13장. 오토인코더 : 엠니스트 이미지 재현 및 분류 신경망

부산대학교 항공우주공학과

정대현 201527137

1. 개요

오토인코더(autoencoder)는 입력을 재현하는 방식의 비지도학습으로 레이블 없이 데이터의 분포 특성을 스스로 파악하는 신경망을 뜻합니다. 오토인코더는 인코더와 디코더로 구성되며 잘 학습된 오토인코더의 인코더는 적은 양의 레이블링된 데이터만을 이용하는 지도학습과 미리 정해진 카테고리 없이 데이터 내용만으로 유사한 데이터를 찾는 시멘틱 해싱 비지도 학습 등 다양한 용도로 이용될 수 있습니다.

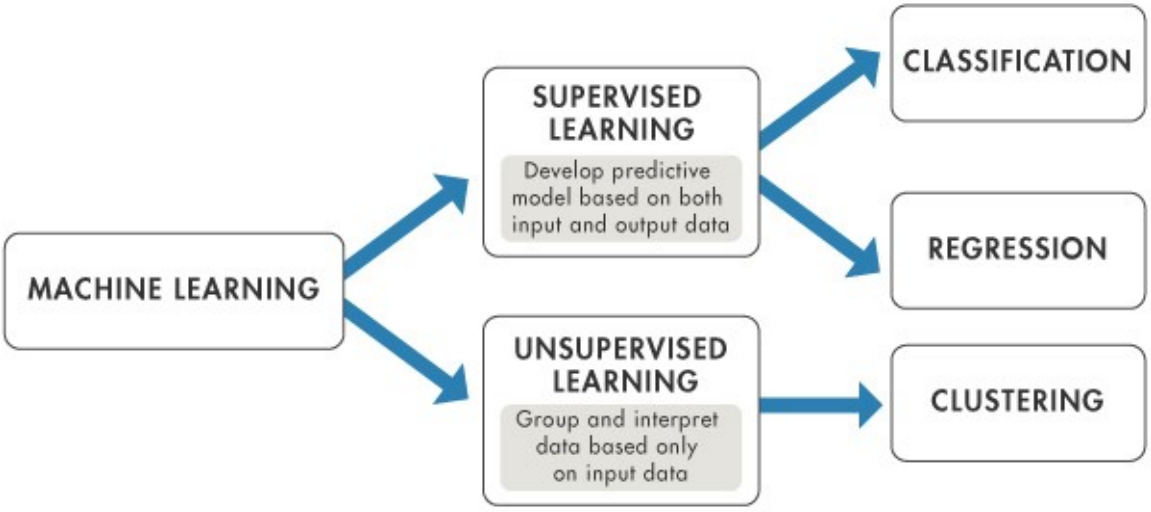

Fig. 1 지도학습과 비지도학습 차이

2. 지도학습과 비지도학습

2.1 지도학습(Supervised Learning)

지도학습은 정답이 있는 데이터를 활용해 데이터를 학습시키는 것입니다. 어떤 input 값이 주어지면 그 입력값에 대한 Label 을 주어 학습시키며 대표적인 지도학습은 분류(Classification), 회귀(Regression)등이 해당합니다. 단점으로는 학습용 데이터를 모으기 위해서 레이블링을 하는데 많은 시간과 비용이 들어갑니다.

2.2 비지도학습(Unsupervised Learning)

비지도 학습은 정답 라벨이 없는 데이터를 비슷한 특징끼리 군집화 하여 새로운 데이터에 대한 결과를 예측하는 방법을 뜻합니다. 라벨링 되어있지 않은 데이터로부터 패턴이나 형태를 찾기 때문에 지도학습보다 난이도가 있으며, 지도학습에서 적절한 feature 를 찾기 위한 전처리 방법으로 비지도 학습을 이용하기도 합니다. 비지도 학습의 대표적인 예시로는 군집화(Clustering)등이 있으며 최근 많이 연구가 이루어지는 적대적생성신경망 GAN(generative Adversarial Network) 모델도 비지도 학습에 해당합니다.

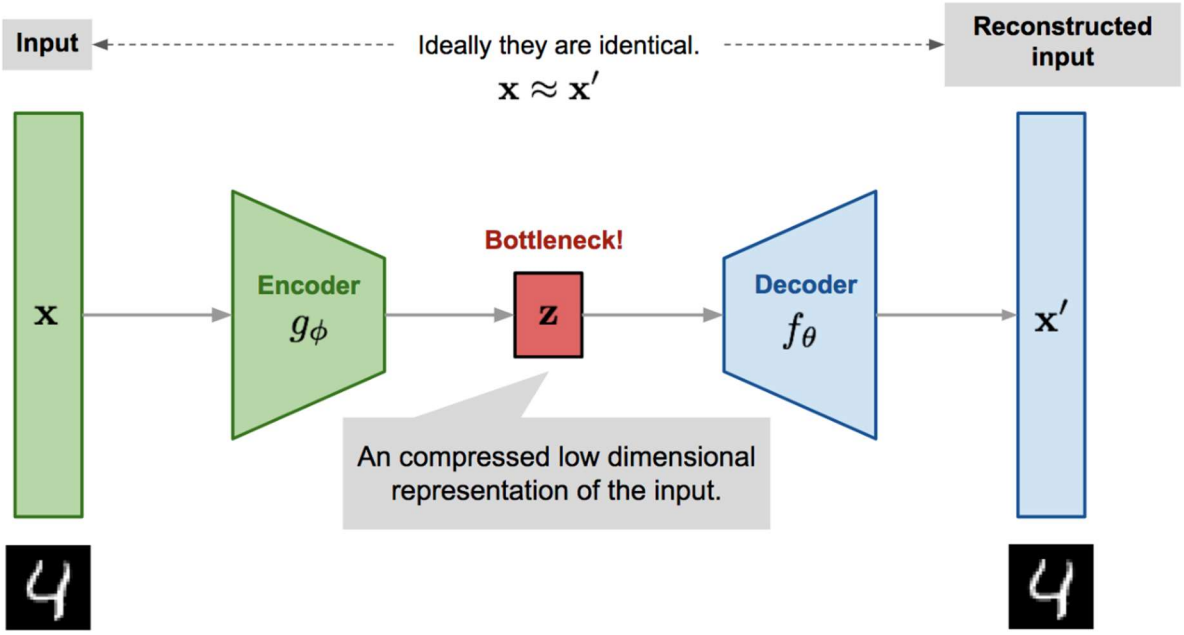

Fig. 2 오토인코더의 구성

3. 오토인코더

오토인코더는 인코더(encoder)와 디코더(decoder)의 두 부분으로 구성되며 디코더의 출력은 인코더의 입력과 같은 형태를 갖도록 설계됩니다. 오토인코더는 입력과 최대한 비슷한 출력을 만들어내는 것을 학습 목적으로 삼습니다. 출력은 입력과 동일한 형태를 가질 뿐만 아니라 내용까지도 최대한 비슷하게 재현(representation)해야 합니다. 따라서 오토인코더에 정답을 따로 제공할 필요가 없으며 입력 자체가 출력이 지향해야할 정답입니다. 오토인코더는 보통 입력과 출력 사이의 평균제곱오차를 손실 함수로 삼아 학습을 수행합니다.

오토인코더에서 인코더는 입력을 처리하여 입력의 내부 중간 표현으로 만들어내며 이 중간표현을 코드라고 합니다. 반대로 디코더는 인코더가 생성한 코드를 처리하여 출력을 만들게 됩니다. 이런 처리를 하는 목적은 입력을 코드로 변환해주는 인코더를 쓸모있는 모습으로 학습시키는 것이기 때문입니다.

4. 실험

4.1 폭이 10인 은닉계층 다층퍼셉트론 mset_all 데이터셋

```
Model mnist_mlp_all train started:
    Epoch 2: cost=0.514, accuracy=0.864/0.880 (3/3 secs)
    Epoch 4: cost=0.450, accuracy=0.884/0.930 (4/7 secs)
    Epoch 6: cost=0.424, accuracy=0.896/0.820 (4/11 secs)
    Epoch 8: cost=0.408, accuracy=0.901/0.860 (4/15 secs)
    Epoch 10: cost=0.403, accuracy=0.903/0.960 (4/19 secs)
Model mnist_mlp_all train ended in 19 secs:
Model mnist_mlp_all test report: accuracy = 0.900, (0 secs)
정답 [4 4 9]  vs.  추정 [4 4 9]
```

4.2 폭이 10인 은닉계층 다층퍼셉트론 mset_1p 데이터셋

```
Model mnist_mlp_1p train started:
    Epoch 2: cost=1.402, accuracy=0.533/0.540 (0/0 secs)
    Epoch 4: cost=0.870, accuracy=0.681/0.550 (1/1 secs)
    Epoch 6: cost=0.579, accuracy=0.781/0.670 (0/1 secs)
    Epoch 8: cost=0.419, accuracy=0.817/0.790 (0/1 secs)
    Epoch 10: cost=0.337, accuracy=0.871/0.750 (0/1 secs)
Model mnist_mlp_1p train ended in 1 secs:
Model mnist_mlp_1p test report: accuracy = 0.684, (0 secs)
Model mnist_mlp_1p Visualization
정답 [0 2 0]  vs.  추정 [0 8 5]
```

4.3 확장 인코더를 이용한 엠니스트 분류 모델 학습

```
Model mnist_auto_1 autoencode started:
    Epoch 2: cost=2556.567, accuracy=-0.526/-0.526 (5/5 secs)
    Epoch 4: cost=2545.596, accuracy=-0.523/-0.523 (7/12 secs)
    Epoch 6: cost=2542.326, accuracy=-0.522/-0.522 (9/21 secs)
    Epoch 8: cost=2540.189, accuracy=-0.522/-0.522 (8/29 secs)
    Epoch 10: cost=2538.084, accuracy=-0.521/-0.521 (8/37 secs)
Model mnist_auto_1 autoencode ended in 37 secs:
Model mnist_auto_1 train started:
    Epoch 2: cost=1.375, accuracy=0.552/0.540 (0/0 secs)
    Epoch 4: cost=0.955, accuracy=0.677/0.590 (0/0 secs)
    Epoch 6: cost=0.821, accuracy=0.706/0.640 (0/0 secs)
    Epoch 8: cost=0.718, accuracy=0.758/0.660 (0/0 secs)
    Epoch 10: cost=0.638, accuracy=0.787/0.760 (0/0 secs)
Model mnist_auto_1 train ended in 0 secs:
Model mnist_auto_1 test report: accuracy = 0.746, (0 secs)
Model mnist_auto_1 Visualization
정답 [6 9 9]  vs.  추정 [6 9 9]
```

4.4 지도학습시 인코더 학습을 동결시킨 확장 오토인코더 모델 학습

```
Model mnist_auto_fix autoencode started:
    Epoch 5: cost=2448.894, accuracy=-0.493/-0.493 (16/16 secs)
    Epoch 10: cost=2445.195, accuracy=-0.492/-0.492 (19/35 secs)
Model mnist_auto_fix autoencode ended in 35 secs:
Model mnist_auto_fix train started:
```

```
    Epoch 5: cost=0.734, accuracy=0.740/0.690 (0/0 secs)
    Epoch 10: cost=0.630, accuracy=0.798/0.740 (0/0 secs)
Model mnist_auto_fix train ended in 0 secs:
Model mnist_auto_fix test report: accuracy = 0.749, (0 secs)
Model mnist_auto_fix Visualization
정답 [9 8 3]  vs.  추정 [8 8 3]
```

4.5 인코더와 디코더 구성을 강화한 확장 오토인코더 모델 학습

```
Model mnist_auto_2 autoencode started:
    Epoch 5: cost=2692.180, accuracy=-0.564/-0.564 (112/112 secs)
    Epoch 10: cost=2646.832, accuracy=-0.551/-0.551 (140/252 secs)
Model mnist_auto_2 autoencode ended in 252 secs:
Model mnist_auto_2 train started:
    Epoch 5: cost=0.764, accuracy=0.735/0.660 (1/1 secs)
    Epoch 10: cost=0.611, accuracy=0.798/0.880 (0/1 secs)
Model mnist_auto_2 train ended in 1 secs:
Model mnist_auto_2 test report: accuracy = 0.796, (0 secs)
Model mnist_auto_2 Visualization
정답 [7 6 8]  vs.  추정 [7 6 8]
```

4.6 중간 코드 폭을 축소시킨 확장 오토인코더 모델 학습

```
Model mnist_auto_3 autoencode started:
    Epoch 5: cost=5323.163, accuracy=-1.201/-1.201 (83/83 secs)
    Epoch 10: cost=5090.021, accuracy=-1.152/-1.152 (137/220 secs)
Model mnist_auto_3 autoencode ended in 220 secs:
Model mnist_auto_3 train started:
    Epoch 5: cost=2.223, accuracy=0.154/0.180 (1/1 secs)
    Epoch 10: cost=2.192, accuracy=0.177/0.130 (1/2 secs)
Model mnist_auto_3 train ended in 2 secs:
Model mnist_auto_3 test report: accuracy = 0.187, (0 secs)
Model mnist_auto_3 Visualization
정답 [3 6 6]  vs.  추정 [0 3 0]
```

실험	조건	정확도
1	다층퍼셉트론 mset_all 데이터셋	0.900
2	다층퍼셉트론 mset_1p 데이터셋	0.684

3	확장 인코더를 이용한 엠니스트 분류	0.746
4	지도학습시 인코더 학습을 동결시킨 확장 오토인코더	0.749
5	인코더와 디코더 구성을 강화한 확장 오토인코더	0.796
6	중간 코드 폭을 축소시킨 확장 오토인코더	0.187

5. 결론

앞서 진행했던 실험은 모두 지도학습으로 정답 Label을 가지고 있는 데이터만으로 학습을 진행하였습니다. 이번 오토인코더에서는 비지도학습으로 입력 재현을 출력 목표로 삼는 오토 인코딩 과정을 통하여 레이블 정보 없이도 입력의 분포특성을 학습하는 오토인코더를 학습하고 오토인코딩 결과를 지도학습에 사용하는 방법을 살펴보았습니다. 오토인코더와 다층퍼셉트론을 비교해보고 중간 폭을 다양하게 조절하여 실험을 진행하였습니다.

가장 먼저 1,2번과 3번 실험을 비교해보았을 때 실험 3번의 정확도는 74.6%가 구해졌습니다. 전체 학습데이터로 학습시킨 실험 1번 모델의 90%에는 미치지 못하지만 같은 1%의 레이블 정보로 학습한 실험2의 모델의 68.4%보다는 향상된 결과를 얻었습니다. 시각화 결과에서도 추정결과를 모두 맞힌 것을 확인할 수 있습니다.

실험 4의 경우 지도학습 단계에서 인코더 파라미터가 갱신되지 않도록 고정하고 실험을 했습니다. 이렇게 하면 인코더는 오토인코딩 학습에서만 학습될뿐 지도학습 단계에서는 순전파 과정에서 자신의 처리 결과를 지도학습기에 전달할 뿐입니다. 즉 역전파 과정에서는 학습을 수행하지 않습니다. 그렇다고 학습이 절반만 되는 것은 아니며 지도학습기에서 모든 학습이 이루어지기 때문에 모델의 품질이 저하될 가능성이 높지만 실험에서는 74.6%에서 74.9%로 소폭 증가했습니다. 예상한 것 만큼 심각한 품질 훼손은 없는 것으로 보입니다.

실험 5에서는 인코더와 디코더에 폭 64의 계층을 하나씩 추가하여 모델을 강화하였습니다. 기존 학습모델에 비해서 5%의 성능 향상을 얻을 수 있습니다.

마지막 실험6에서는 반대로 중간 코드폭을 크게 줄였습니다. 인코더의 두번째 완전 연결 계층 폭을 10에서 1로 줄여서 학습하였으며 중간 코드의 폭이 1이 됨에 따라 지도학습기도 최초 입력 벡터의 크기가 10에서 1로 작아지는 변화를 겪게 됩니다. 적은 양의 정보만 갖게 되는 만큼 성능 하락이 예상되며 실제 18.7%로 형편없는 정확도를 얻게 되었습니다.