를 얻을 수 있었으며 도메인 정확도는 2.3% 증가하였으며 상품 정확도는 3.7% 증가하였습니다. 단순히 아담알고리즘을 적용했다고 해서 획기적인 성능을 보여준 것은 아니지만 기본

적인 경사하강법 보다 아담 알고리즘을 이용할 때 조금이라도 더 나은 결과가 나왔습니다. 이후 4번째와 5번째 실험에서는 은닉계층의 폭과 epoch를 늘려 보았습니다. 제일 높은 성능을 보인 실험 5에서는 도메인 정확도는 87.4%, 상품 정확도는 31.7%를 기록했습니다. 실험 4와 5를 비교해보았을 때 사용된 시간은 두배정도 늘어났으나 향상된 정확도는 미미했으며 실험 6에서 폭을 4층으로 해보았으나 오히려 정확도는 감소했습니다.

지금까지 복합 출력을 다루는 방법과 학습 성능 개선에 사용하는 아담 알고리즘 활용방법을 살펴보았습니다. 지난 5장에서도 살펴보았듯이 파라미터가 지나치게 많을 경우 데이터 부족으로 학습이 어려워집니다. 같은 이유로 실험6번에서 은닉계층을 4개로 조절하였으나 오히려 정확도는 대폭 떨어졌습니다. 첫번째 실험의 경우 정확도는 데이터의 분포를 그대로 따라 갔으며, 데이터 불균형 문제도 학습에 영향을 미친 것으로 보입니다. 결국 모델을 어떻게 구성할 것인가에 대해서 고민을 하는 만큼 '어떻게 하면 좋은 학습데이터를 얻을 수 있을 것인가?', '구성할 것인가?' 역시 학습 모델의 성능을 결정하는 중요한 요소임을 다시 한번 확인하게 되었습니다.