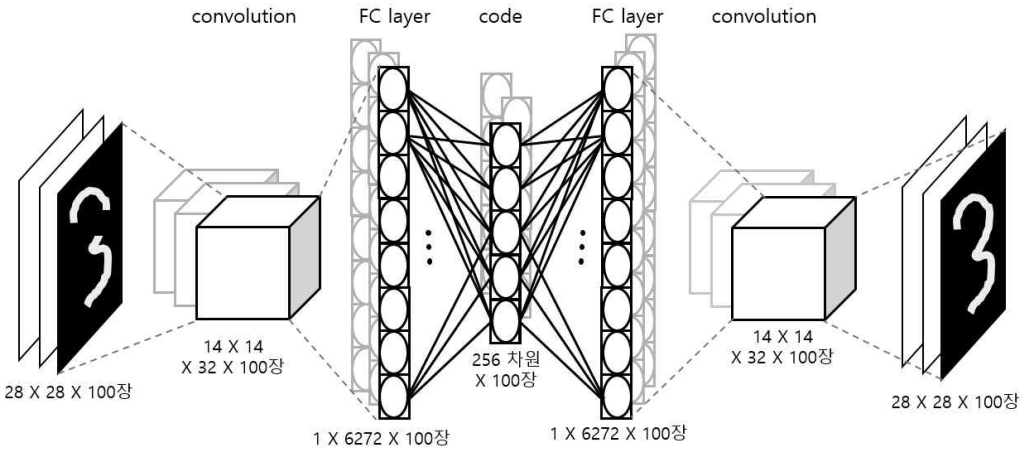
딥러닝 HW 4

빅데이터 플랫폼
곽민지

1. Denoising autoencoder의 네트워크 구조

네트워크 구조는 다음 그림과 같이 요약할 수 있다.
(batch size=100, 아래 그림은 제가 PPT로 직접 제작한 그림입니다.)



우선 픽셀의 인접한 지점끼리 학습하여 feature를 추출하기 위해 convolution layer를 생성하였고 그 결과 도출한 convolution layer를 1차원으로 펼쳐서 fully connected하여 256차원으로 축소하였다. 이 과정이 encoder가 한 일이다.

256차원으로 축소한 feature를 가지고 다시 fully connect하여 6272차원의 layer로 확장하였다. 이 layer를 3차원으로 변환하여 convolution을 거쳐 2차원의 그림을 최종적으로 출력하였다.

2. 인자 탐색 관정 및 결과

* 처음에는 fully connected한 network를 통해 encoding을 하고 다시 fully connected한 network를 통과시켜서 decoding을 수행하였다.

* 그 후에 원래의 encoding 앞부분에 convolutional network를 붙이고 decoding 뒷부분에 convolutional network를 붙여서 layer를 늘렸다. cost가 더 잘 감소하는 경향을 보였다. 40번의 batch 학습 결과 cost가 0.0083으로 감소하였다.

* learning rate를 0.01에서 0.001로 변경 후 40 epoch를 돌렸다니 cost 가 0.0046까지 떨어졌으며 그림도 더 뚜렷해졌다.
Epoch: 0036 Avg. cost = 0.0047

Epoch: 0037 Avg. cost = 0.0047
Epoch: 0038 Avg. cost = 0.0047
Epoch: 0039 Avg. cost = 0.0047
Epoch: 0040 Avg. cost = 0.0046

* convolution에서의 kernel의 층 수(knum)를 32에서 16으로 줄여봤으나 cost는 줄어들지 않고 오히려 늘었다. knum 값은 32로 유지하기로 하였다.

* hidden layer의 차원수 (n_hidden)도 128로 감소시켜보았으나 cost에는 크게 변화가 없었다.

* hidden layer의 차원수(n_hidden)을 512로 증가시켰더니 cost가 잘 감소하였으나 256일때와는 크게 차이가 없었다.

<hyper parameter 튜닝>
n_hidden = 256
learning_rate = 0.001
knum = 32
ksize = 5

3. 결과 토론

*input images

*hole images

*reconstructed images

* 1, 7, 2 등 직선 구성이 많은 숫자가 잘 채워지는 경향이 있었다. 모델 자체가 한붓으로 이어진 숫자 모델을 가정하고 있어서 직선 부분의 경우 그 곳에 큰 공간이 생긴다면 그곳에 채워져야 한다는 사실을 예측하기 쉽기 때문으로 보인다.

*가장자리와 둥근 부분에 구멍이 생긴 경우 상대적으로 복원력이 떨어졌다. 그런 부분의 경우 숫자의 가장자리 일부분이 아주 조금 뜯겨나갔음에도 전혀 새로운 모양의 숫자로 과잉 복원하는 경우가 있었다.