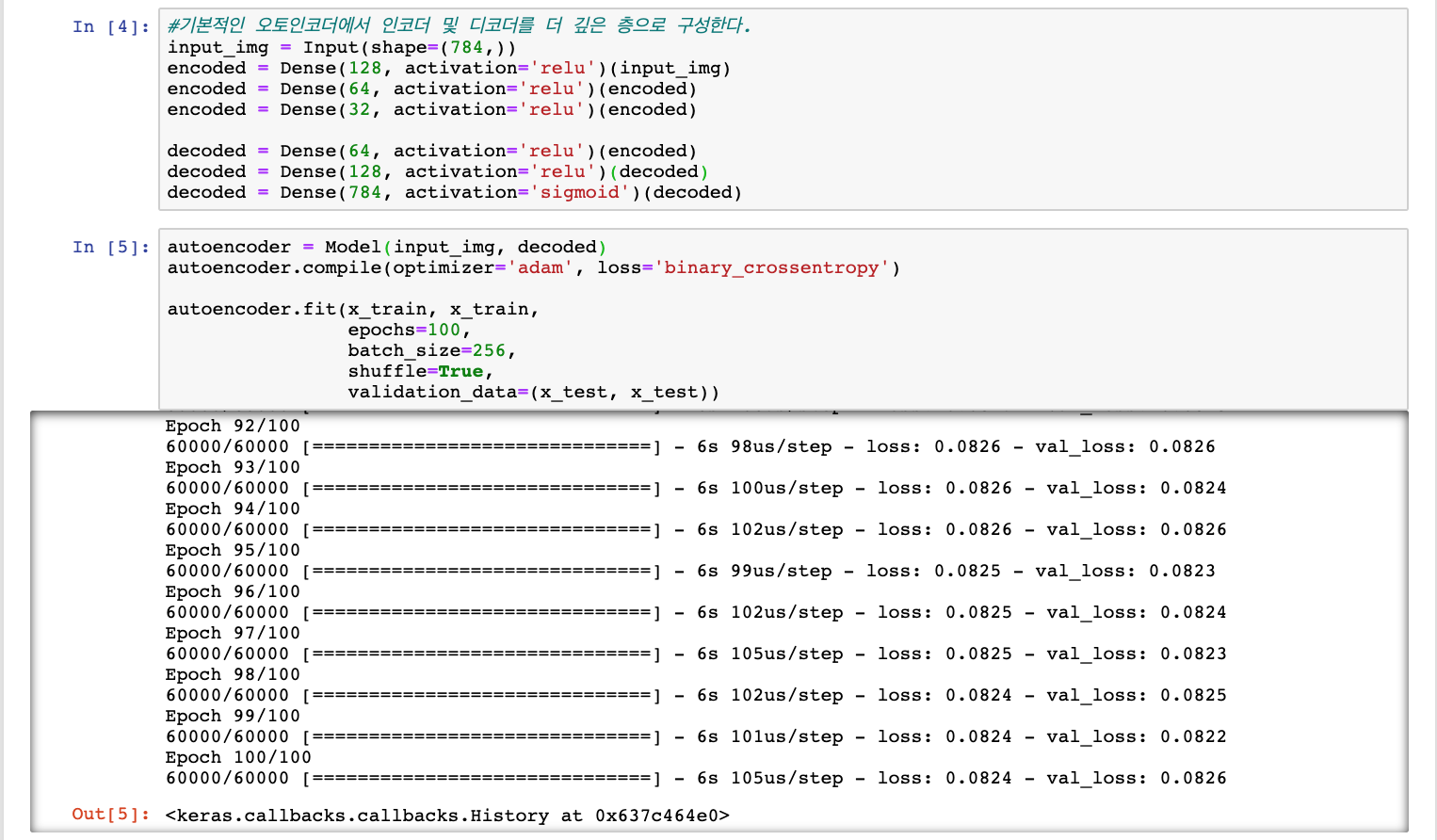
머신러닝과 딥러닝

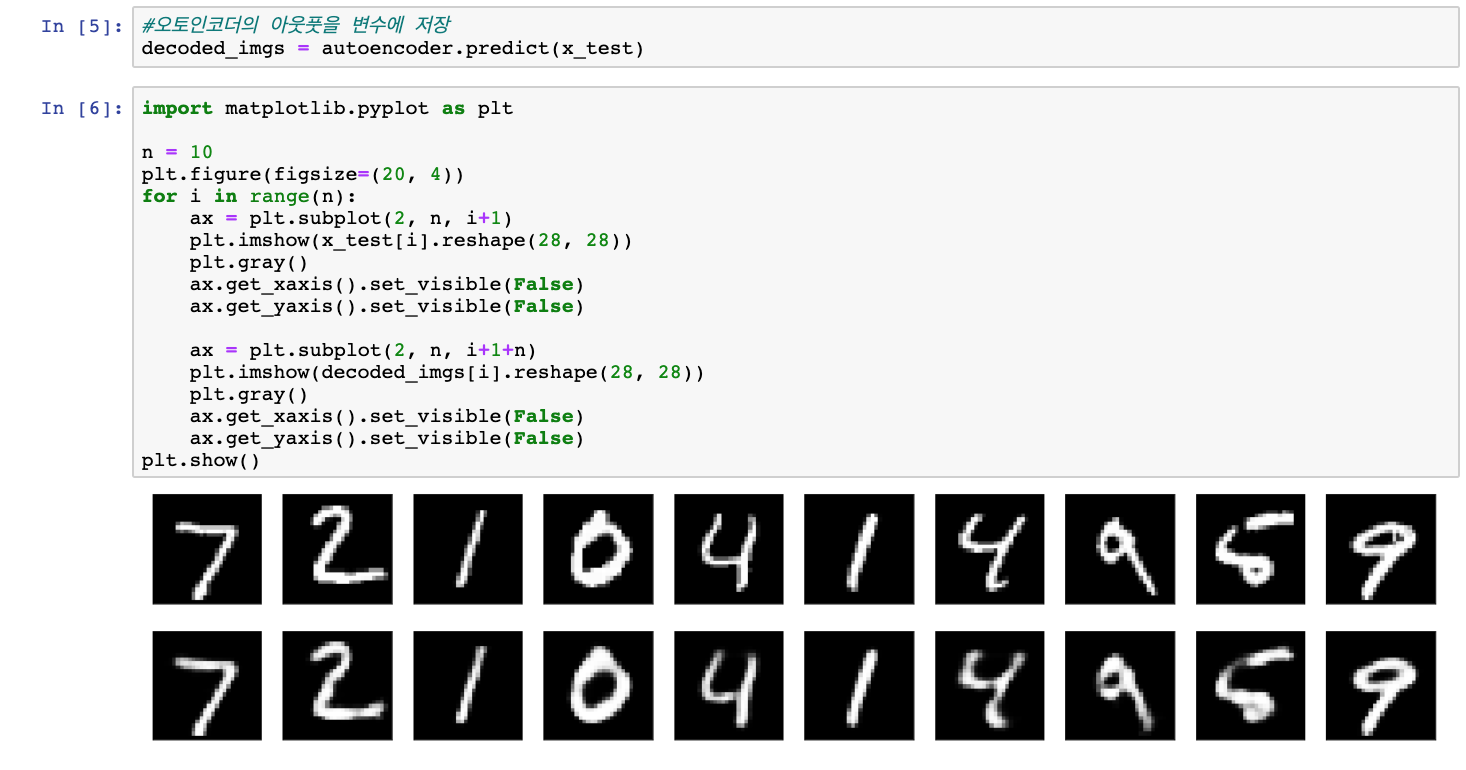
Report14

소프트웨어학과

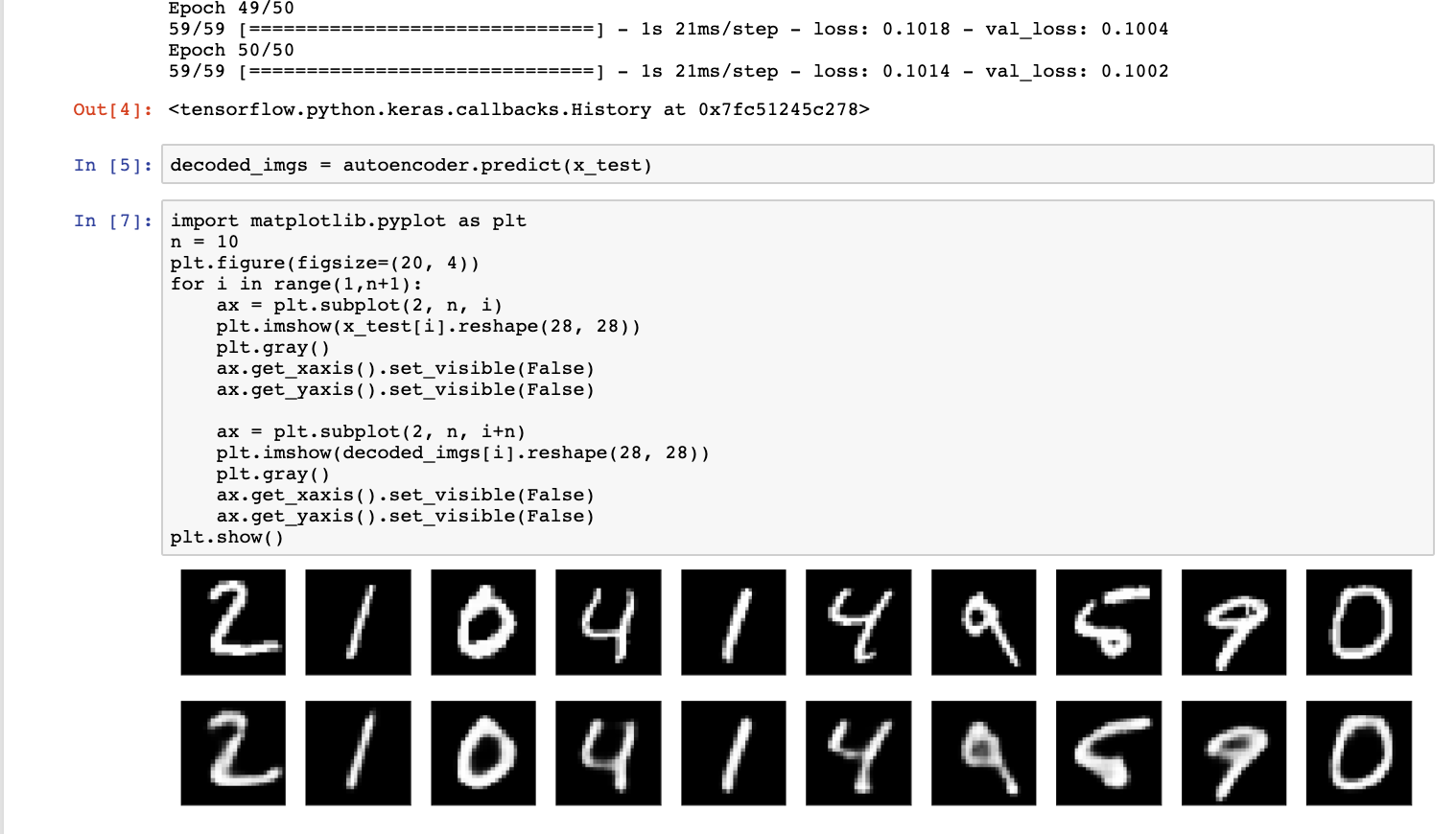
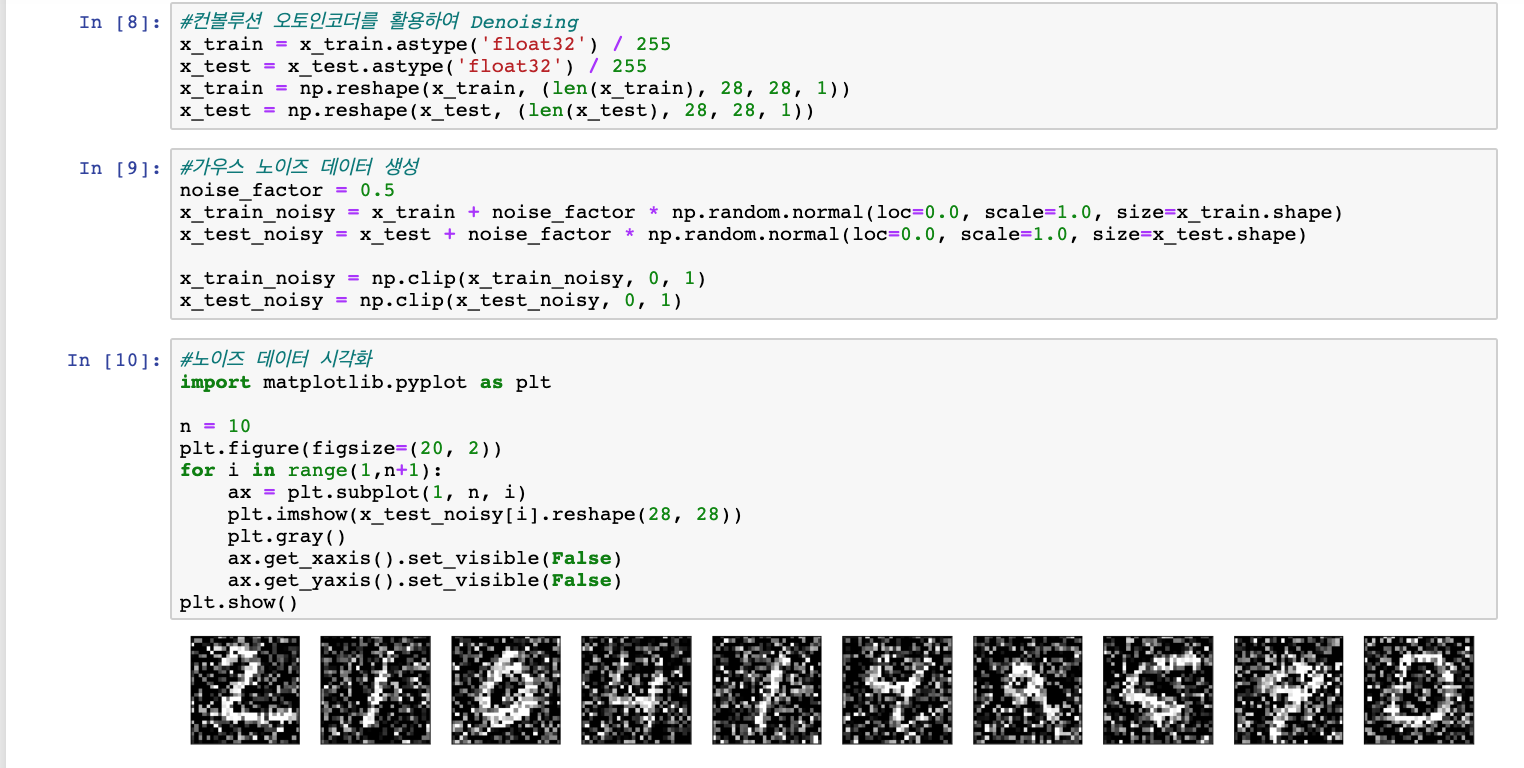
2016312568 정희윤

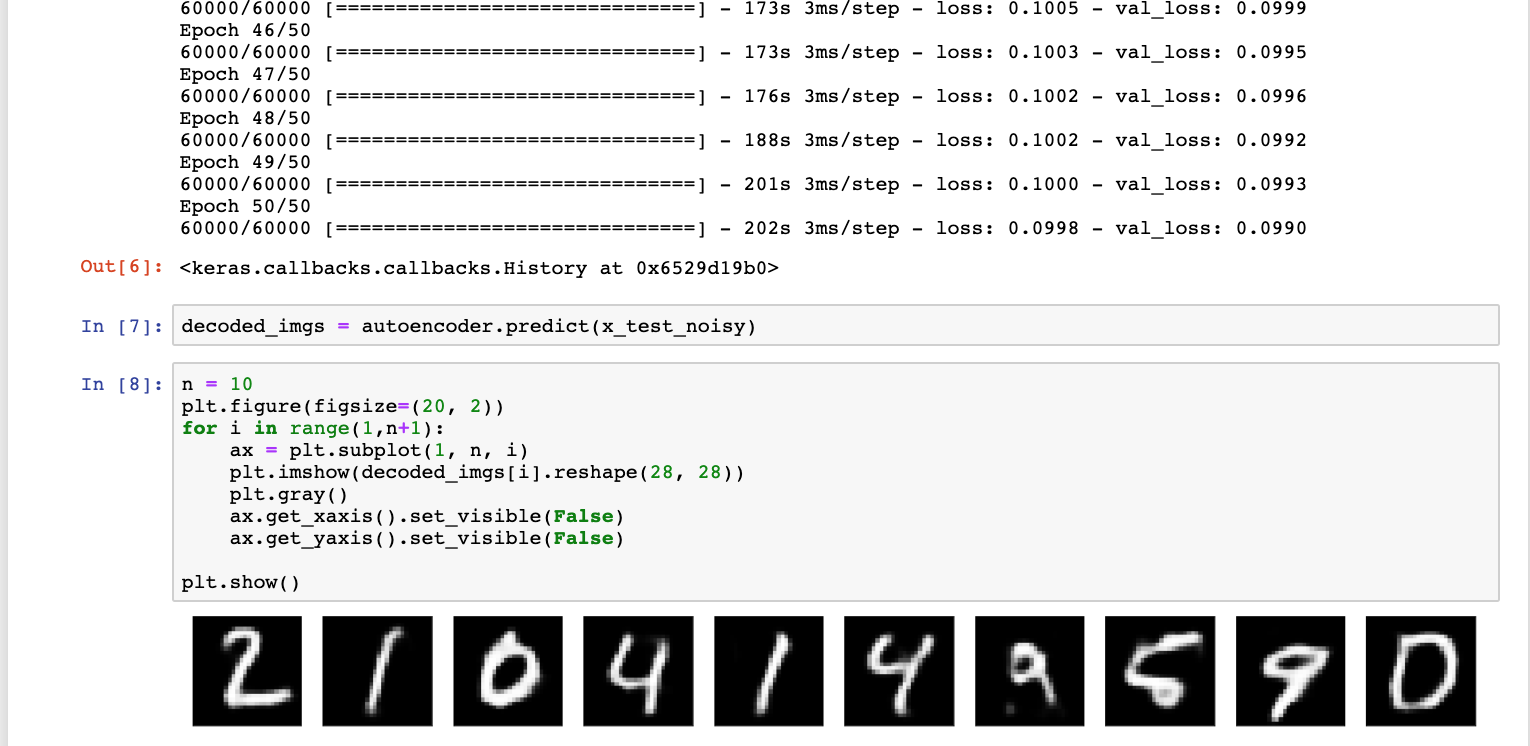
1. Deep Autoencoder

위 사진은 Deep Autoencoder의 Model 부분이다. Deep Autoencoder 또는 Stack Autoencoder이라고 부르는 이 형식은 여러개의 히든 레이어를 가지는 Auto Encoder이며 레이어를 추가할수록 Autoencoder가 더 복잡한 부호화를 학습할 수 있다. 위의 사진에서 볼 수 있다시피 Deep Autoencoder의 loss는 0.08정도로 실습 자료에 있는 기본적인 Autoencoder의 loss 값인 0.09보다 아주 조금 나아진 정도를 볼 수 있다.

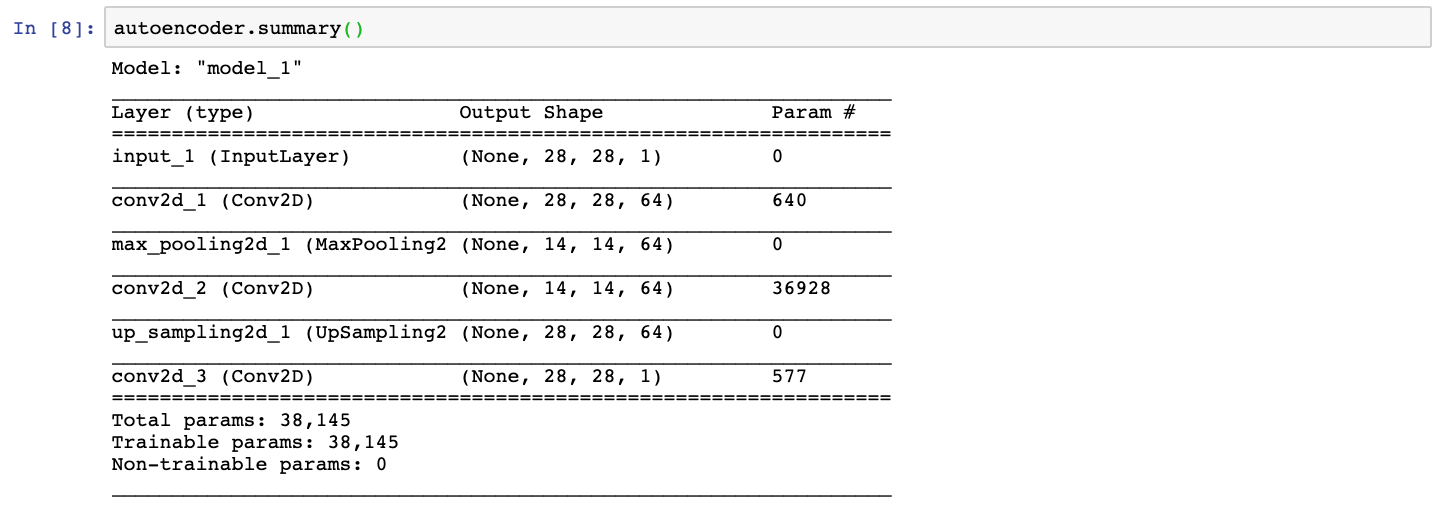
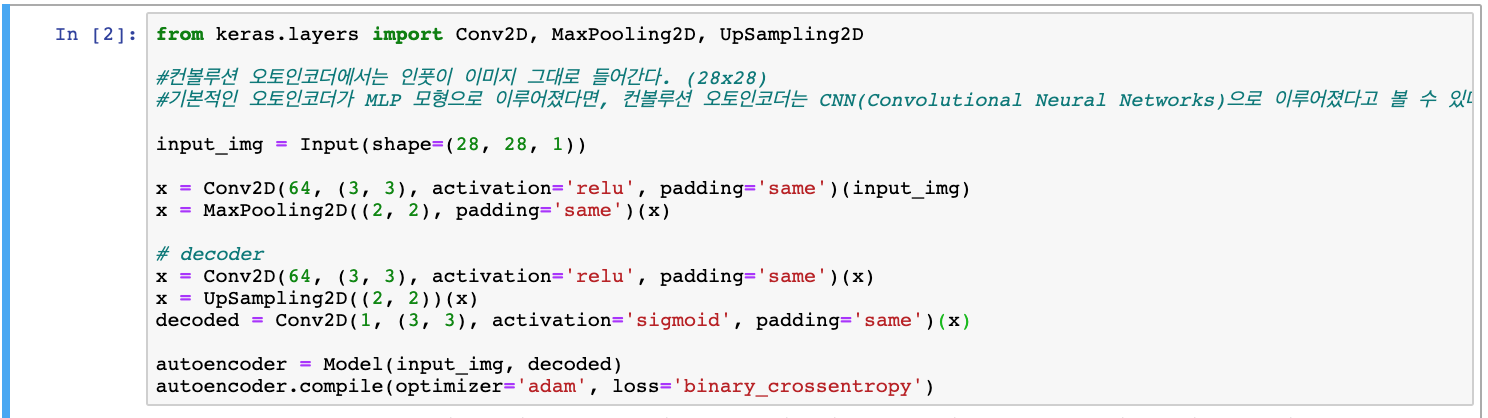
다음 사진은 Deep Autoencoder로 decode한 Mnist 이미지들이다. 첫줄이 원래 인풋, 두번째 줄이 decode된 이미지들이다. 기본적인 Autoencoder보다 조금 나아진 정도를 볼 수 있다. 이는 loss값이 조금 나아져서 그런 것이다

.

1. Convolutional Autoencoder  
   결국 Autoencoder도 이미지를 encoding, decoding하는 것이기에 CNN구조를 사용할 수 있다. 위과 같은 CNN구조로 Autoencoder를 구성하였을때 결과는 다음과 같다.  
   Loss는 0.10정도로 Deep Autoencoder보다는 조금 안좋은 loss값을 보이며 아래에 보이는 사진결과 또한 흐릿한것으로 손실양이 조금 많다.
2. Denoising Autoencoder  
   위 사진은 가우스 노이즈를 통한 Mnist 사진으로 노이즈 데이터를 만든 것이다. 위 노이즈 데이터를 decode하는 것이 목적이며 그를 위한 모델은 아래와 같다. 구조는 아래와 같으며 Activation은 relu, optimizer은 adam을 이용한 모습이다.

  
위 사진은 Convolutional Autoencoder로 Noised data를 Denoising한 모습이다. loss값은 0.1정도이며 denoise가 잘 된 모습이다.

1. CNN구조 바꿔보기  
   Convolutional Autoencoder로 진행할 것이며 구조는 다음과 같이 간단하게 바꾸어 보았다.



위와 같은 구조로 CNN모델을 바꾸어 보았으며 결과는 다음과 같다.  


놀랍게도 CNN구조를 간단하게 바꿔서 했을경우에 Loss값이 0.062정도로 많이 줄었으며 결과 숫자 사진에서 보아도 더 선명하여 손실이 적음을 알 수 있다.