FastCampus Pytorch

Ch9. Auto-Encoders

HARRY KIM

Lecture Content

1. Generative Model
2. Auto-Encoder
3. Convolutional Auto-Encoder
4. Variational Auto-Encoder
5. SVHN

Fast campus

- 강의 자료
  - Online
    - UVA DEEP LEARNING COURSE [University of Amsterdam, 2018]
    - Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.
    - Doersch, C. (2016). Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.
    - Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).
    - Theis, L., Oord, A. V. D., & Bethge, M. (2015). A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844.

Fast campus

1. Generative Model

Generative Model

Generative Model

Auto-Encoder

Convolutional Auto-Encoder

Variational Auto-Encoder

SVHN

- **Until Now…**
  - NN
    - 선형 회귀
    - MNIST 분류
  - CNN
    - MNIST 분류
    - CIFAR10 분류
    - 다람쥐/청설모 분류
  - RNN
    - 텍스트 생성
    - 가격 예측

Generative Model

- **Until Now…**
  - NN
    - 선형 **회귀**
    - MNIST **분류**
  - CNN
    - MNIST **분류**
    - CIFAR10 **분류**
    - 다람쥐/청설모 **분류**
  - RNN
    - 텍스트 생성
    - 가격 **예측**

Generative Model

- **Until Now…**
  - NN
    - 선형 **회귀**
    - MNIST **분류**
  - CNN
    - MNIST **분류**
    - CIFAR10 **분류**
    - 다람쥐/청설모 **분류**
  - RNN
    - 텍스트 생성
    - 가격 **예측**

Supervised Learning

Generative Model

- **Until Now…**

http://www.ohmynews.com/NWS_Web/View/at_pg.aspx?CNTN_CD=A0002234510

Generative Model

- From Now...

https://novakdjokovicfoundation.org/learn-to-decode-childrens-drawings/

Generative Model

- **From Now…**
  - 컴퓨터를 분류만이 아니라, 특정한 것을 생성하게 해보자.
  - 스스로 데이터를 생성해내는 것

Generative Model

Generative Model

Auto-Encoder

Convolutional Auto-Encoder

Variational Auto-Encoder

SVHN

- **From Now...**
  - 컴퓨터를 분류만이 아니라, 특정한 것을 생성하게 해보자.
  - 스스로 데이터를 생성해내는 것
  - Unsupervised Learning
    - = Label이 필요 없다.
    - = 무수히 많은 데이터를 사용 가능하다.

Generative Model

- **From Now…**
  - 컴퓨터를 분류만이 아니라, 특정한 것을 생성하게 해보자.
  - 스스로 데이터를 생성해내는 것
  - Unsupervised Learning
    - = Label이 필요 없다.
    - = 무수히 많은 데이터를 사용 가능하다.
  - 어떻게 생성해낼 것인가?

2. Auto-Encoder

Auto-Encoder

- 어떻게 그리는가?

Fast campus

Auto-Encoder

Generative Model

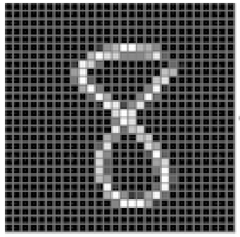
Auto-Encoder

Convolutional Auto-Encoder

Variational Auto-Encoder

SVHN

- 어떻게 그리는가?



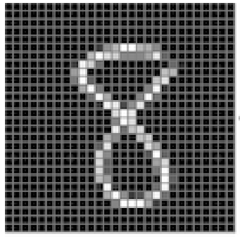
28 x 28
784 pixels

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	62	146	182	254	254	181	176	139	15	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	34	186	253	217	208	136	136	136	166	232	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	61	242	208	111	3	0	0	0	0	0	18	32	107	43	0	0	0	0	0	0
0	0	0	0	0	0	0	0	156	242	23	0	0	0	0	0	0	0	13	191	181	6	0	0	0	0	0	0
0	0	0	0	0	0	0	0	121	255	98	3	0	0	0	0	0	8	194	225	12	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	169	253	120	3	0	0	0	0	128	247	51	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	111	244	169	19	0	14	131	249	117	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	59	241	235	72	142	229	66	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	25	218	254	231	36	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	133	253	221	33	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	19	237	111	196	217	19	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	138	0	23	193	204	18	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	96	224	0	0	0	25	218	169	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	138	0	0	0	0	86	253	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	3	162	214	11	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	0	118	253	68	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	185	157	0	0	0	0	0	40	254	98	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	50	244	61	0	0	0	0	112	244	58	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	251	142	59	83	167	244	111	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	6	133	253	253	253	169	61	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

http://lucenaresearch.com/deep-neural-networks/

Auto-Encoder

- **어떻게 그리는가?**
  - 256^{28*28}?



28 x 28
784 pixels

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	62	146	182	254	254	181	176	139	15	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	34	186	253	217	208	136	136	136	166	232	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	61	242	208	111	3	0	0	0	0	0	18	32	107	43	0	0	0	0	0	0
0	0	0	0	0	0	0	0	156	242	23	0	0	0	0	0	0	0	13	191	181	6	0	0	0	0	0	0
0	0	0	0	0	0	0	0	121	255	98	3	0	0	0	0	0	8	194	225	12	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	169	253	120	3	0	0	0	0	128	247	51	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	111	244	169	19	0	14	131	249	117	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	59	241	235	72	142	229	66	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	25	218	254	231	36	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	133	253	221	33	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	19	237	111	196	217	19	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	138	0	23	193	204	18	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	96	224	0	0	0	25	218	169	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	138	0	0	0	0	86	253	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	3	162	214	11	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	0	118	253	68	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	185	157	0	0	0	0	0	40	254	98	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	50	244	61	0	0	0	0	112	244	58	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	251	142	59	83	167	244	111	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	6	133	253	253	253	169	61	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

http://lucenaresearch.com/deep-neural-networks/

Auto-Encoder

Generative Model

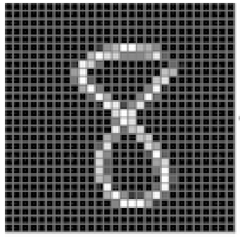
Auto-Encoder

Convolutional Auto-Encoder

Variational Auto-Encoder

SVHN

- **어떻게 그리는가?**
  - 256^{28*28}? **NO!**



28 x 28
784 pixels

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	62	146	182	254	254	181	176	139	15	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	34	186	253	217	208	136	136	136	166	232	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	61	242	208	111	3	0	0	0	0	0	18	32	107	43	0	0	0	0	0	0
0	0	0	0	0	0	0	0	156	242	23	0	0	0	0	0	0	0	13	191	181	6	0	0	0	0	0	0
0	0	0	0	0	0	0	0	121	255	98	3	0	0	0	0	0	8	194	225	12	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	169	253	120	3	0	0	0	0	128	247	51	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	111	244	169	19	0	14	131	249	117	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	59	241	235	72	142	229	66	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	25	218	254	231	36	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	133	253	221	33	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	19	237	111	196	217	19	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	138	0	23	193	204	18	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	96	224	0	0	0	25	218	169	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	138	0	0	0	0	86	253	99	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	3	162	214	11	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	215	97	0	0	0	0	0	118	253	68	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	185	157	0	0	0	0	0	40	254	98	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	50	244	61	0	0	0	0	112	244	58	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	174	251	142	59	83	167	244	111	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	6	133	253	253	253	169	61	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

http://lucenaresearch.com/deep-neural-networks/

Auto-Encoder

- Generative Model
- Auto-Encoder
- Convolutional Auto-Encoder
- Variational Auto-Encoder
- SVHN

어떻게 그리는가?

- 고양이?
  - 전체적인 모양
  - 눈/코/입/귀
  - 색깔
  - 꼬리
  - …

Auto-Encoder

Generative Model

Auto-Encoder

Convolutional Auto-Encoder

Variational Auto-Encoder

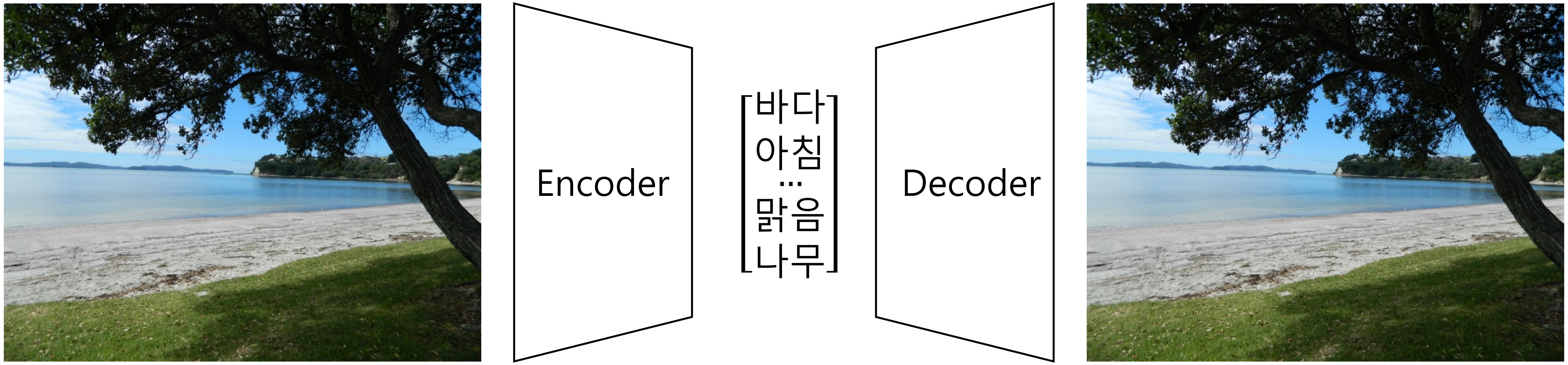
SVHN

- **어떻게 그리는가?**
  - 풍경?
    - 바다/산
    - 아침/점심/저녁/새벽
    - 맑음/흐림/비
    - 사람/나무/집
    - …

Fast campus

Auto-Encoder

- **어떻게 그리는가?**
  - 일반적으로 특징에 대해 먼저 생각하고, 그림을 그리게 됨
  - = 특징을 통해 그림을 그릴 수 있다
  - = 반대로, 그림에서 특징을 뽑아낼 수 있다



Auto-Encoder

- **어떻게 생성하는가?**

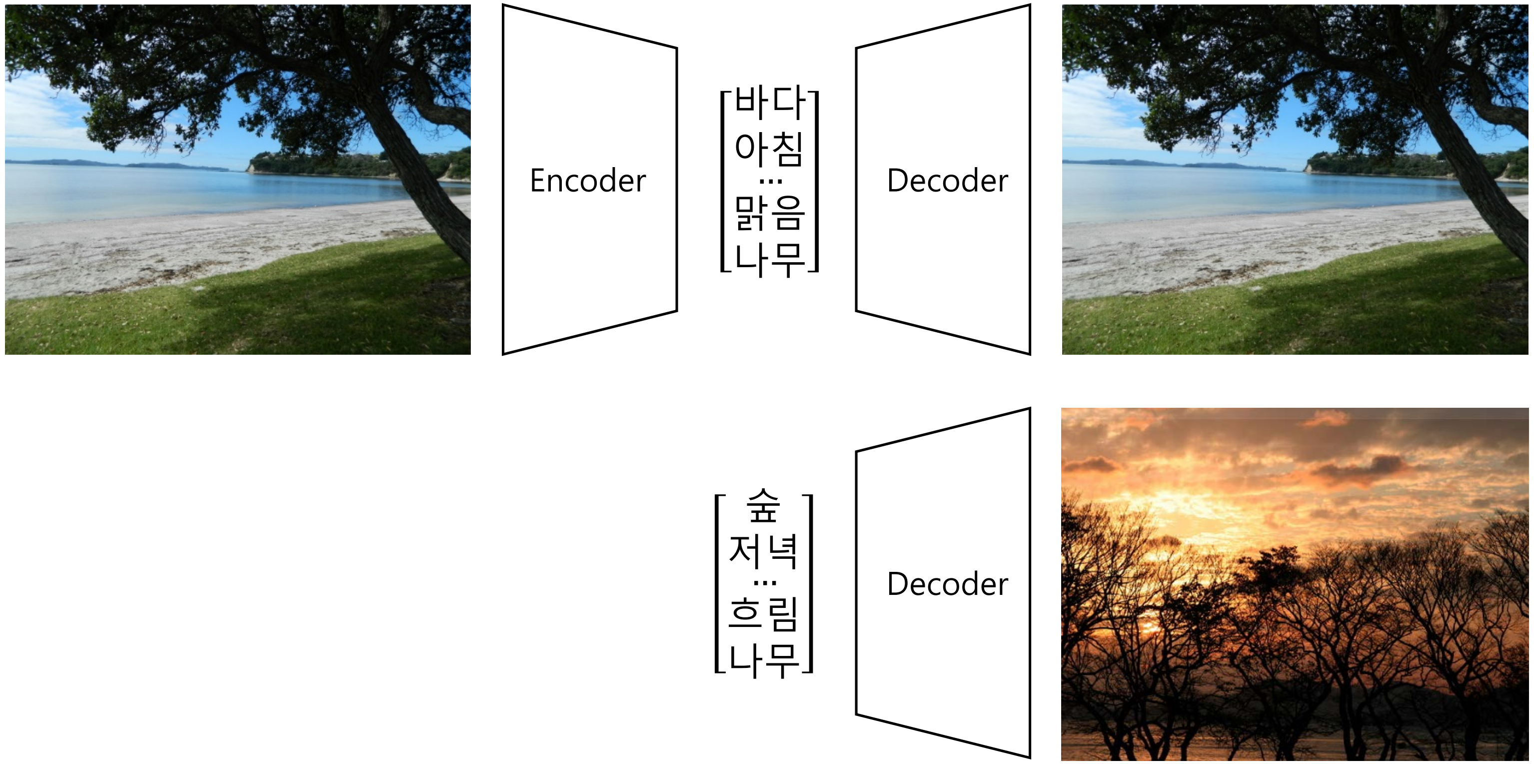
데이터를 통해 **특징**을 **추출**해낼 수 있다!

=

특징을 통해 **데이터**를 **재구성**할 수 있다!

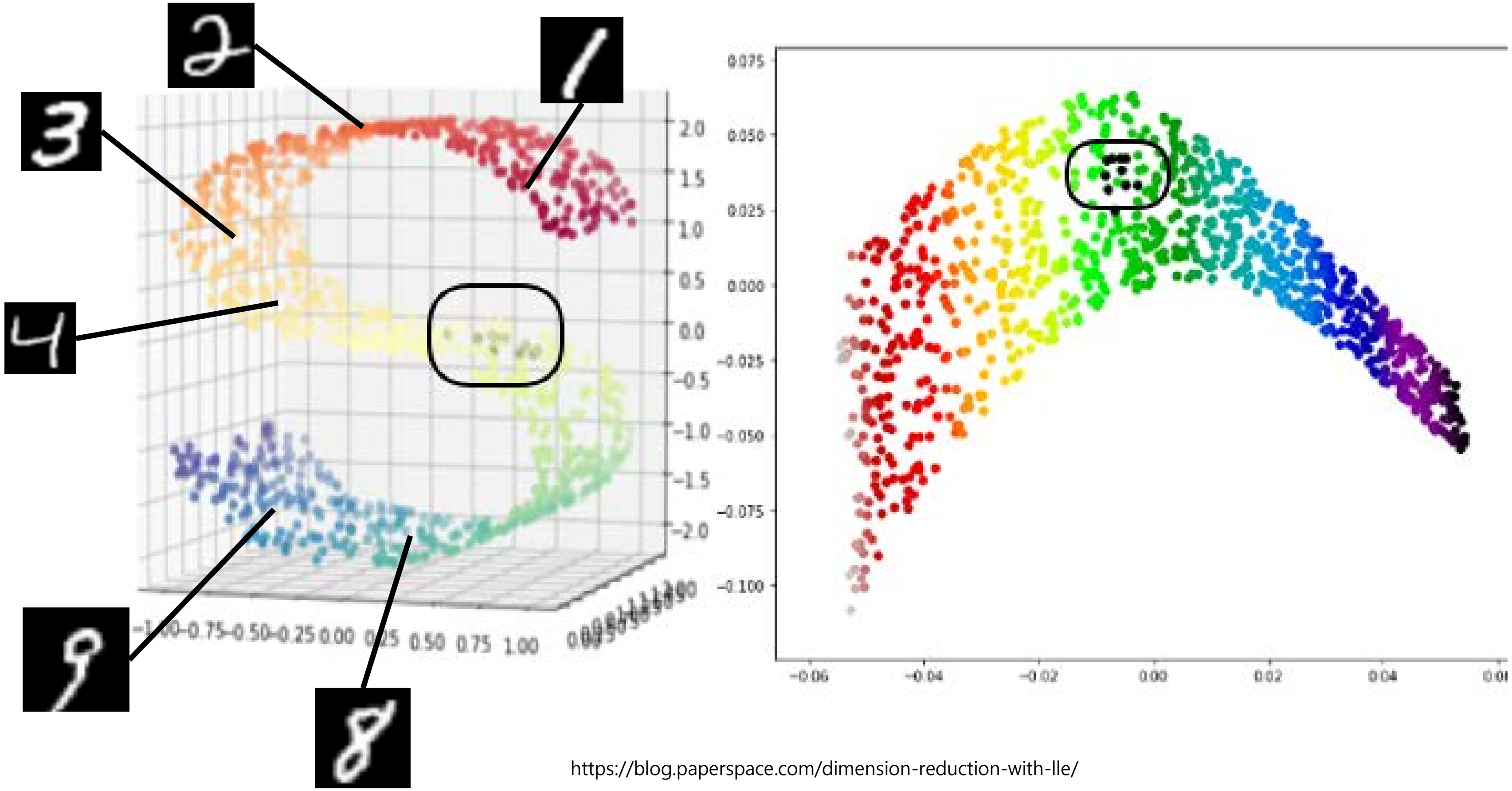
Auto-Encoder

- **어떻게 생성하는가?**
  - 그렇다면, **원래 데이터(Data)**의 **특징 벡터(Latent Vector)**를 추출하고 다시 **특징 벡터(Latent Vector)**로 **데이터(Data)**를 만들도록 학습시키자
  - 그 후 다른 특징 벡터(Latent Vector)을 입력하면 새로운 그림을 생성해낼 수 있을 것이다!



Auto-Encoder

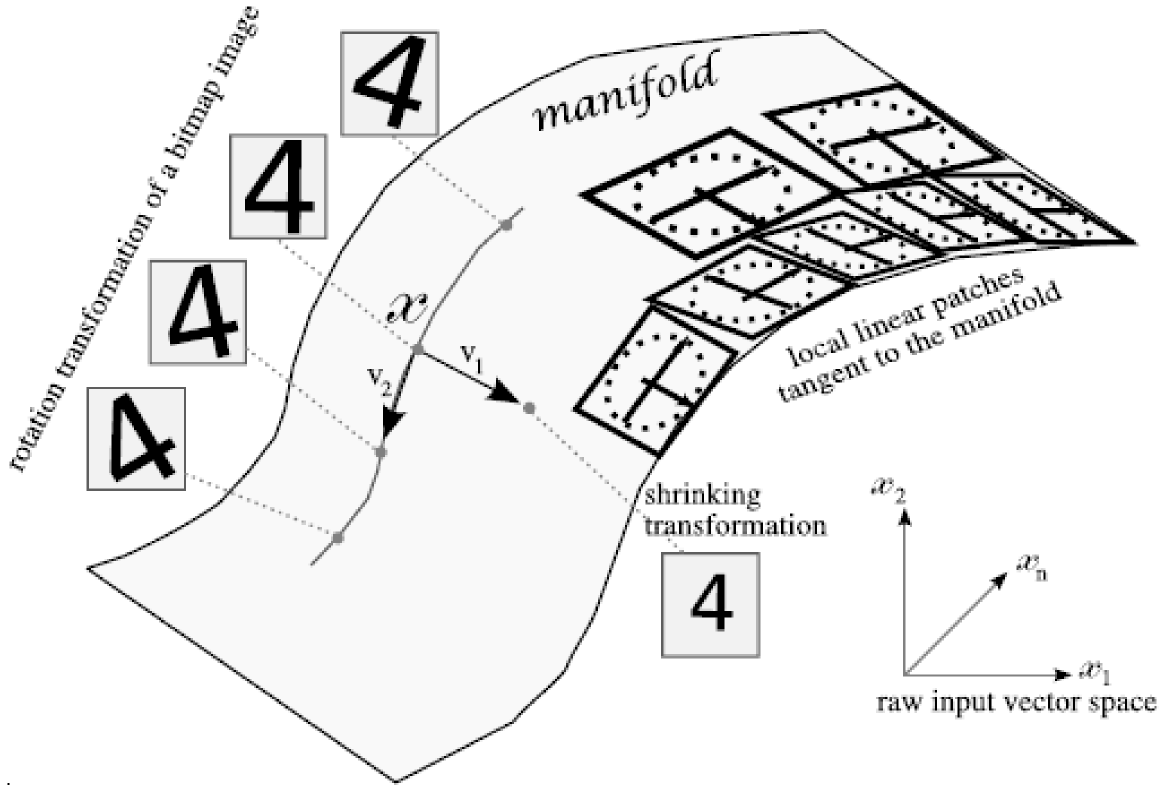
- **특징 벡터(Latent Vector)에 대한 고찰**
  - 이미지가 모든 공간(256^{28*28})을 사용하지는 않을 것
  - 서로 비슷한 이미지는 서로 연결이 되어있을 것이다



https://blog.paperspace.com/dimension-reduction-with-lle/

Auto-Encoder

- **특징 벡터(Latent Vector)에 대한 고찰**
  - 이미지가 분포되어 있는 실질적인 공간 = Manifold
    - 수학적으로는, 고차원 공간 중에 존재하는 실질적으로는 보다 저차원으로 표시 가능한 도형을 Manifold라 한다.



https://calculatedcontent.com/2015/04/01/why-deep-learning-works-ii-the-renormalization-group/

Auto-Encoder

Generative Model

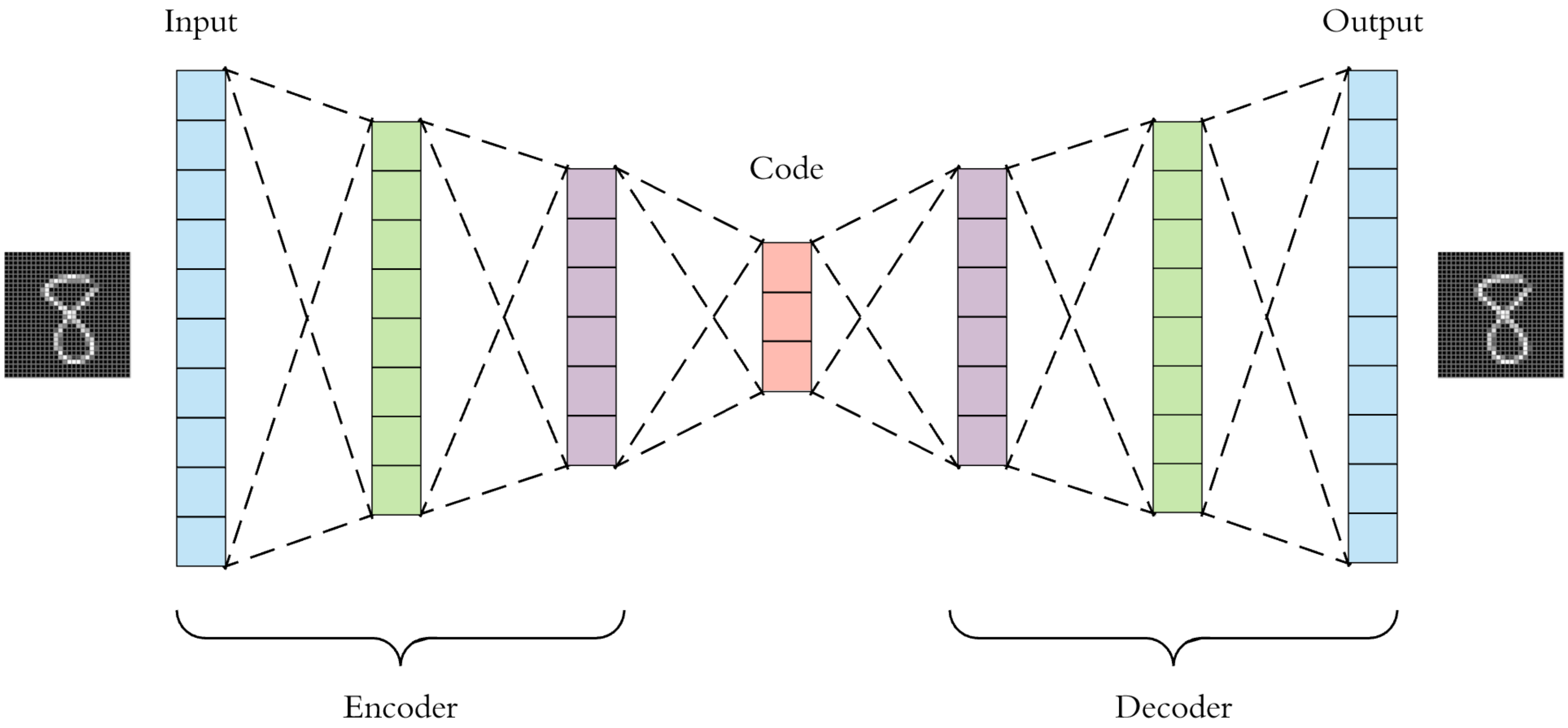
Auto-Encoder

Convolutional Auto-Encoder

Variational Auto-Encoder

SVHN

- **Standard Auto-Encoder(AE)**
  - Input(x)가 Layer를 거쳐 동일한 Output(x)로 나오도록 학습

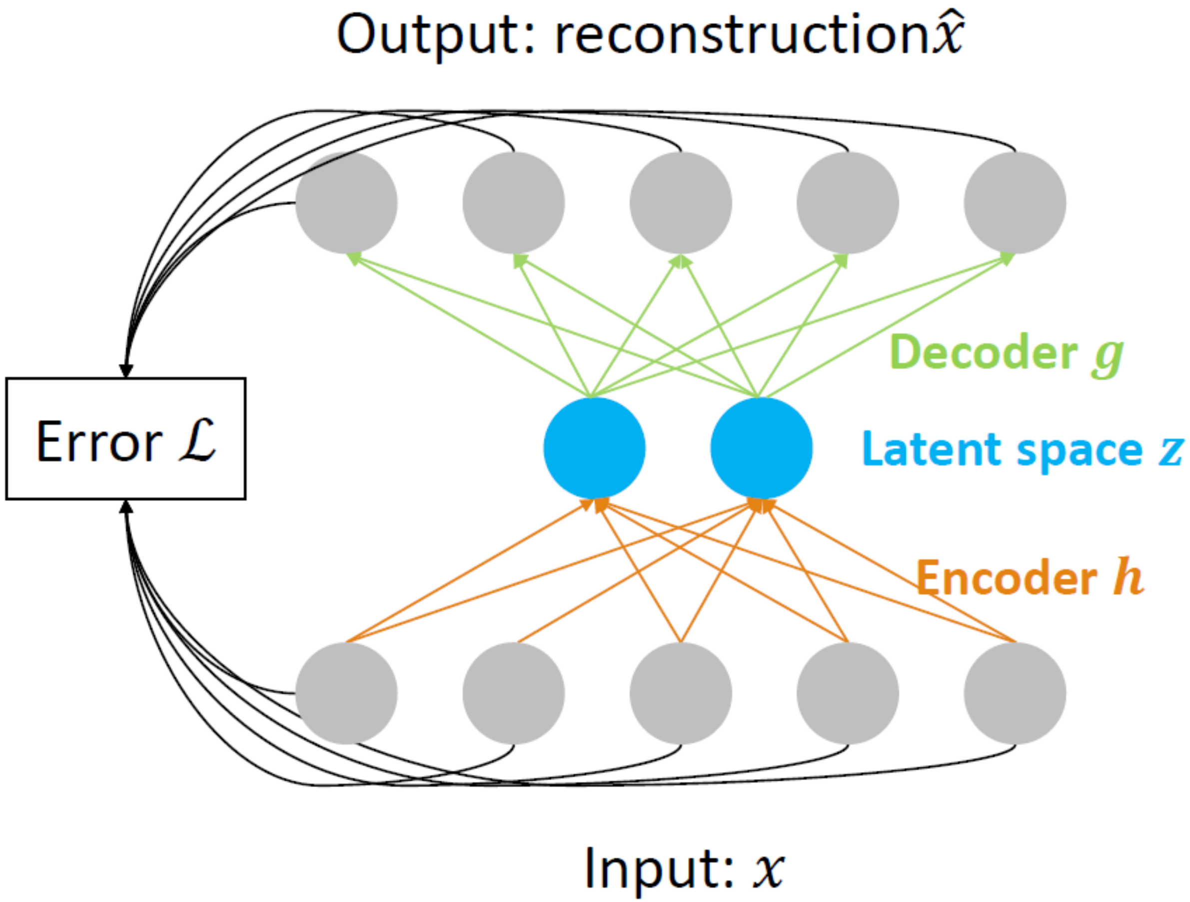


https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Auto-Encoder

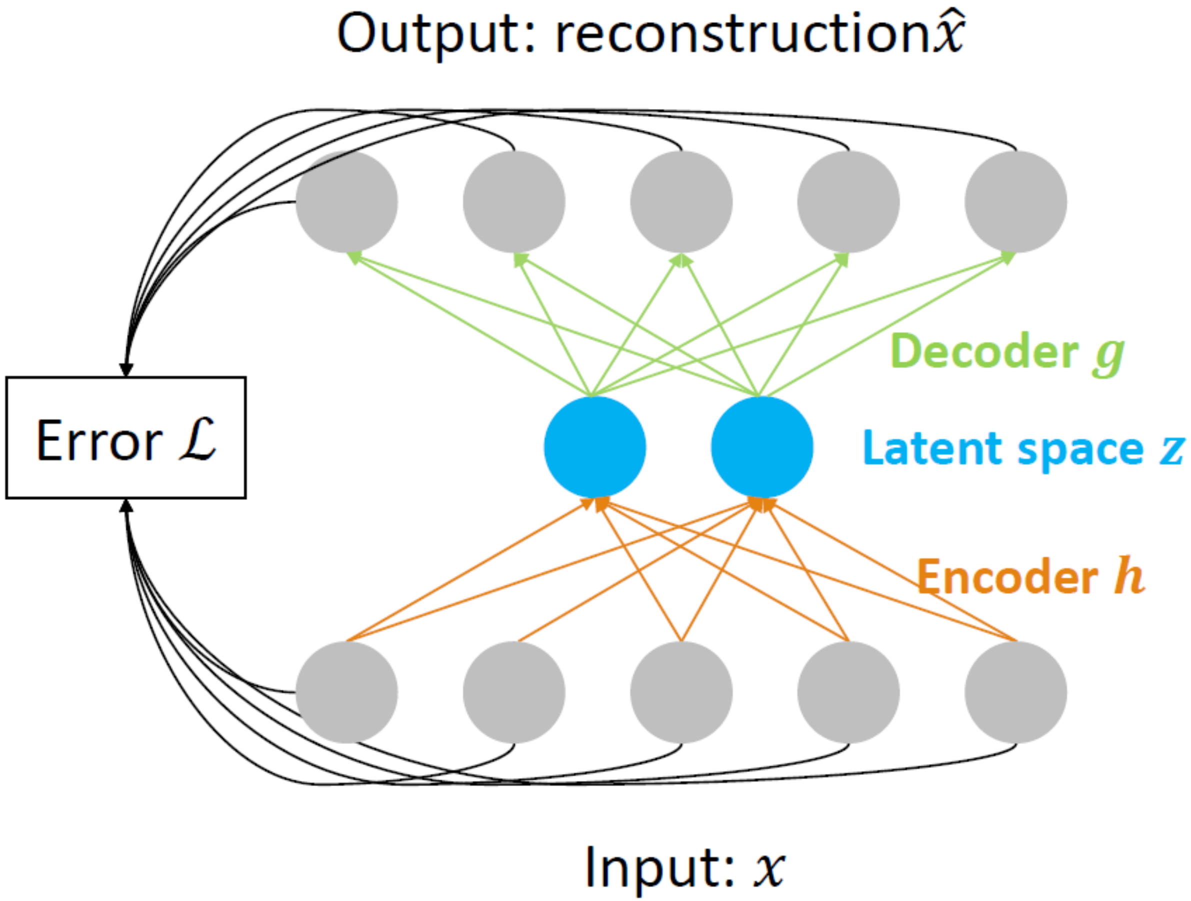
Generative Model
Auto-Encoder
Convolutional Auto-Encoder
Variational Auto-Encoder
SVHN

- **Standard Auto-Encoder(AE)**
  - Latent Space
  - $z = h(x)$
  - Reconstructed Data
  - $\hat{x} = g(h(x))$
  - Reconstruction Error
  - $\mathcal{L} = \sum_d \ell(x, \hat{x})$
  - ex) $\ell = |x - \hat{x}|^2$



Auto-Encoder

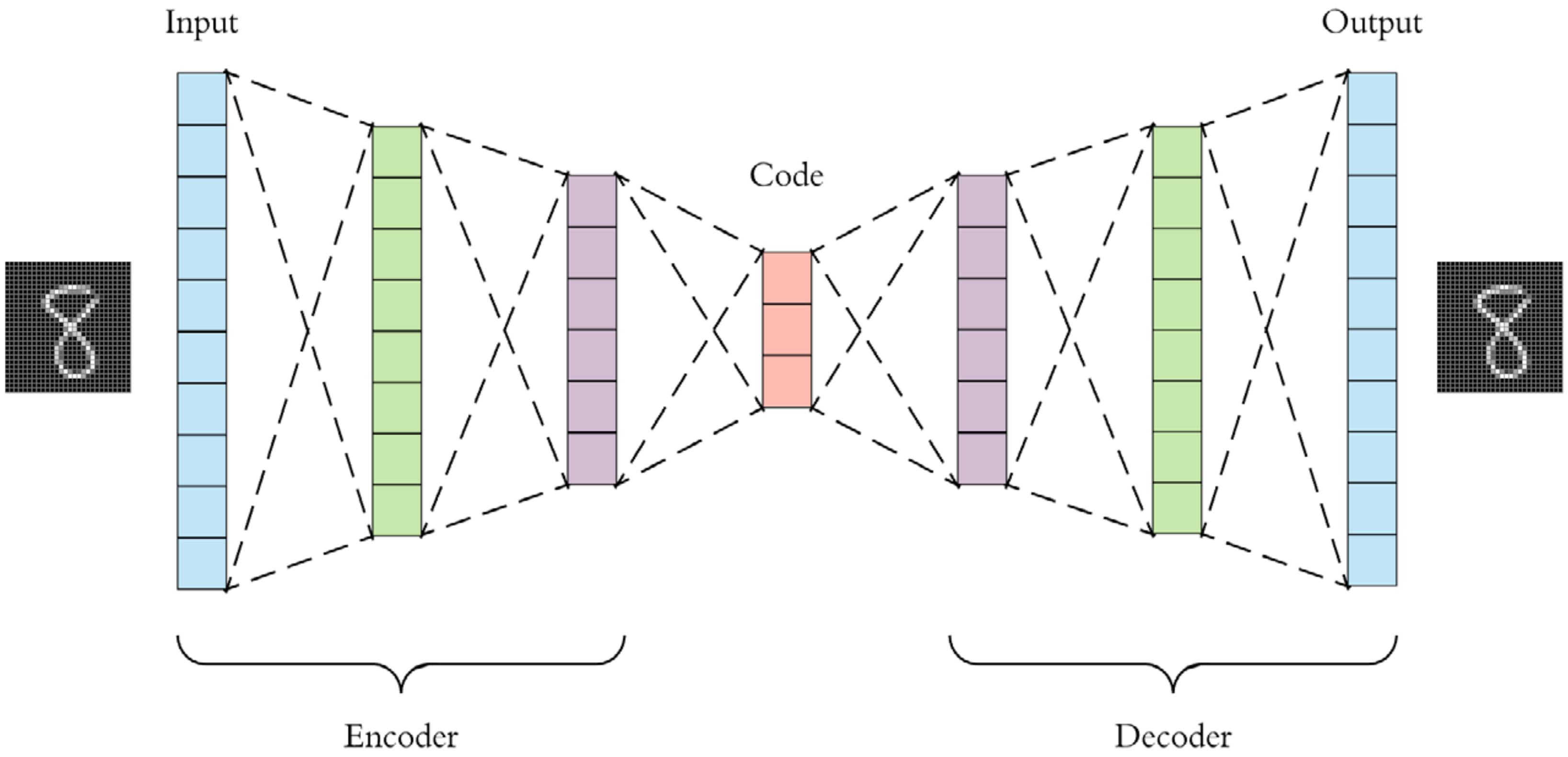
- **Standard Auto-Encoder(AE)**
  - 물론, z는 x보다 적은 차원을 가져야 함
    - Latent Space의 의미 부여
    - Identity Model로의 수렴 방지
    - cf) Overcomplete AE
  - 이러한 알고리즘은 PCA와 유사



3. Convolutional Auto-Encoder

Convolutional Auto-Encoder

- **NN v.s. CNN**
  - 데이터가 그림이라면?
  - NN을 사용하는 것보다, CNN을 사용하는 것이 더 좋지 않을까?

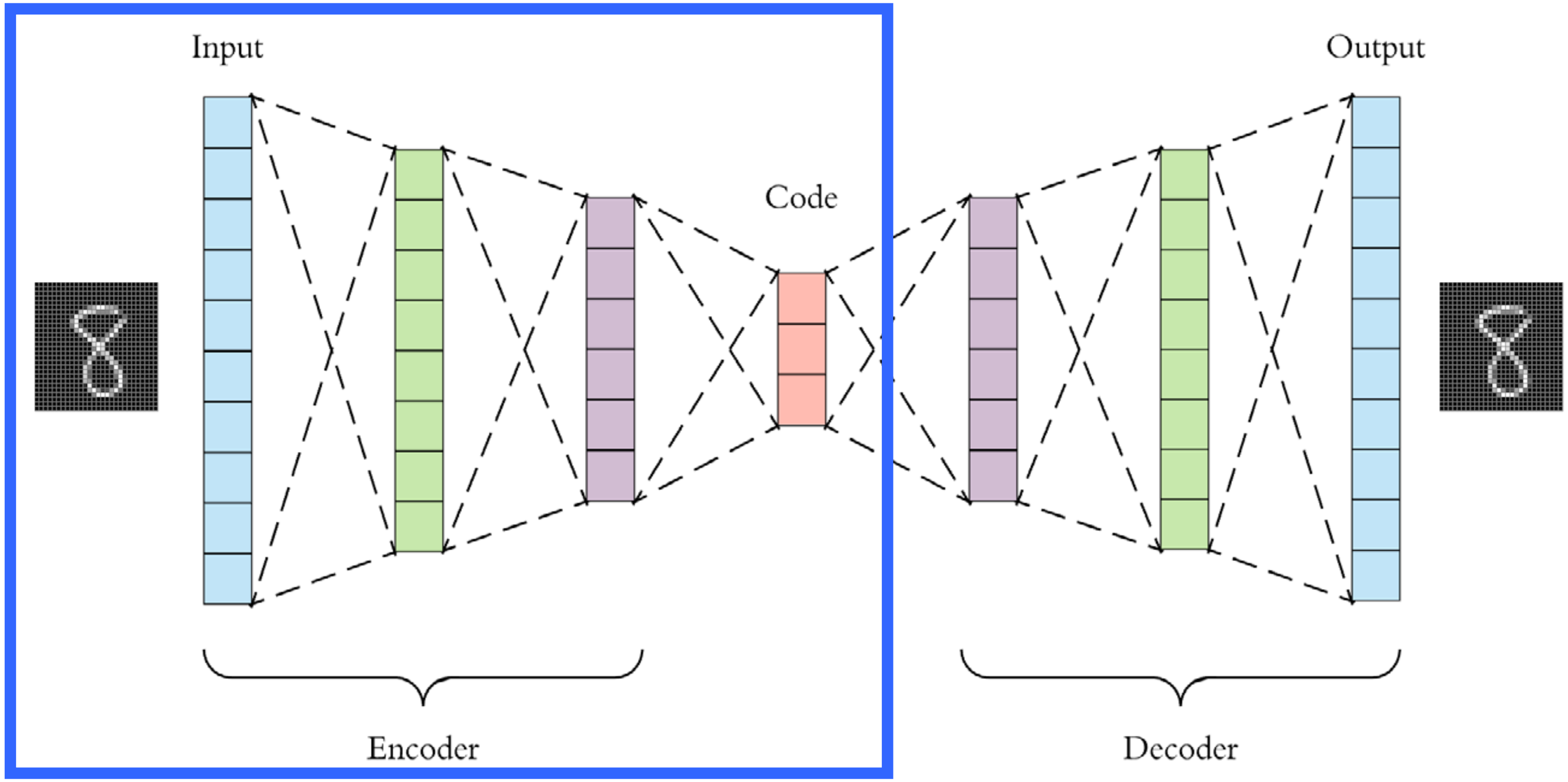


https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Fast campus

Convolutional Auto-Encoder

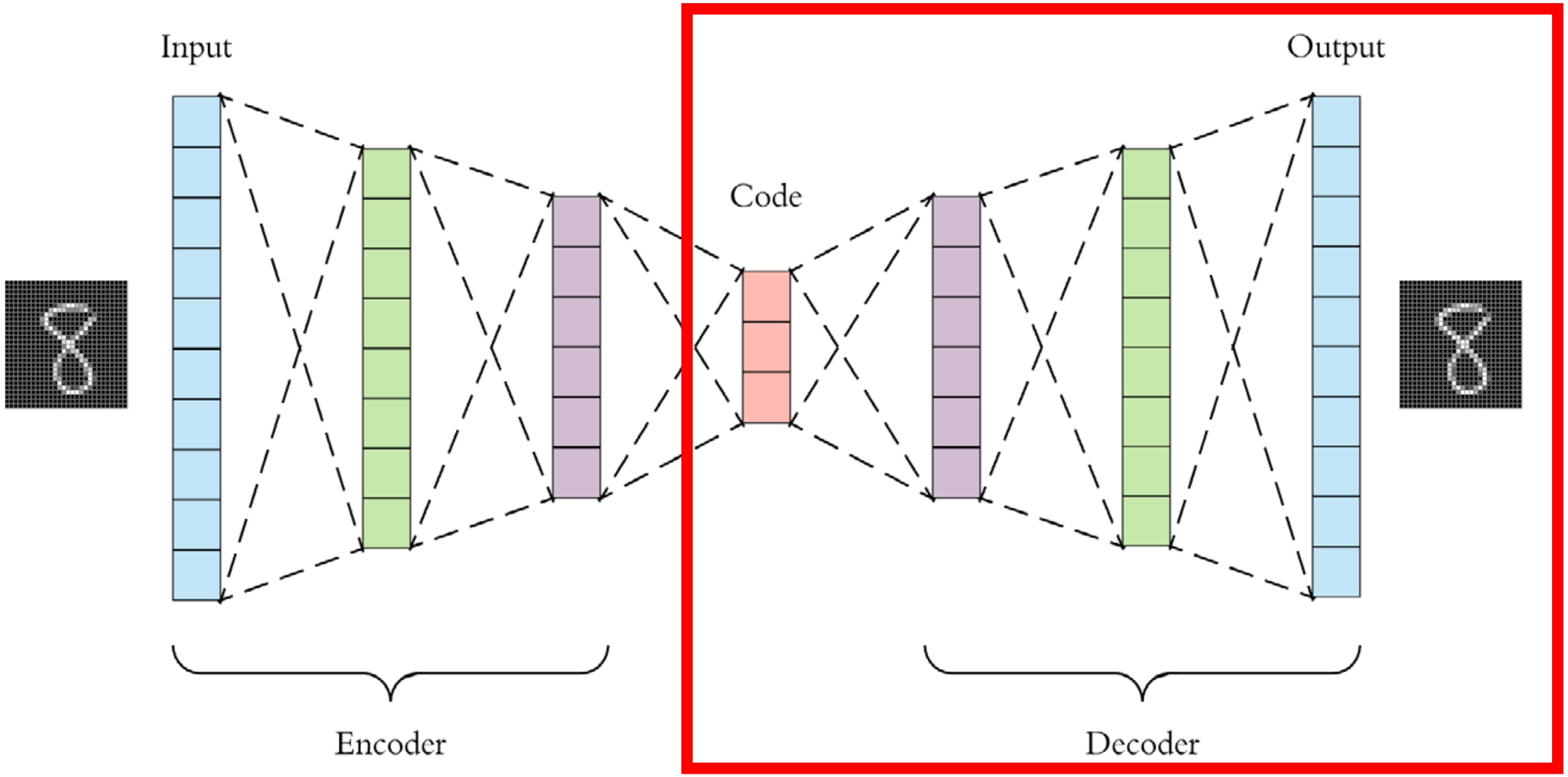
- **NN v.s. CNN**
  - 데이터가 그림이라면?
  - NN을 사용하는 것보다, CNN을 사용하는 것이 더 좋지 않을까?



https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Convolutional Auto-Encoder

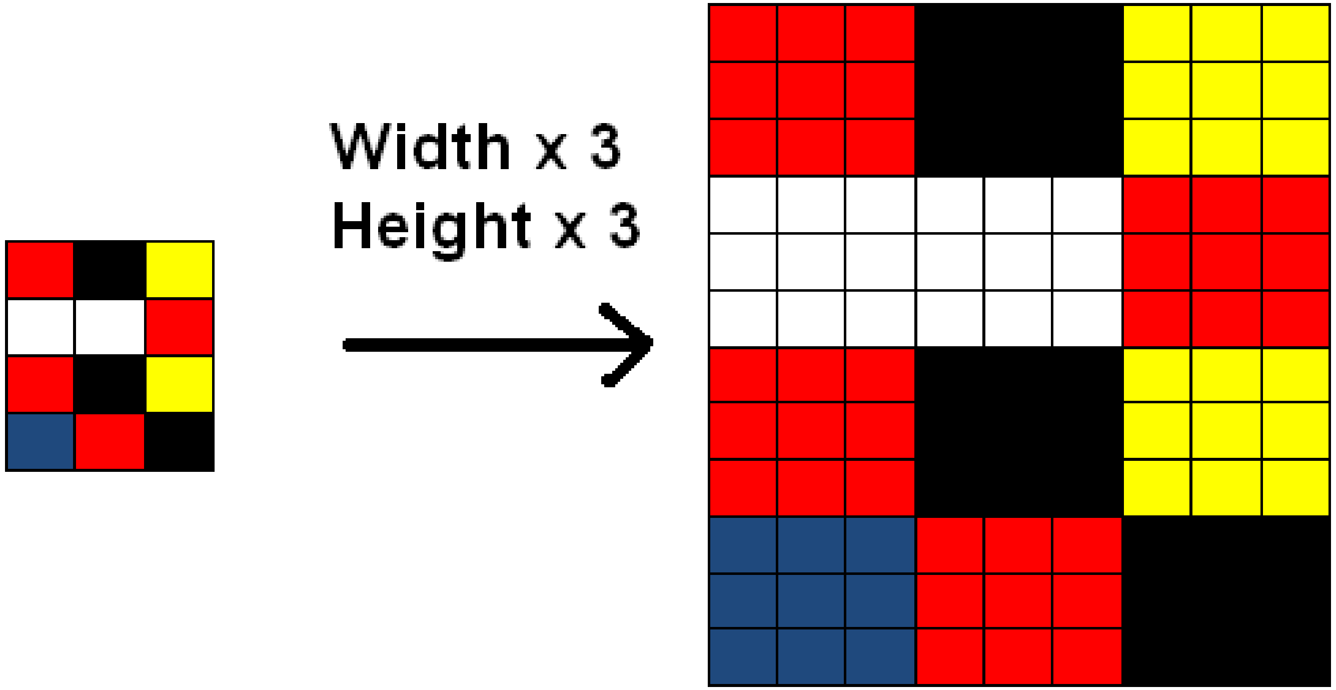
- **NN v.s. CNN**
  - 데이터가 그림이라면?
  - NN을 사용하는 것보다, CNN을 사용하는 것이 더 좋지 않을까?



https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Convolutional Auto-Encoder

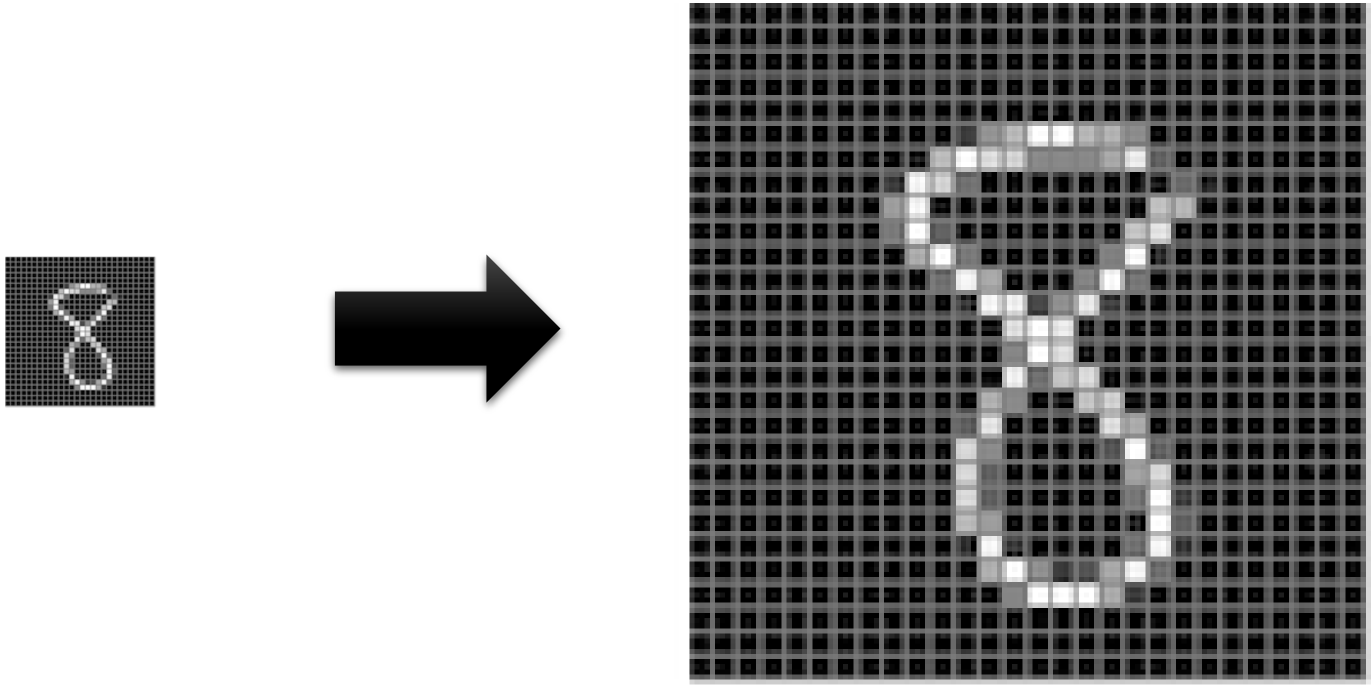
- **Up-sampling**
  - 적은 데이터로 큰 데이터를 생성해내는 것
  - 가장 단순한 방법 : 보간법(Interpolation)



https://www.giassa.net/?page_id=200

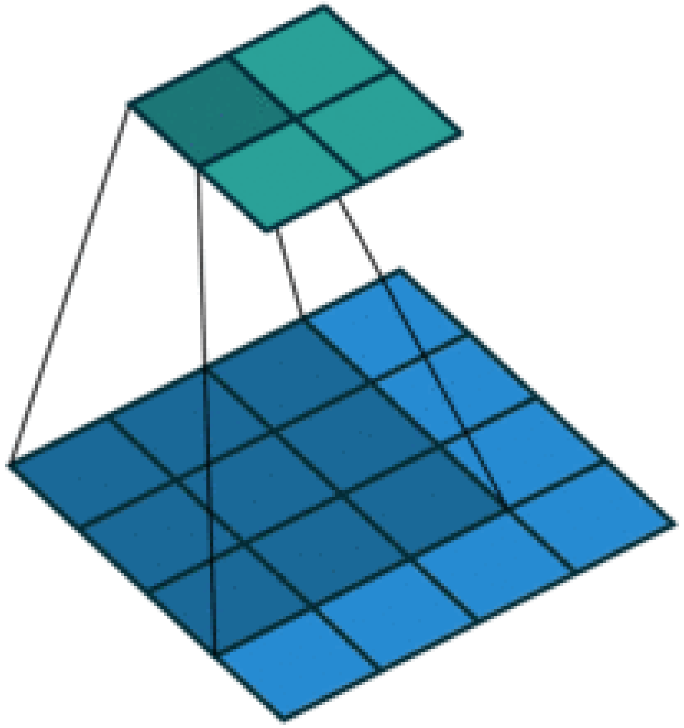
Convolutional Auto-Encoder

- **Up-sampling**
  - 단순 보간법의 문제점
    - 벡터를 확대한 것에 불과 (특징으로 그림을 만들어 내는 것이 아님)
    - 또한 그림의 경우 픽셀 깨짐 현상 발생



Convolutional Auto-Encoder

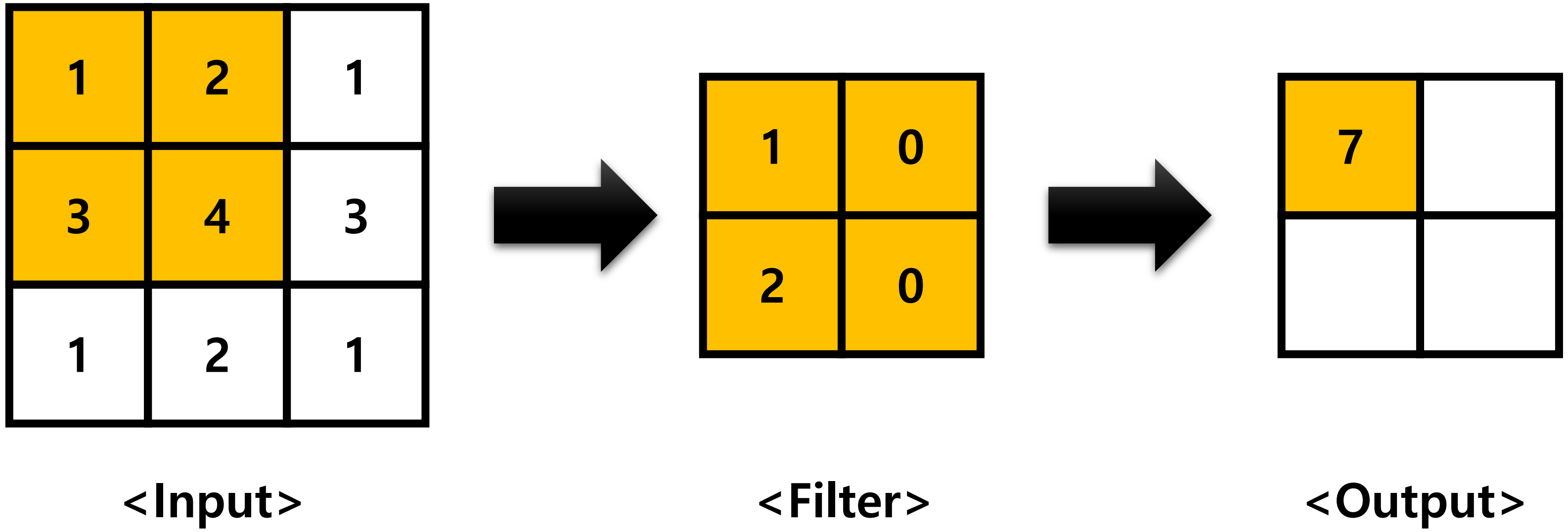
- **Transposed Convolution (Deconvolution, Upconvolution)**
  - Convolution을 역연산하자



http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

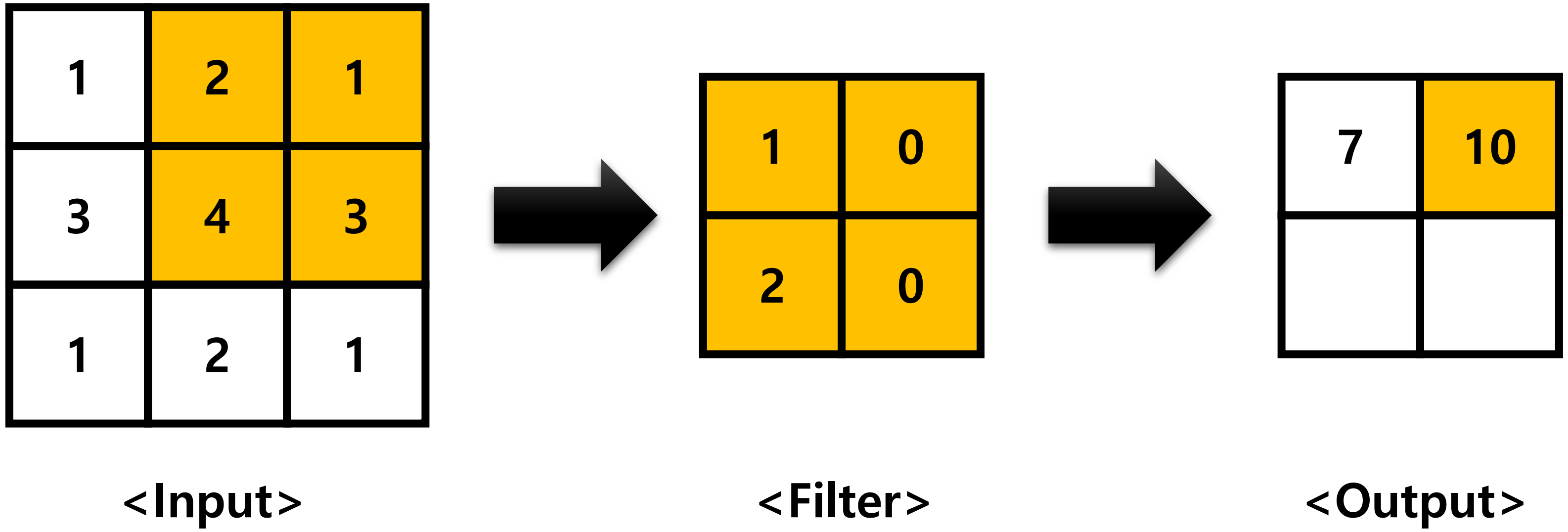
Convolutional Auto-Encoder

- Transposed Convolution



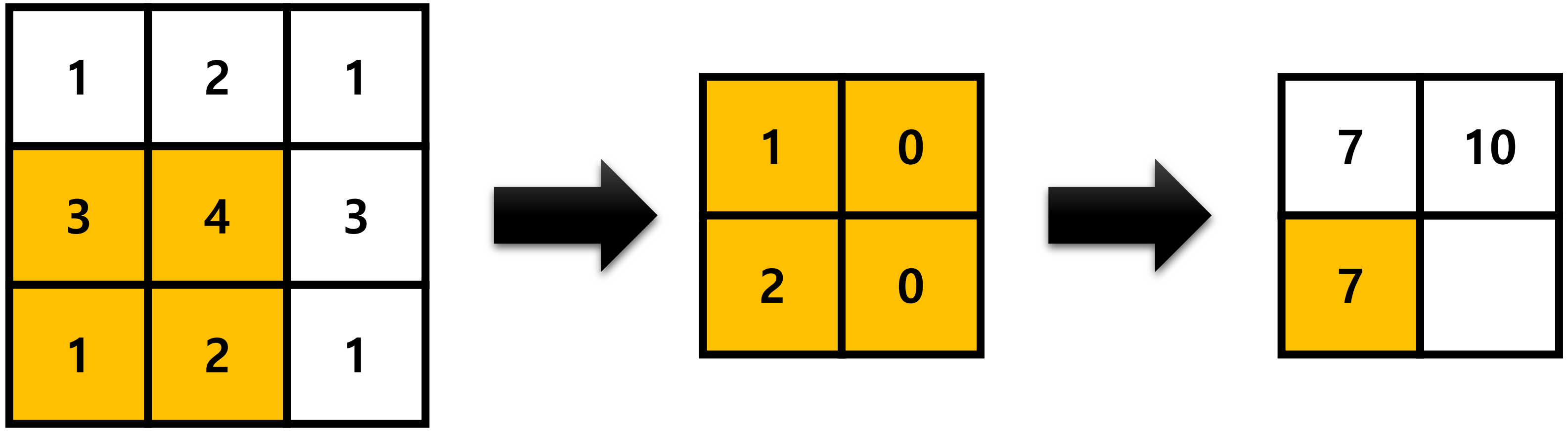
Convolutional Auto-Encoder

- Transposed Convolution



Convolutional Auto-Encoder

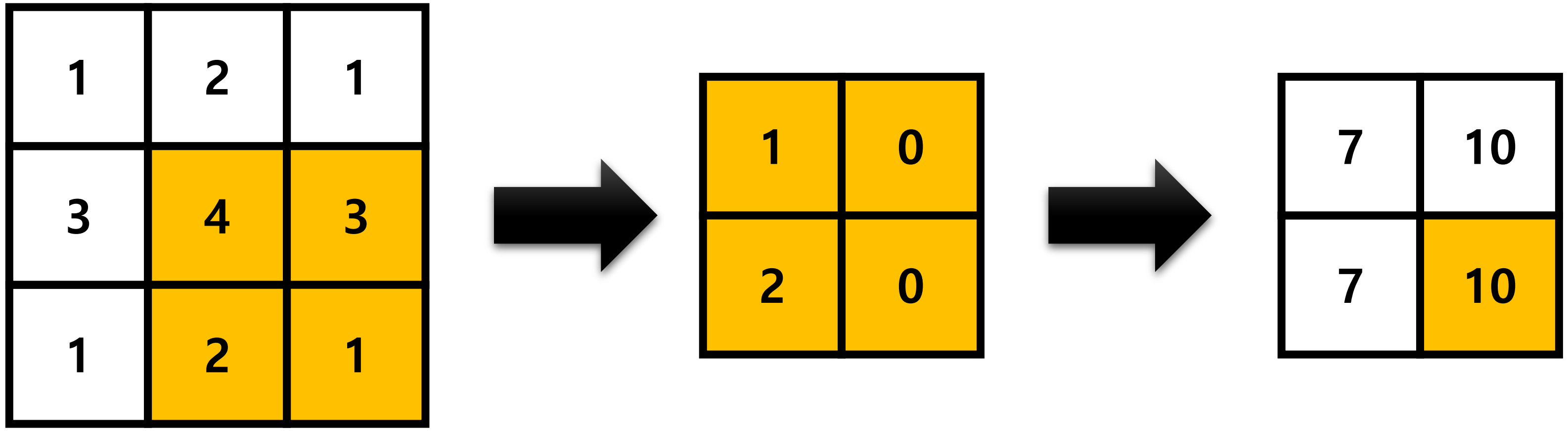
- Transposed Convolution



<Input> <Filter> <Output>

Convolutional Auto-Encoder

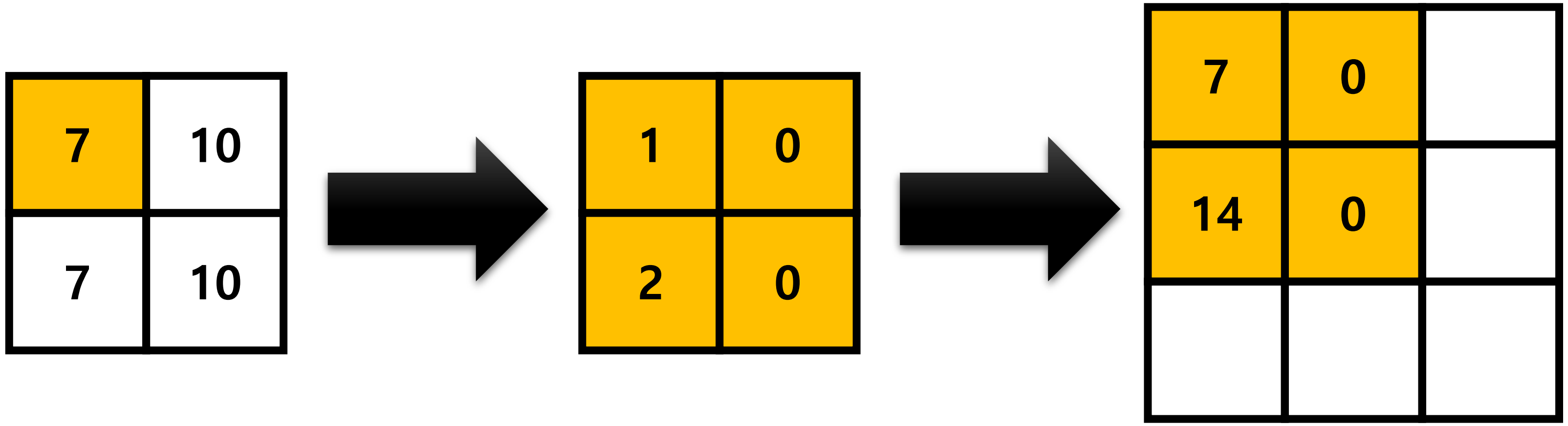
- **Transposed Convolution**



<Input> <Filter> <Output>

Convolutional Auto-Encoder

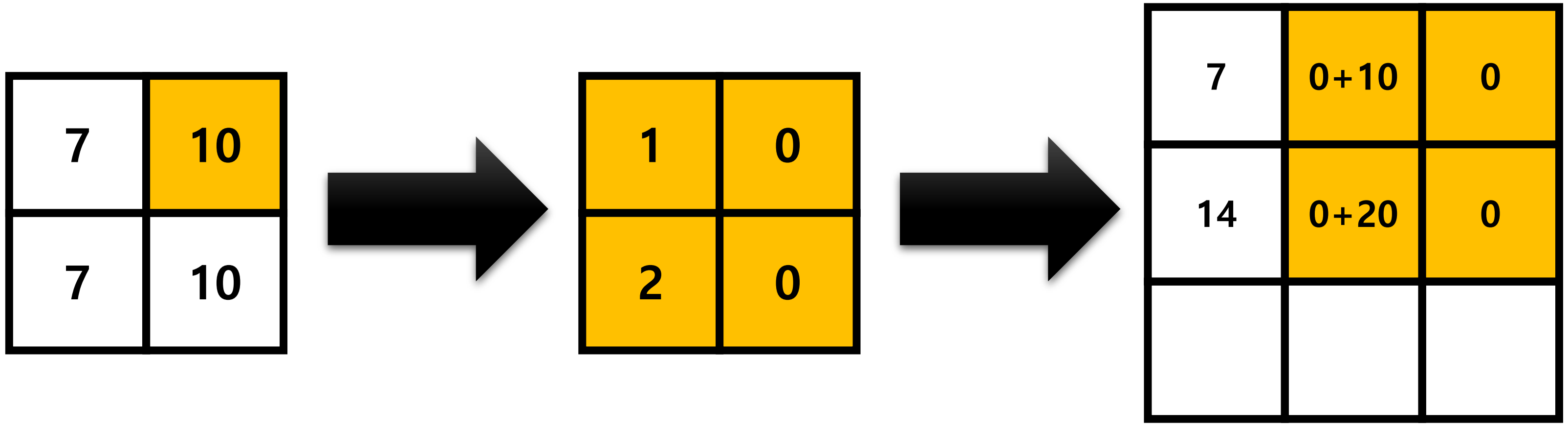
- Transposed Convolution



<Input> <Filter> <Output>

Convolutional Auto-Encoder

- **Transposed Convolution**



<Input> <Filter> <Output>

Convolutional Auto-Encoder

- Transposed Convolution

7	10
7	10

<Input>

1	0
2	0

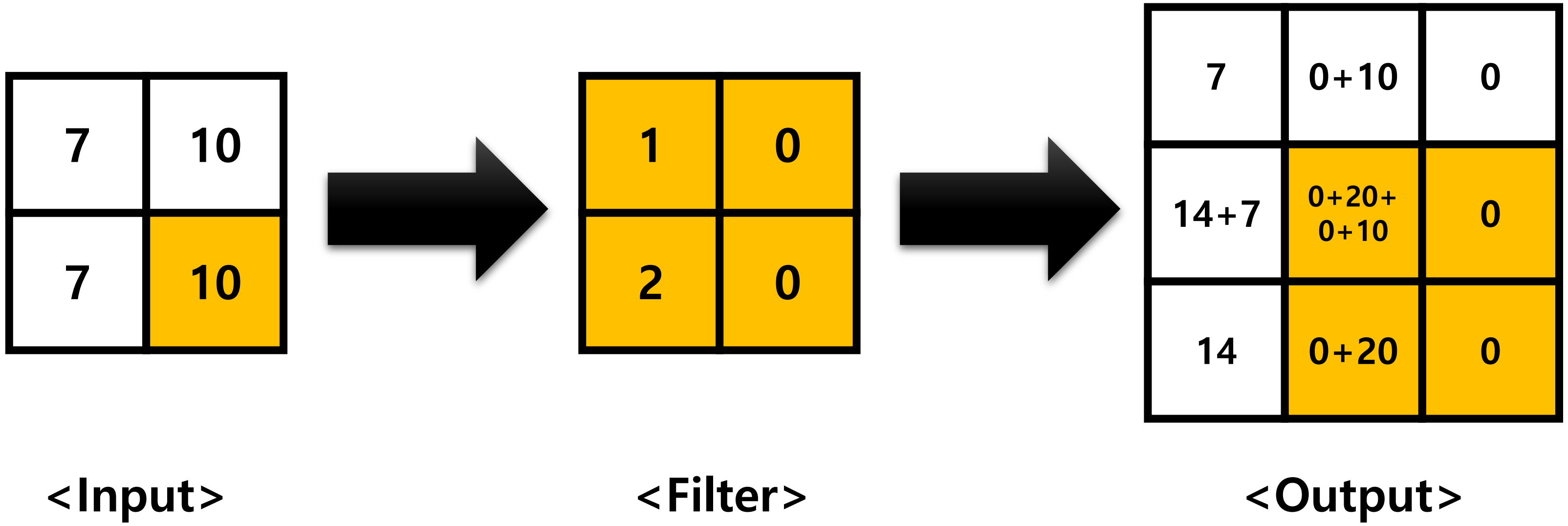
<Filter>

7	0+10	0
14+7	0+20 +0	0
14	0	

<Output>

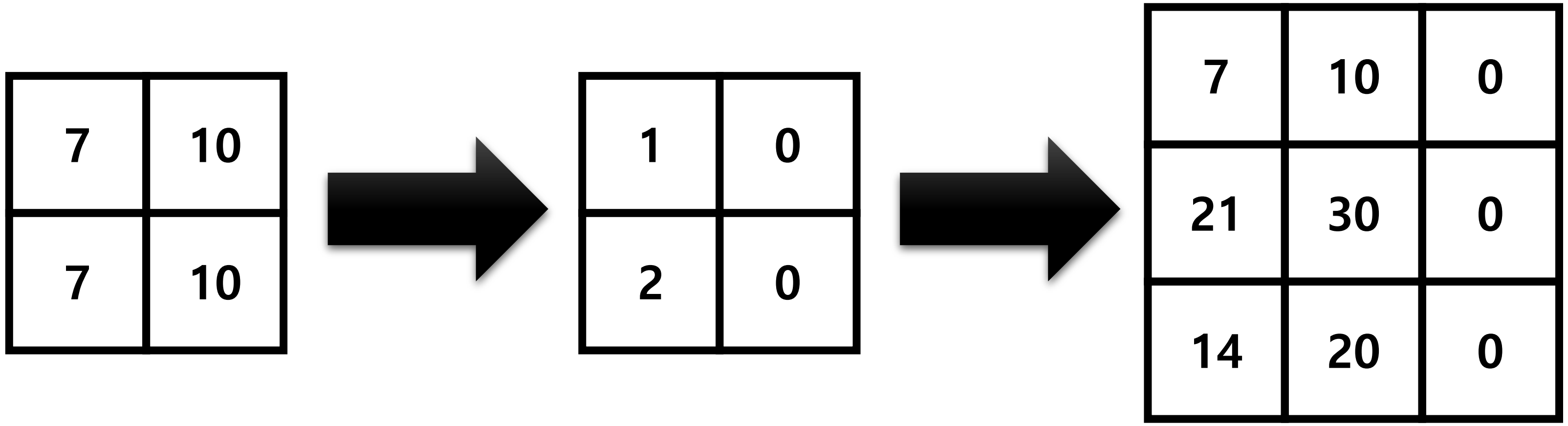
Convolutional Auto-Encoder

▪ Transposed Convolution



Convolutional Auto-Encoder

- **Transposed Convolution**



<Input> <Filter> <Output>

Fast campus

Convolutional Auto-Encoder

- **Transposed Convolution**
  - Input : $N \times N$
  - Padding : P
  - Stride : S
  - Filter : $F \times F$
  - Output : $\left(\frac{N+2P-F}{S} + 1\right) \times \left(\frac{N+2P-F}{S} + 1\right)$

Fast campus

Convolutional Auto-Encoder

- **Transposed Convolution**
  - $\mathrm{o} = (\frac{I+2P-F}{S} + 1)$
  - **Transposed**
  - $\mathrm{I} = \left(\frac{O+2P-F}{S} + 1\right)$
  - $\mathrm{I} - 1 = \frac{O+2P-F}{S}$
  - $\mathrm{S} * (\mathrm{I} - 1) = O + 2P - F$
  - $\therefore\ \mathbf{O} = \mathbf{S} * (\mathbf{I} - \mathbf{1}) + \mathbf{F} - \mathbf{2P}$

Convolutional Auto-Encoder

- **Transposed Convolution**
  - $\mathrm{O} = S * (I - 1) + \mathrm{F} - 2\mathrm{P}$
  - $\mathrm{O} = 1 * (2 - 1) + 2 - 2 * 0 = 1 + 2 = 3$

7	10
7	10

<Input>

1	0
2	0

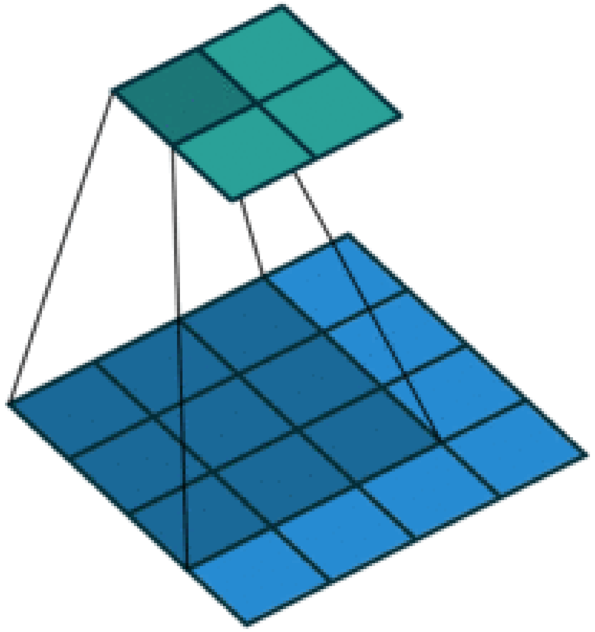
<Filter>

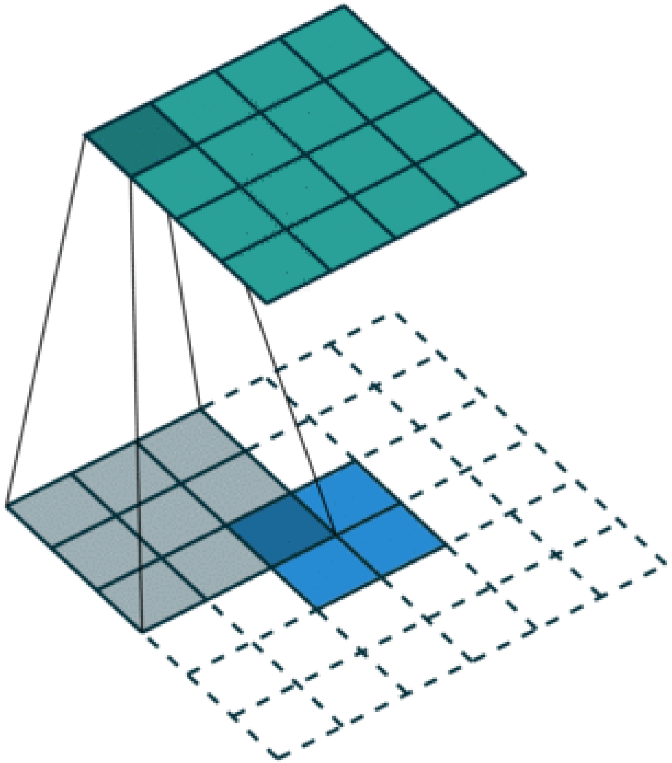
7	10	0
21	30	0
14	20	0

<Output>

Convolutional Auto-Encoder

- **Transposed Convolution**
  - $O = S * (I - 1) + F - 2P$

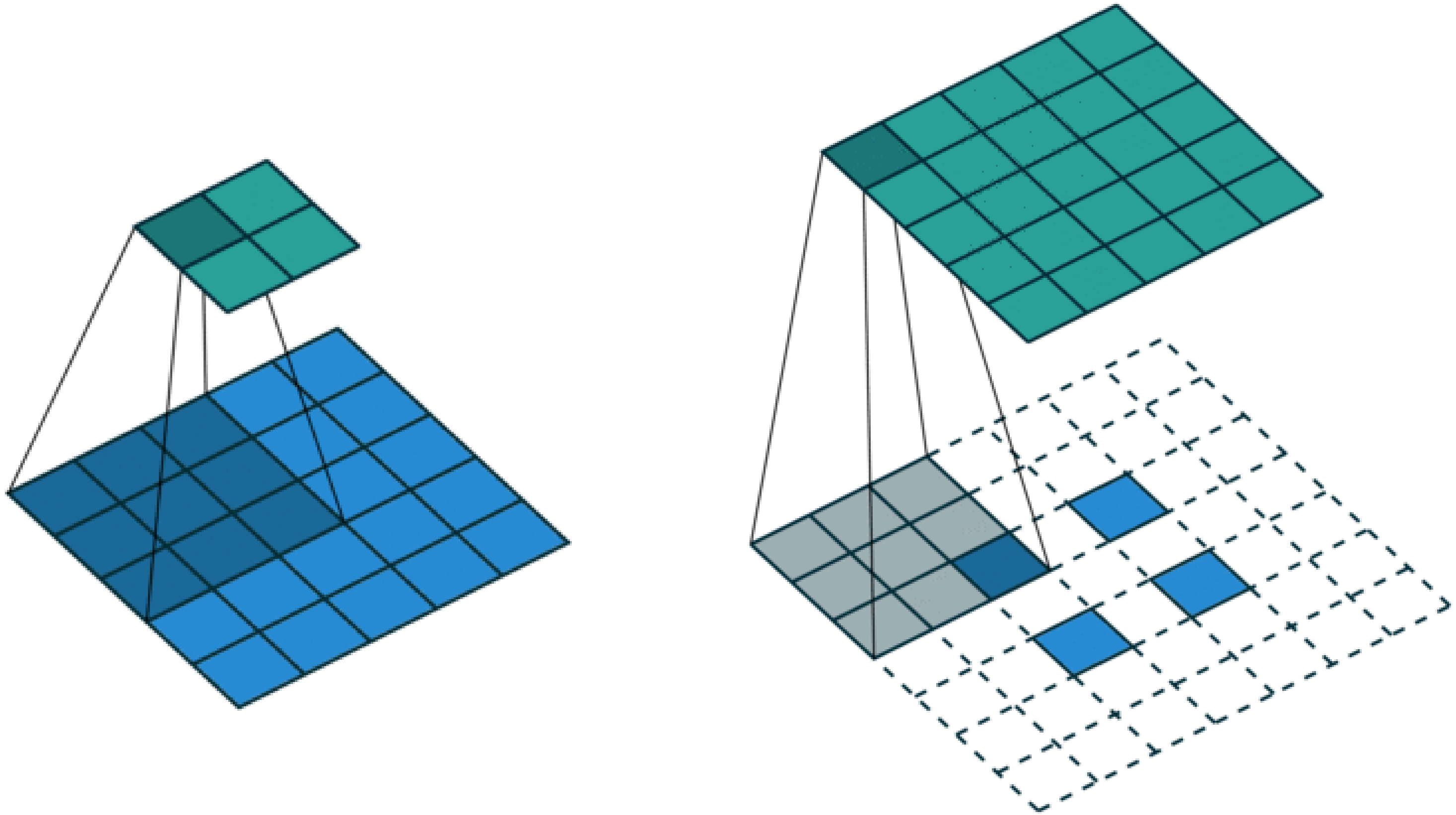




http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Convolutional Auto-Encoder

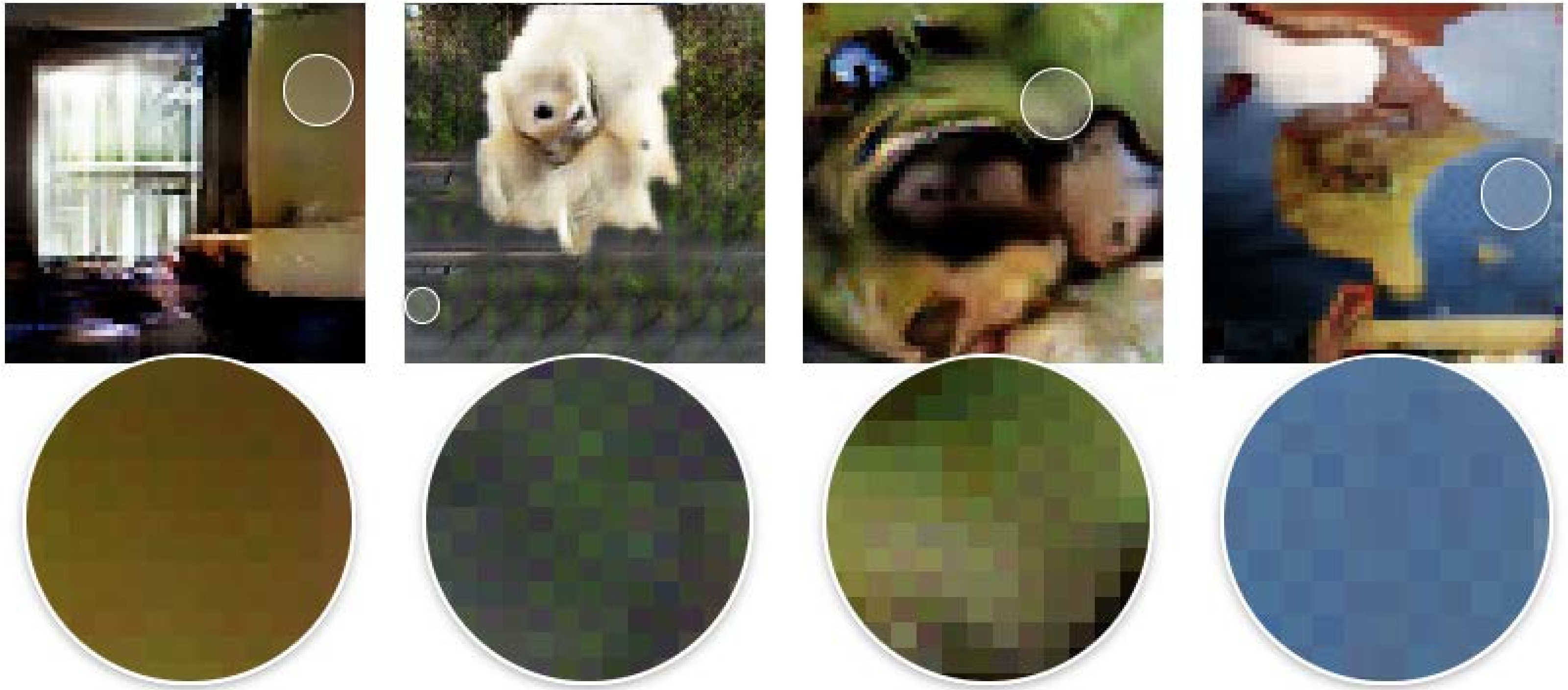
- **Transposed Convolution**
  - $O = S * (I - 1) + F - 2P$



http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html

Convolutional Auto-Encoder

- **Transposed Convolution의 한계**
  - Checker Board 현상

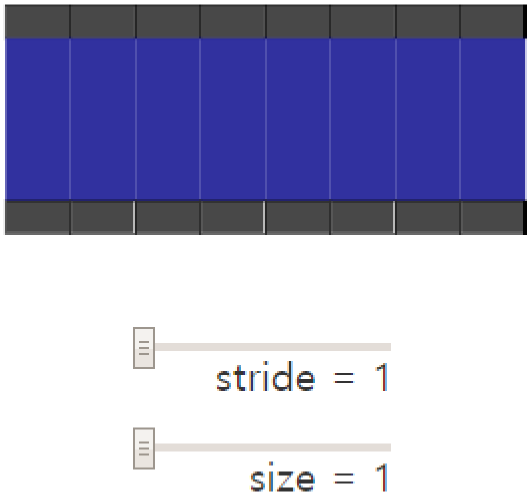


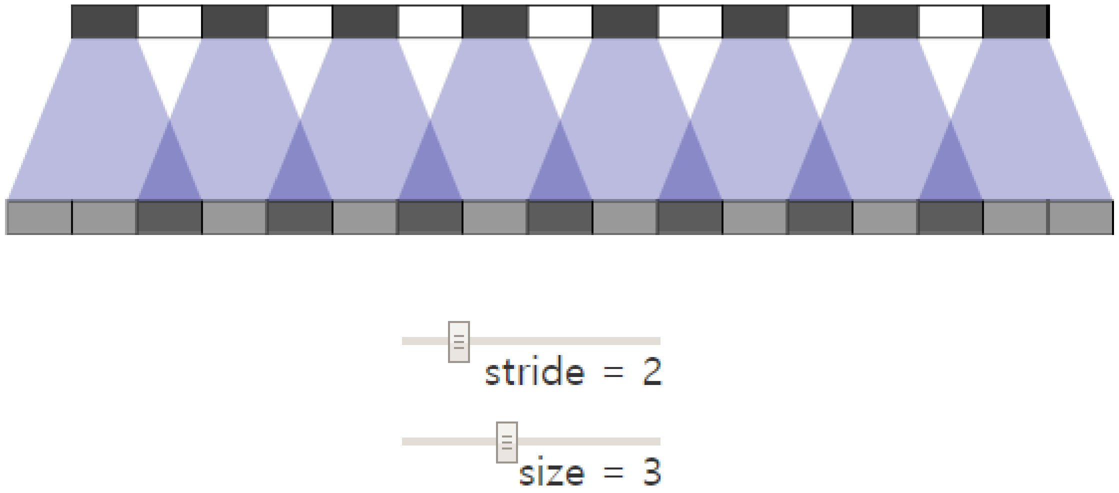
Radford, et al., 2015 [1]   Salimans et al., 2016 [2]   Donahue, et al., 2016 [3]   Dumoulin, et al., 2016 [4]

https://distill.pub/2016/deconv-checkerboard/

Convolutional Auto-Encoder

- Transposed Convolution의 한계

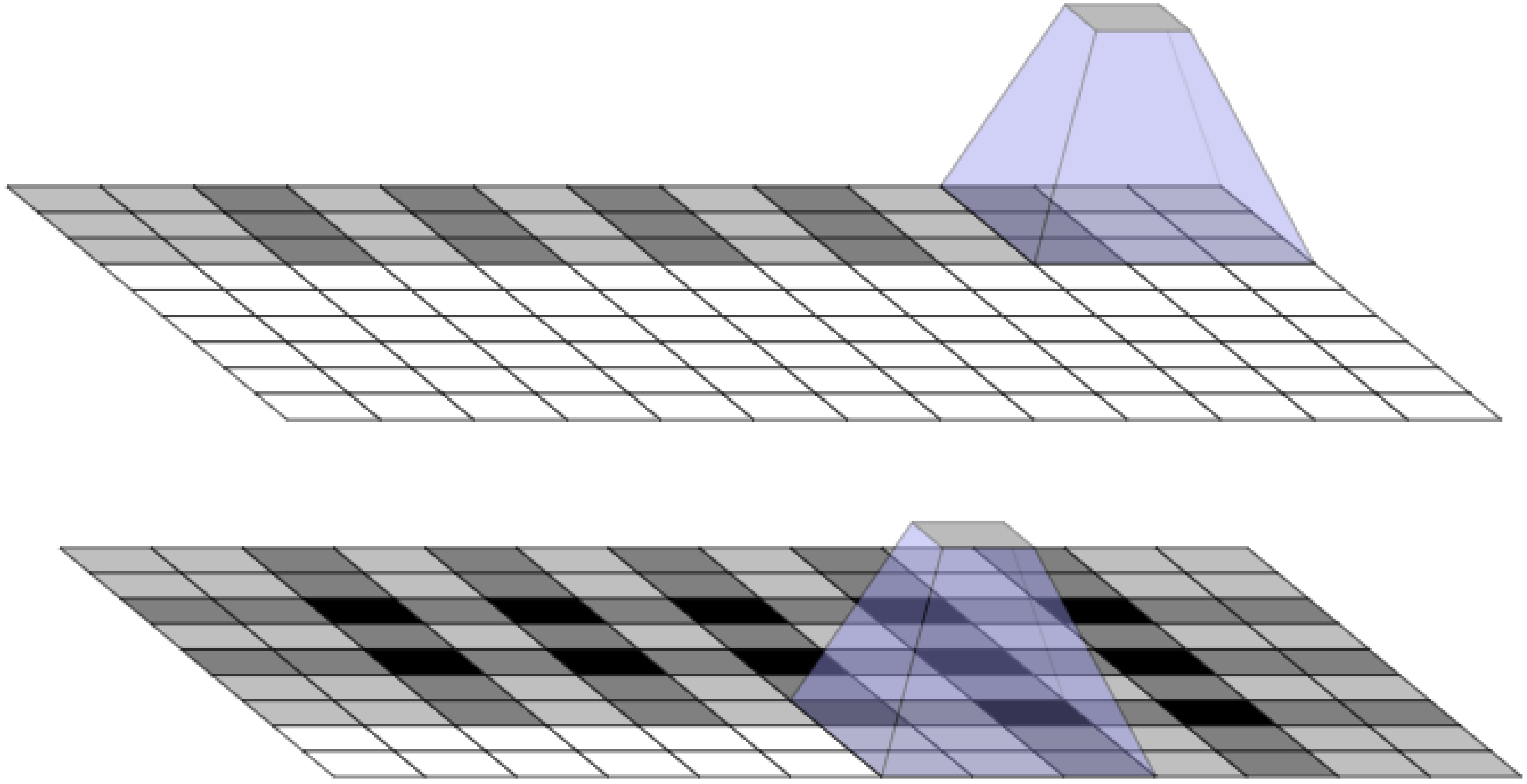




https://distill.pub/2016/deconv-checkerboard/

Convolutional Auto-Encoder

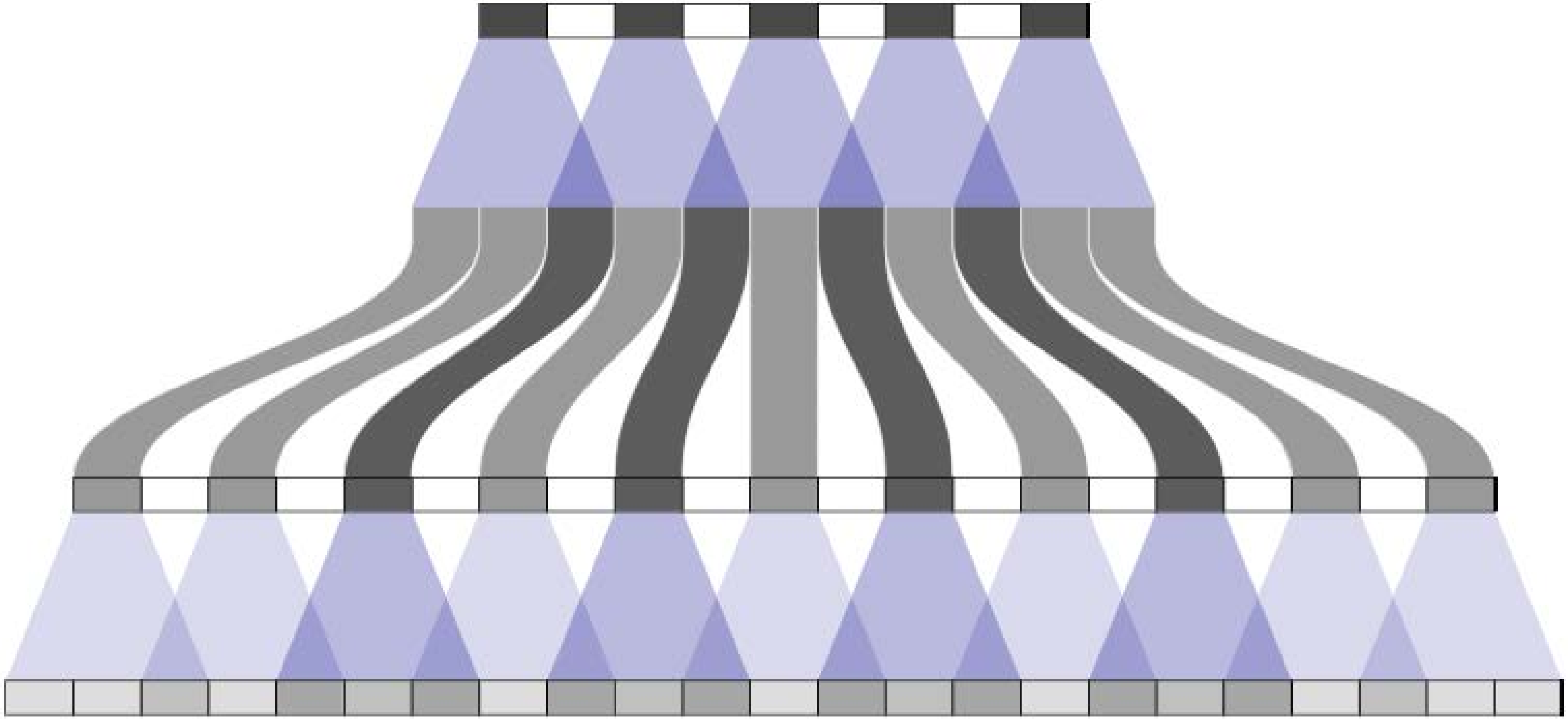
- Transpose Convolution의 한계



https://distill.pub/2016/deconv-checkerboard/

Convolutional Auto-Encoder

- **Transposed Convolution의 한계**



https://distill.pub/2016/deconv-checkerboard/

Convolutional Auto-Encoder

- **Transposed Convolution의 한계**
  - Checker Board 현상을 극복하기 위해, Transposed Convolution 후 보간법을 사용
  - 자세한 내용 : https://distill.pub/2016/deconv-checkerboard/

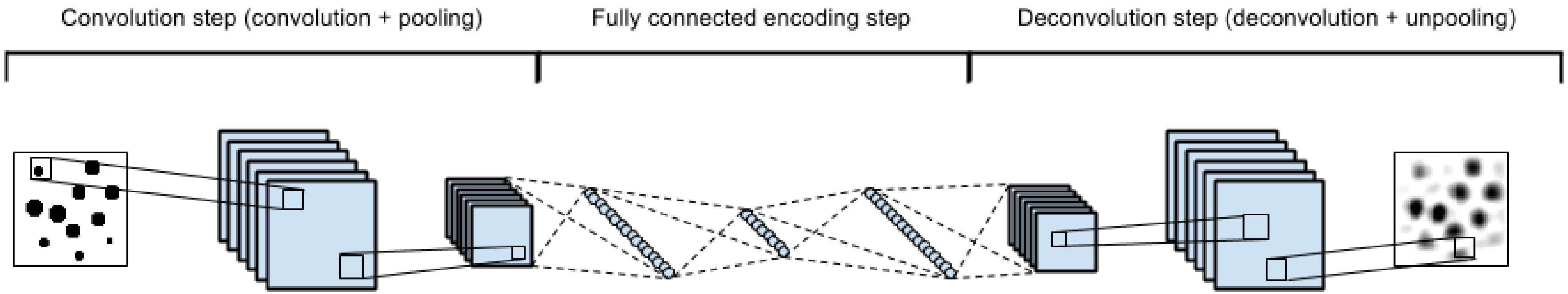
Deconv in last two layers.
Other layers use resize-convolution.
Artifacts of frequency 2 and 4.

Deconv only in last layer.
Other layers use resize-convolution.
Artifacts of frequency 2.

All layers use resize-convolution.
No artifacts.

Convolutional Auto-Encoder

- Convolutional Auto-Encoder



https://swarbrickjones.wordpress.com/2015/04/29/convolutional-autoencoders-in-pythontheanolasagne/

Fast campus

Generative Model

Auto-Encoder

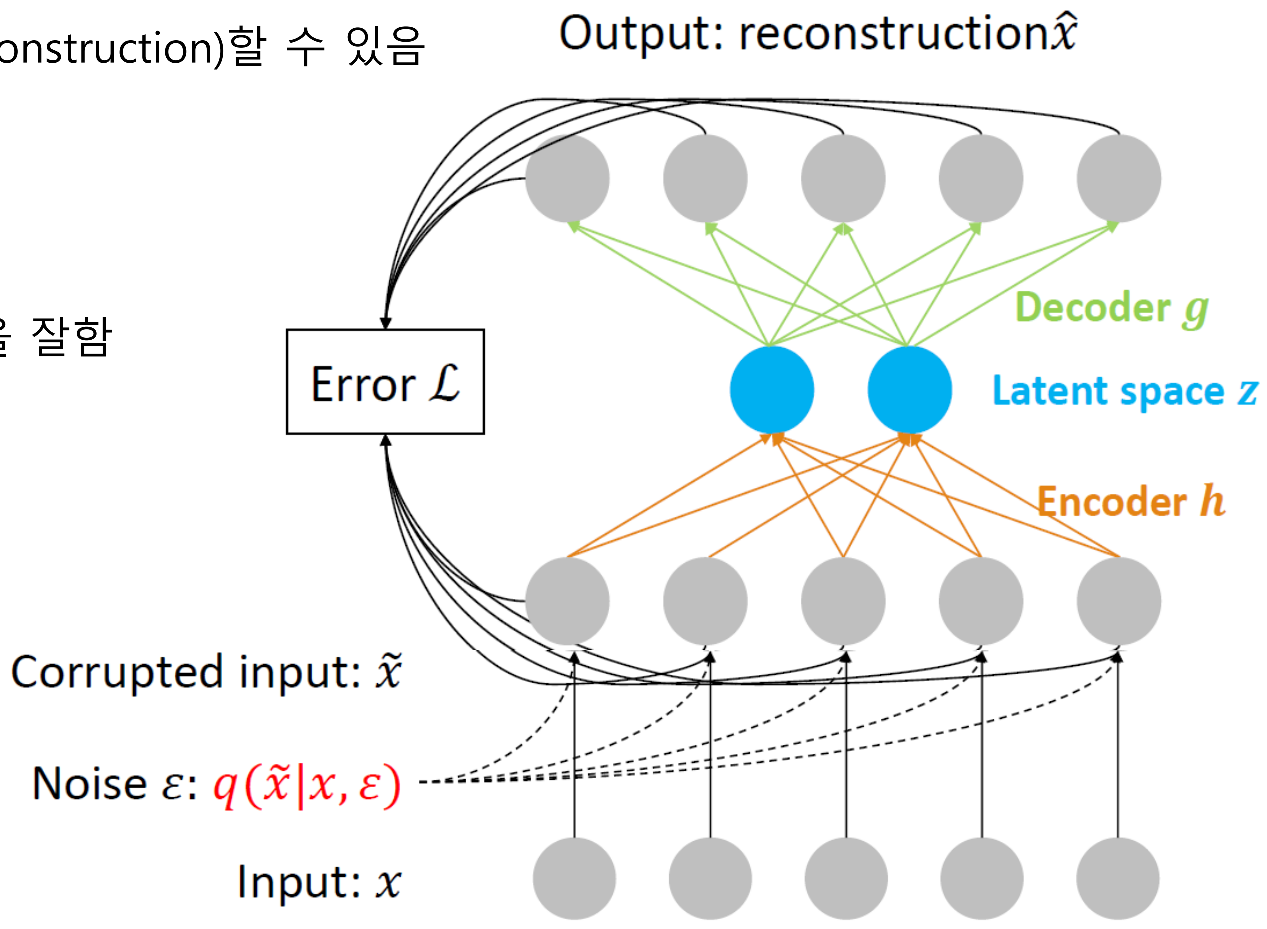
Convolutional Auto-Encoder

Variational Auto-Encoder

SVHN

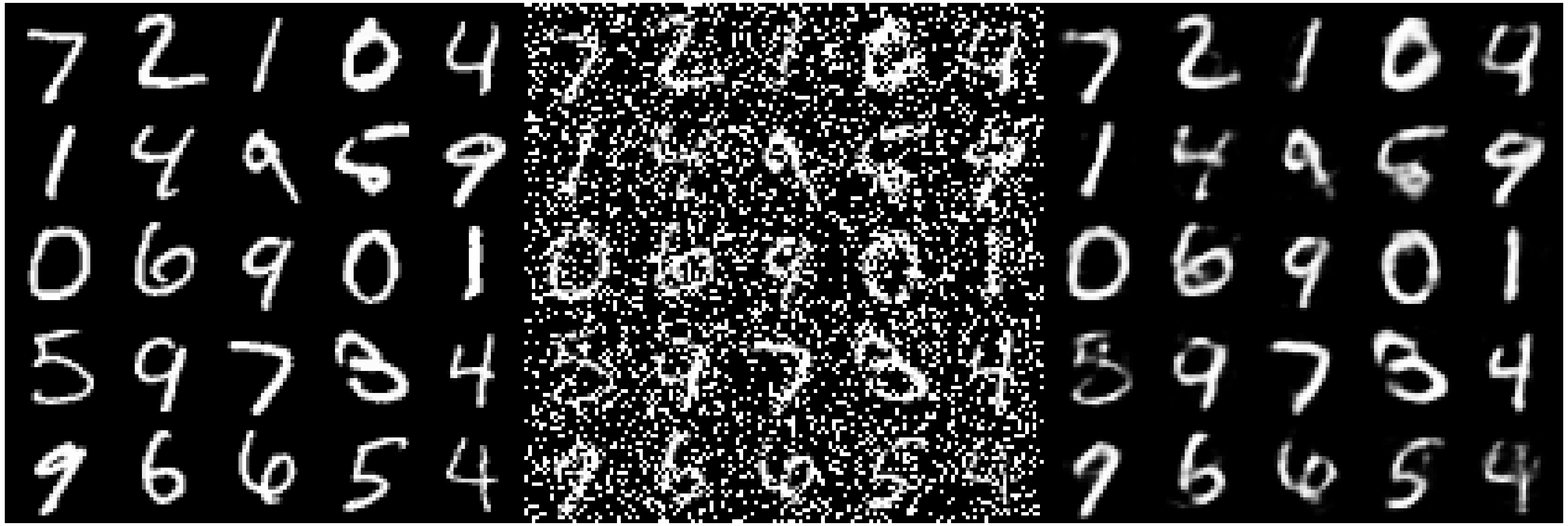
Convolutional Auto-Encoder

- **Denoising Auto-Encoder(DAE)**
  - 데이터의 특징을 잘 추출할 수 있다면,
  - 노이즈가 낀 데이터를 복구(Reconstruction)할 수 있음
  - $\mathcal{L} = \sum_d E_{q(\hat{x}|x)} \ell(x, g(h(\hat{x})))$
  - 대부분 노이즈를 주면 더 생성을 잘함



Convolutional Auto-Encoder

- **Denoising Auto-Encoder(DAE)**

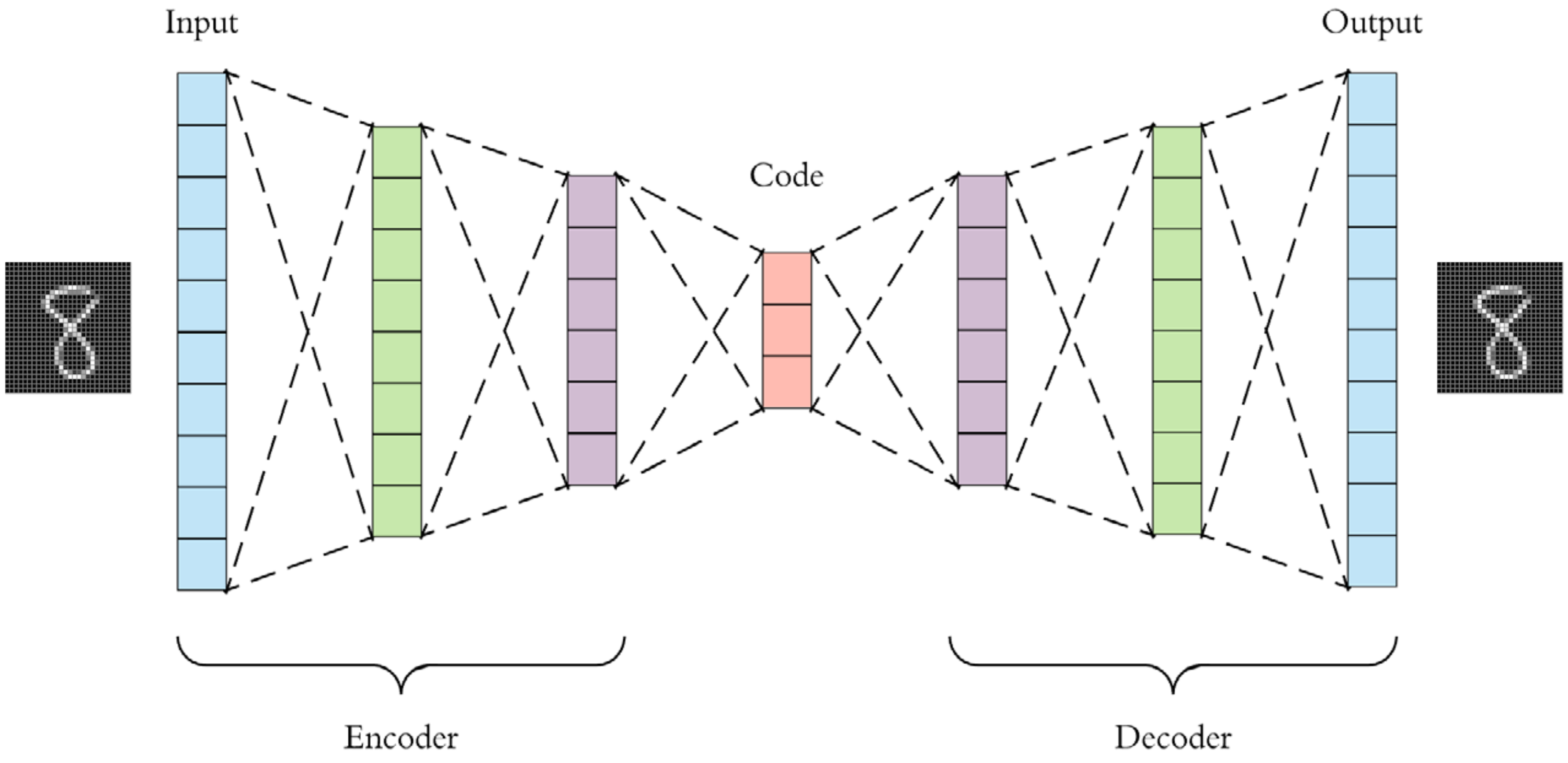


https://www.doc.ic.ac.uk/~js4416/163/website/autoencoders/denoising.html

4. Variational Auto-Encoder

Variational Auto-Encoder

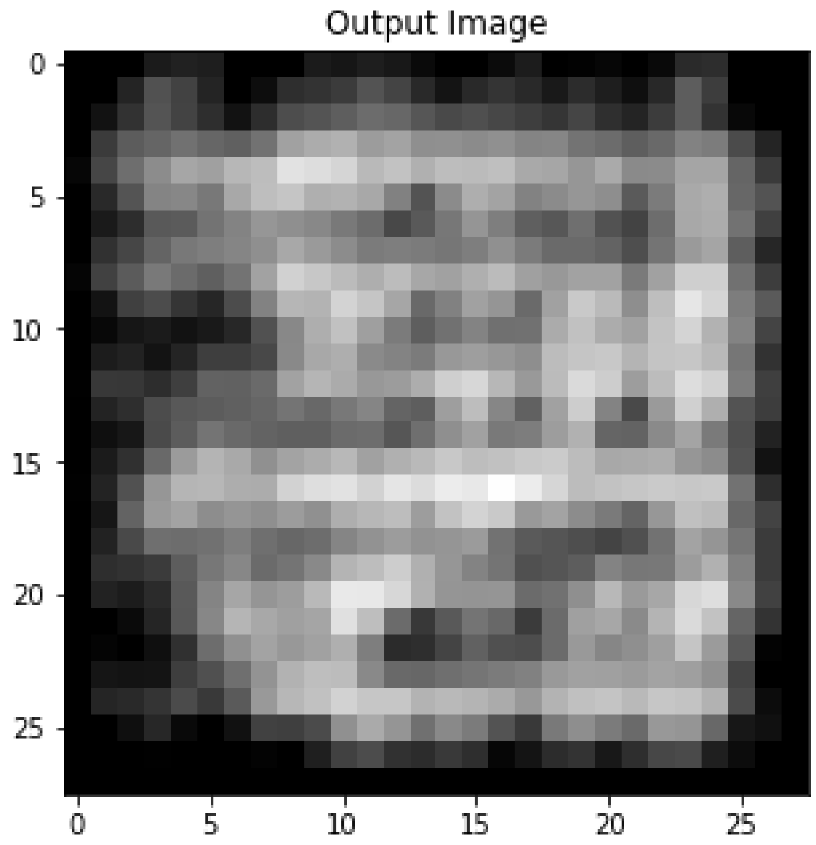
- **Latent Vector의 분석**
  - 얻어진 Latent Vector는 알지만..
  - 어떤 분포나 의미를 갖는지는 알기 쉽지 않음



https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

Variational Auto-Encoder

- **Latent Vector의 분석**
  - 얻어진 Latent Vector는 알지만..
  - 어떤 분포나 의미를 갖는지는 알기 쉽지 않음

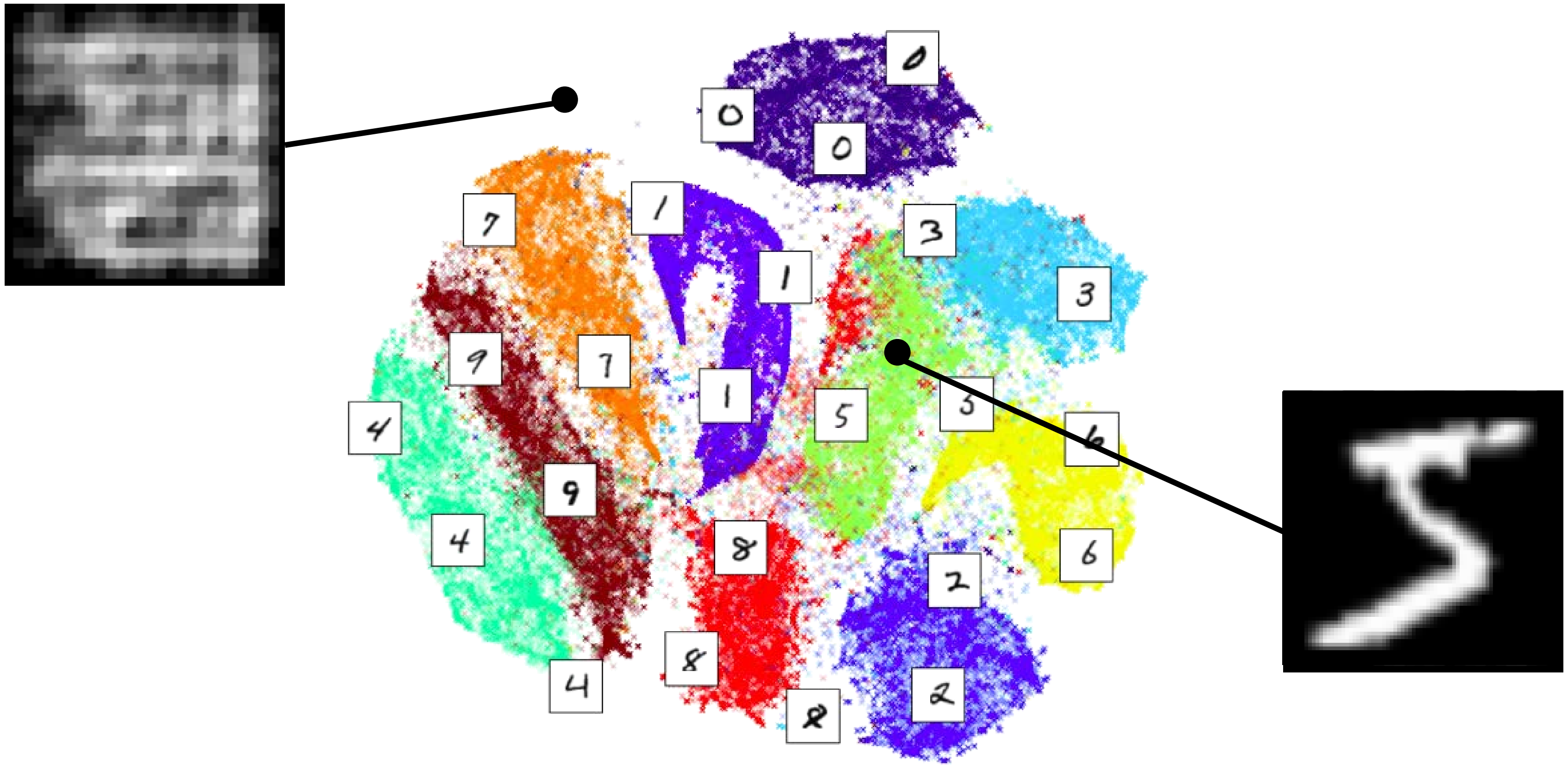


$$\begin{bmatrix} 0.56 \\ -13.2 \\ 383.2 \\ -30.1 \\ \vdots \\ 1.1 \\ 8.45 \end{bmatrix} \textbf{?}$$

Variational Auto-Encoder

▪ Latent Vector의 분석

- Latent Vector가 모든 공간을 사용하지는 않을 것
- 또한 완벽한 모델일 경우, 비슷한 이미지라면 비슷한 Latent Vector를 가지고 있을 것임



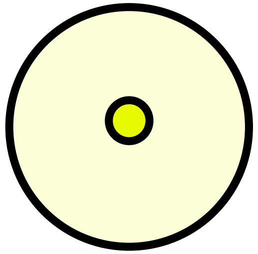
https://nlml.github.io/in-raw-numpy/in-raw-numpy-t-sne/

Variational Auto-Encoder

- **Variational Auto-Encoder**
  - 하지만, Latent Vector의 분포가 3차원 이상일 때는 파악하기 힘듦
  - 그렇다면, 우리가 아는 분포로 Latent Vector를 만들도록 모델을 학습하면 어떨까?
    - 정규분포 $x \sim N(\mu, \sigma)$: 평균, 표준편차
    - Latent Vector가 정규분포를 따르면서 분포되어 있을 때,
    - Latent Vector의 평균과 표준편차를 알면 Latent Vector의 분포 영역 안에서 잘 생성할 수 있을 것

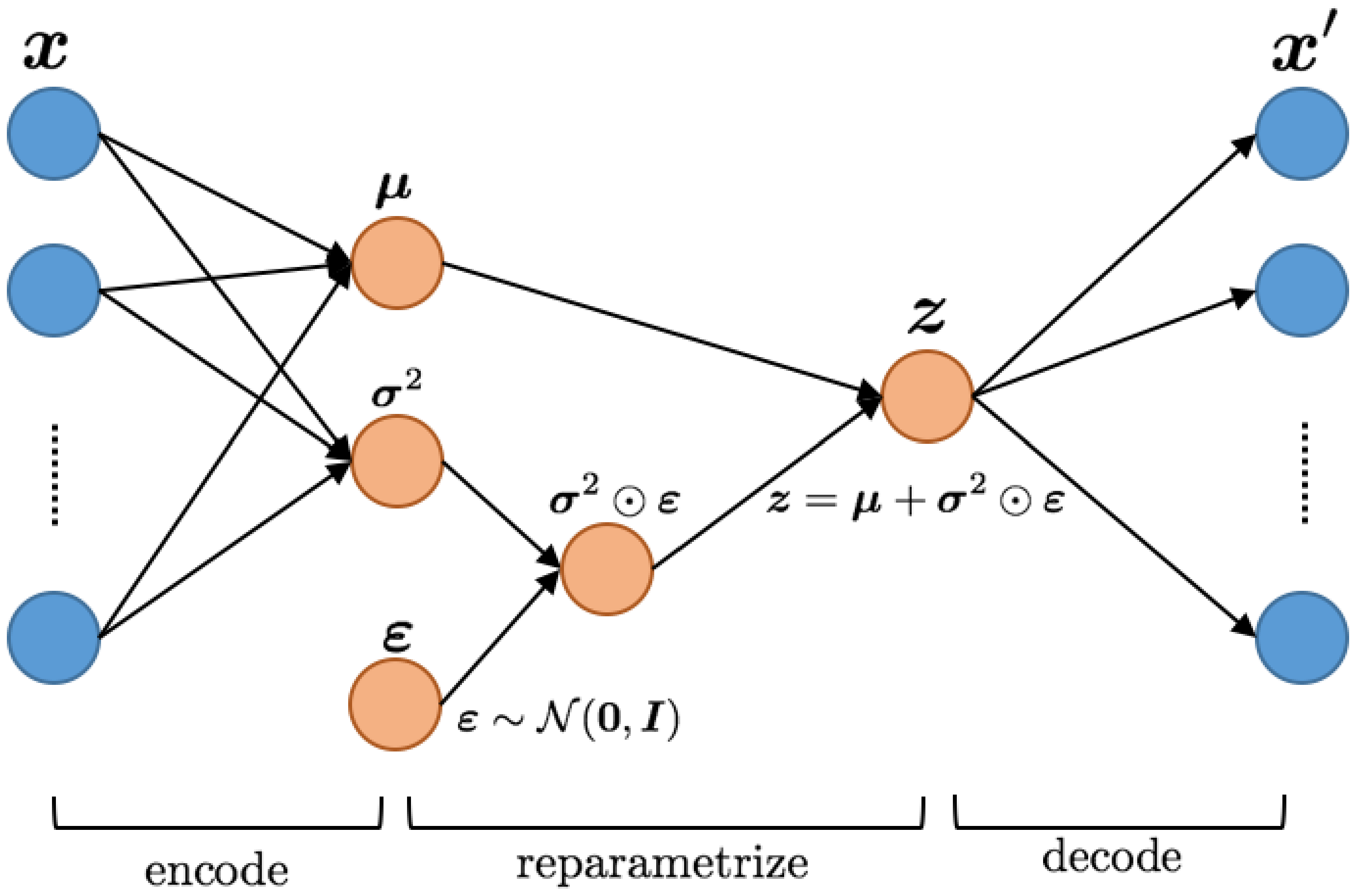
분포되어 있을 때,

면 Latent Vector의 분포 영역 안에서 잘 생성할 수 있을 것



Variational Auto-Encoder

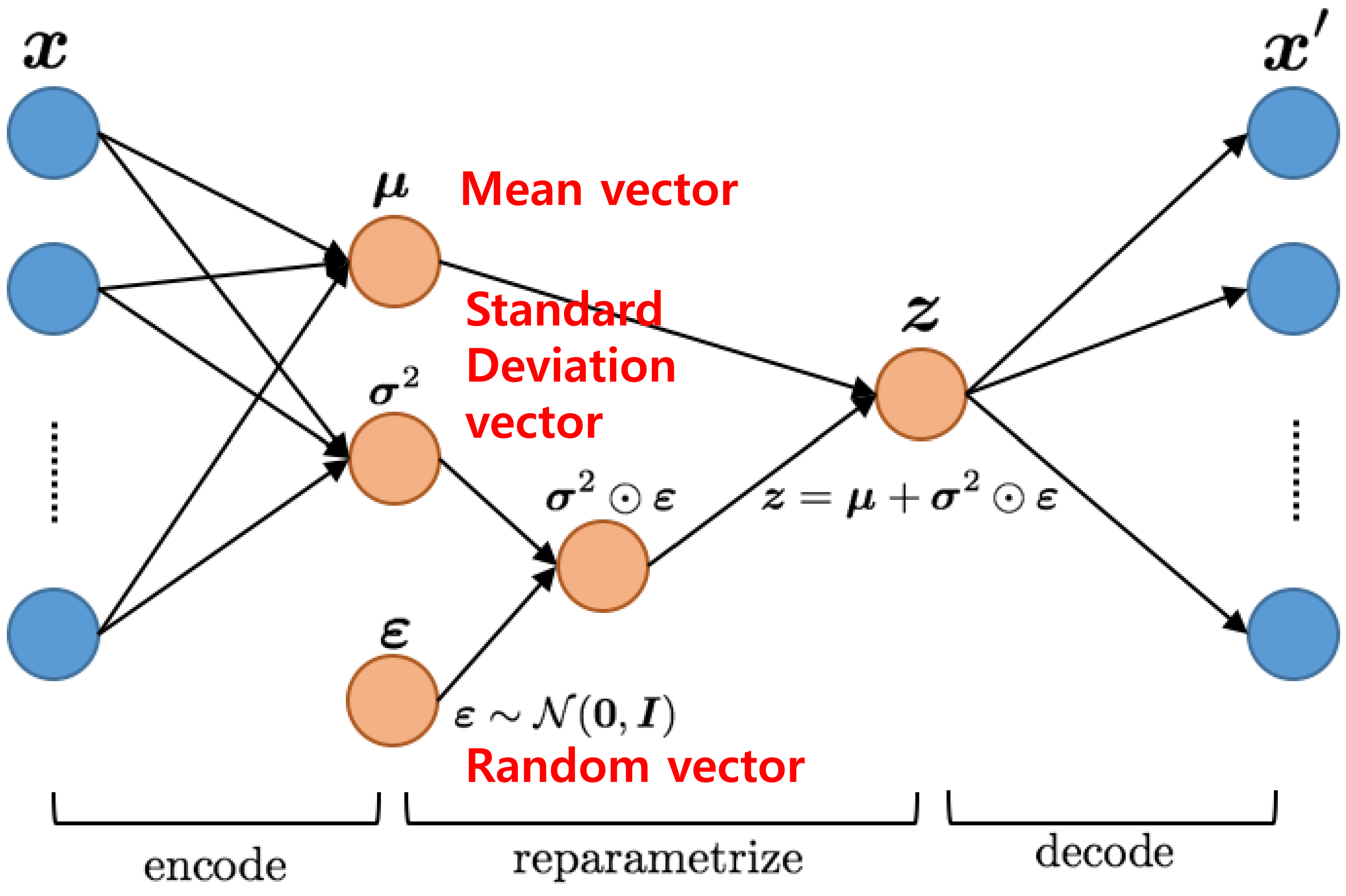
- Variational Auto-Encoder



http://www.renom.jp/notebooks/tutorial/generative-model/VAE/notebook.html

Variational Auto-Encoder

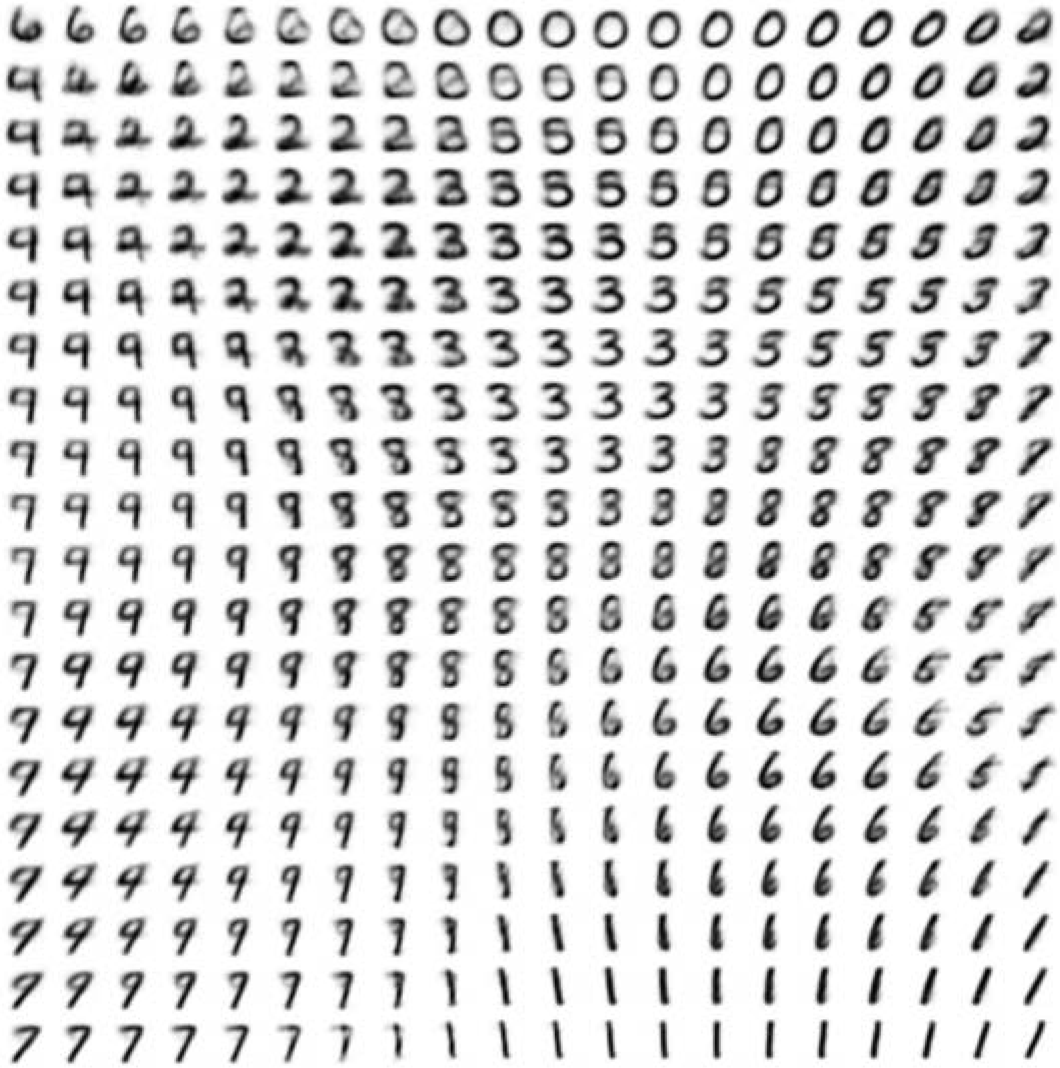
- Variational Auto-Encoder



http://www.renom.jp/notebooks/tutorial/generative-model/VAE/notebook.html

Variational Auto-Encoder

(a) Learned Frey Face manifold



(b) Learned MNIST manifold

Variational Auto-Encoder

(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

5. SVHN

SVHN

- **The Street View House Numbers (SVHN) Dataset**
  - http://ufldl.stanford.edu/housenumbers/
  - SVHN is **a real-world image dataset** for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting. It can be seen as similar in flavor to MNIST (e.g., the images are of small cropped digits), but incorporates an order of magnitude more labeled data (over 600,000 digit images) and comes from a significantly **harder, unsolved, real world problem** (recognizing digits and numbers in natural scene images). SVHN is **obtained from house numbers in Google Street View images**.

SVHN

- **The Street View House Numbers (SVHN) Dataset**
  - 10 classes, 1 for each digit. Digit '1' has label 1, '9' has label 9 and '0' has label 10.
  - 73257 digits for training, 26032 digits for testing, and 531131 additional, somewhat less difficult samples, to use as extra training data
  - Comes in two formats:
    - 1. Original images with character level bounding boxes.
    - 2. MNIST-like 32-by-32 images centered around a single character (many of the images do contain some distractors at the sides).

SVHN

- The Street View House Numbers (SVHN) Dataset

Fast campus

Generative Model

Auto-Encoder

Convolutional Auto-Encoder

Variational Auto-Encoder

SVHN

SVHN

- The Street View House Numbers (SVHN) Dataset

<부록>
Evaluation on Generative Model

Evaluation on Generative Model

- **생성형 모델의 가정과 문제**
  - **가정 :** 데이터(Data) x는 어떠한 특성 변수(Latent Variable) z로부터 생성된다.
  - **목적 :** 궁극적으로 data의 distribution $p_\theta(x)$를 알고자 함.
  - **과정 :**
    - z가 prior distribution $p_\theta(z)$로부터 생성된다.
    - x는 likelihood distribution $p_\theta(x|z)$으로부터 생성된다.
  - **문제 :**
    - 아무것도 알지 못한다.

$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$

Evaluation on Generative Model

Generative Model

Auto-Encoder

Convolutional Auto-Encoder

Variational Auto-Encoder

SVHN

- **생성형 모델의 문제 해결법**
  - $p_\theta(z)$는 우리가 알고 있는 쉬운 분포로 가정하자

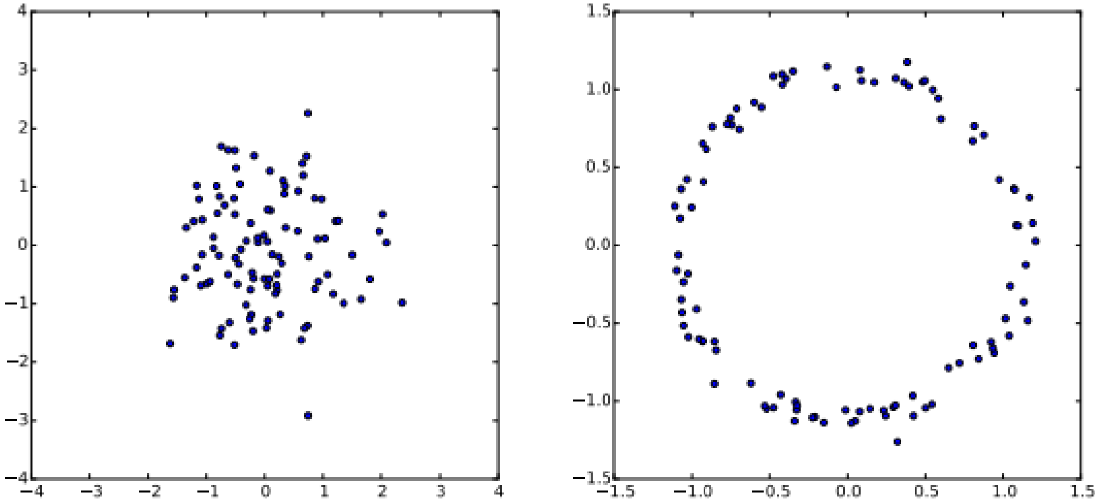
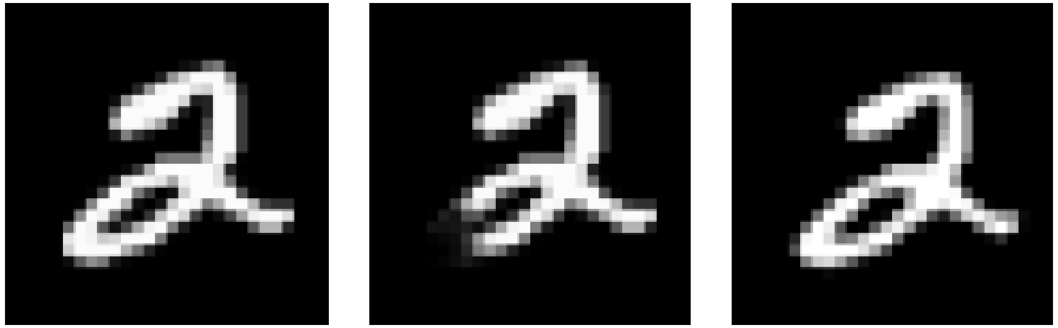


Figure 2: Given a random variable z with one distribution, we can create another random variable $X = g(z)$ with a completely different distribution. Left: samples from a gaussian distribution. Right: those same samples mapped through the function $g(z) = z/10 + z/||z||$ to form a ring. This is the strategy that VAEs use to create arbitrary distributions: the deterministic function g is learned from data.

Evaluation on Generative Model

- **생성형 모델의 문제 해결법**
  - $p_\theta(x|z)$도 우리가 알고 있는 쉬운 분포로 가정할 수 있는가? **NO**

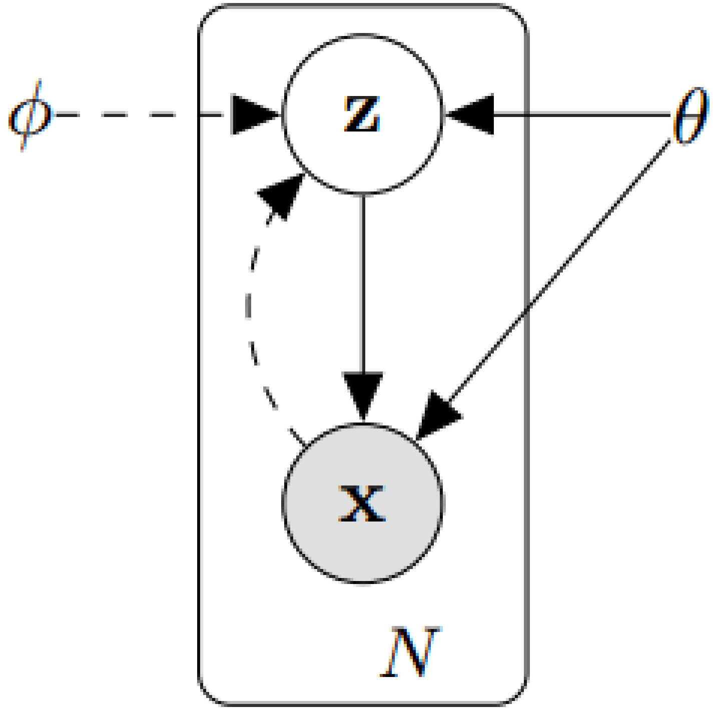


(a) (b) (c)

Figure 3: It's hard to measure the likelihood of images under a model using only sampling. Given an image X (a), the middle sample (b) is much closer in Euclidean distance than the one on the right (c). Because pixel distance is so different from perceptual distance, a sample needs to be extremely close in pixel distance to a datapoint X before it can be considered evidence that X is likely under the model.

Evaluation on Generative Model

- **생성형 모델의 문제 해결법**
  - 그렇다면 $p_\theta(z)$가 아닌 $p_\theta(z|x)$를 활용하여 더 좋은 이미지를 생성하자.
  - 하지만 $p_\theta(z|x)$ 역시 모르므로, 이를 추측할 수 있는 모델 $q_\phi(z|x)$을 만들자.



Evaluation on Generative Model

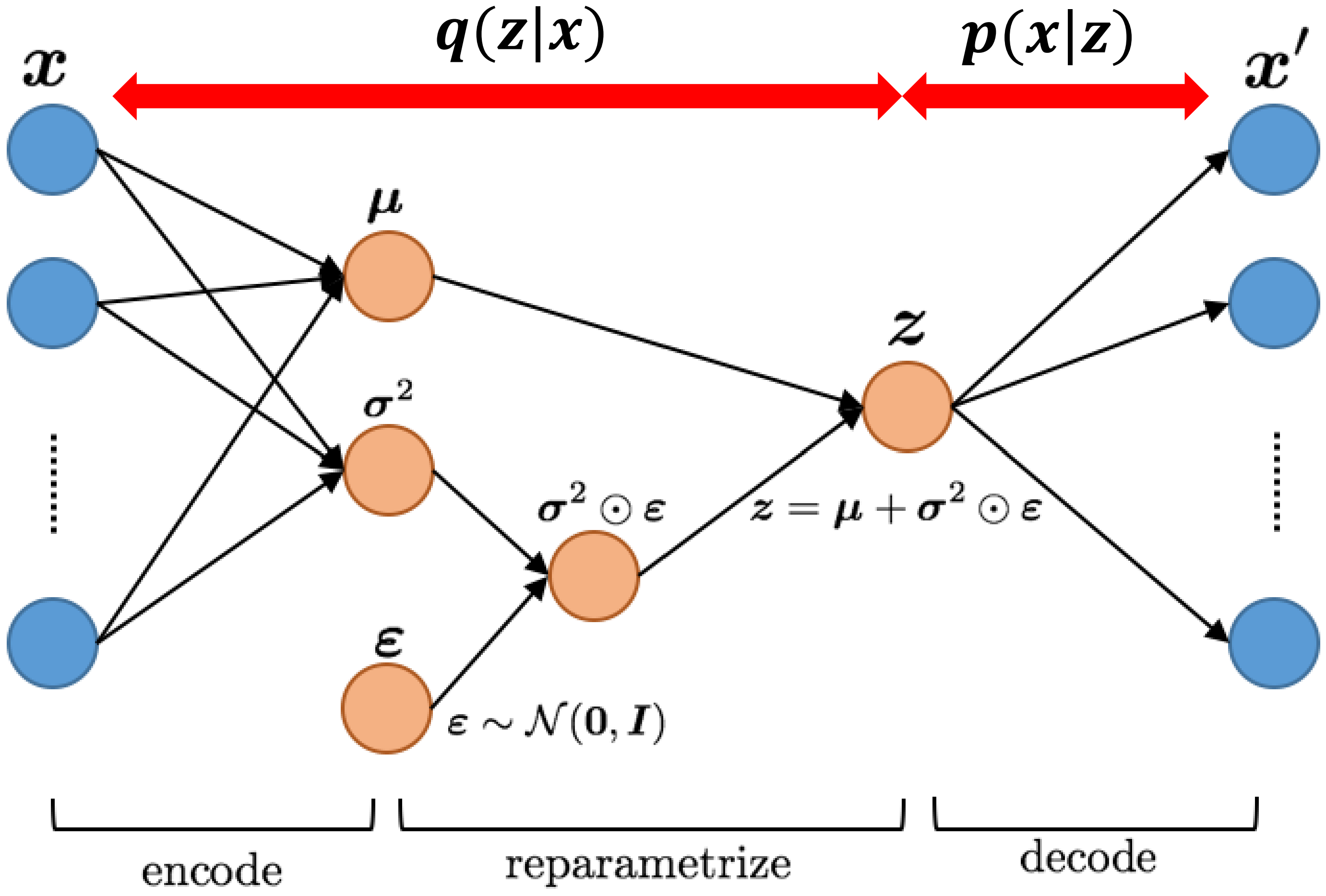
- **생성형 모델의 학습**
  - 그렇다면 $p_\theta(z)$가 아닌 $p_\theta(z|x)$를 활용하여 더 좋은 이미지를 생성하자.
  - 하지만 $p_\theta(z|x)$ 역시 모르므로, 이를 추측할 수 있는 모델 $q_\phi(z|x)$을 만들자.

$$dist(q_\phi(z|x)||p_\theta(z|x))$$

Evaluation on Generative Model

Generative Model
Auto-Encoder
Convolutional Auto-Encoder
Variational Auto-Encoder
SVHN

- Variational Auto-Encoder



http://www.renom.jp/notebooks/tutorial/generative-model/VAE/notebook.html

Fast campus

Evaluation on Generative Model

- Variational Auto-Encoder
  - $\boldsymbol{D_{KL}(q_\phi(z)||p_\theta(z|x))}$

    $= D_{KL}(q_\phi(z)||p_\theta(z)) + \log(p_\theta(z)) - E_{z\sim q_{\phi(Z|X)}}[logp_\theta(x|z)]$
  - $\log(p_\theta(z)) = \boldsymbol{ELBO} + D_{KL}(q_\phi(z)||p_\theta(z|x)) \geq \boldsymbol{ELBO}$
  - $\boldsymbol{ELBO = E_{z\sim q_{\phi(Z|X)}}[logp_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p_\theta(z))}$
  - $Max(\log(p_\theta(z))) = Max(ELBO) = MIN(-ELBO)$

Evaluation on Generative Model

- **Variational Auto-Encoder**

  - $\mathcal{L}(\theta, \phi, x) = \underbrace{-E_{z \sim q_{\phi(Z|X)}}[\log p_\theta(x|z)]}_{\text{Reconstruction Loss}} + \underbrace{D_{KL}(q_\phi(z|x) || p_\theta(z))}_{\text{KL Divergence Loss}}$

Evaluation on Generative Model

- Variational Auto-Encoder

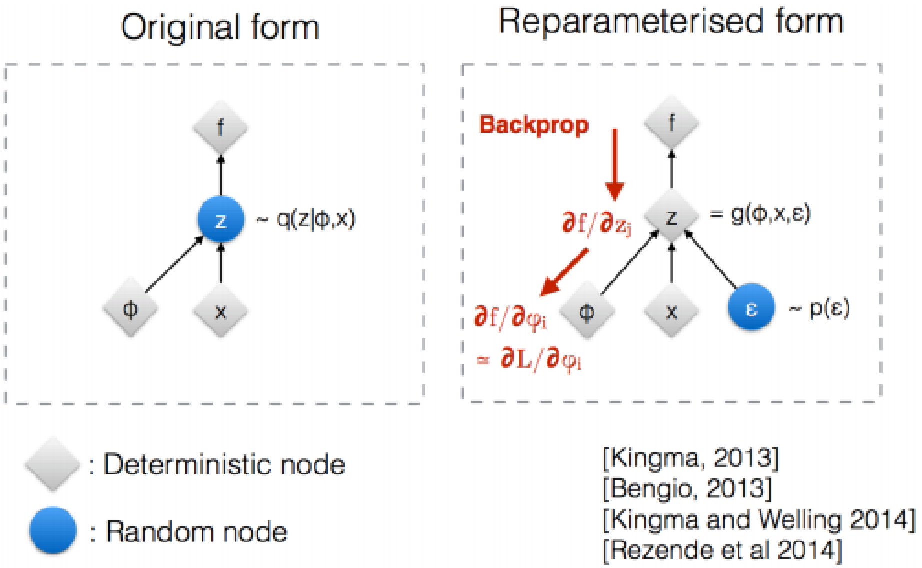
  - $\mathcal{L}(\theta,\phi,x) = \underbrace{-E_{z\sim q_{\phi}(Z|X)}[logp_{\theta}(x|z)]}_{\text{Reconstruction Loss}} + D_{KL}(q_{\phi}(z|x)||p_{\theta}(z))$

  - Variational Inference

  - $q_{\phi}(z|x) \sim N(\mu_q(x), \Sigma_q(x))$

  - Reparameterization trick

  - $z = \mu(x) + \sigma(x) * \varepsilon, \quad \varepsilon \sim N(0,1)$



[Kingma, 2013]
[Bengio, 2013]
[Kingma and Welling 2014]
[Rezende et al 2014]

Evaluation on Generative Model

- **Variational Auto-Encoder**
  - $\mathcal{L}(\theta,\phi,x) = -E_{z\sim q_{\phi}(Z|X)}[logp_{\theta}(x|z)] + D_{KL}(q_{\phi}(z|x)||p_{\theta}(z))$

    KL Divergence Loss
  - **KL Divergence(KLD)**
  - $D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x)log\frac{p(x)}{q(x)}dx$
  - **Z가 zero-mean Gaussian일 때,**
  - $D_{KL}(q_{\phi}(z|x)\big|\big|p_{\theta}(z)\big) = D_{KL}[N(\mu_q(x),\Sigma_q(x))||N(0,1)]$

    $$= \frac{1}{2}\sum_{k}\{\exp(\Sigma_q(x)) + \mu_q(x)^2 - 1 - \Sigma_q(x)\}$$

Fast campus

Evaluation on Generative Model

Generative Model

Auto-Encoder

Convolutional Auto-Encoder

Variational Auto-Encoder

SVHN

- 생성형 모델의 학습 방법
  - 두 분포의 거리를 측정할 수 있는 방법이면 가능
  - Kullback-Leibler Divergence(KLD)
  - Maximum Mean Discrepancy(MMD)
  - Jensen-Shannon Divergence(JSD)

MMD (Gretton et al., 2007)

$$MMD[p,q] = \left(E_{p,q}[k(x,x') - 2k(x,y) + k(y,y')]\right)^{1/2}$$

Where x, x' are indepent and distributed according to the data distribution p, and y, y' are independently distributed according to the model distribution q.

Optimizing an empirical estimate of MMD.

Using a mixture of Gaussian kernels with various bandwidths for k.

JSD (Goodfellow et al., 2014)

$$JSD[p,q] = \frac{1}{2}KLD[p||m] + \frac{1}{2}KLD[q||m]$$

Where $m = (p+q)/2$ is an equal mixture of distributions p and q. We optimized JSD directly using the data density, which is generally not possible in practice where we only have access to samples from the data distribution.

Parameters were initialized at the maximum likelihood solution in all cases, but the same optimum was consistently found using random initializations.

Fast campus

Evaluation on Generative Model

Generative Model

Auto-Encoder

Convolutional Auto-Encoder

Variational Auto-Encoder

SVHN

- **생성형 모델의 학습 방법**
  - 두 분포의 거리를 측정할 수 있는 방법이면 가능
  - Kullback-Leibler Divergence(KLD)
  - Maximum Mean Discrepancy(MMD)
  - Jensen-Shannon Divergence(JSD)

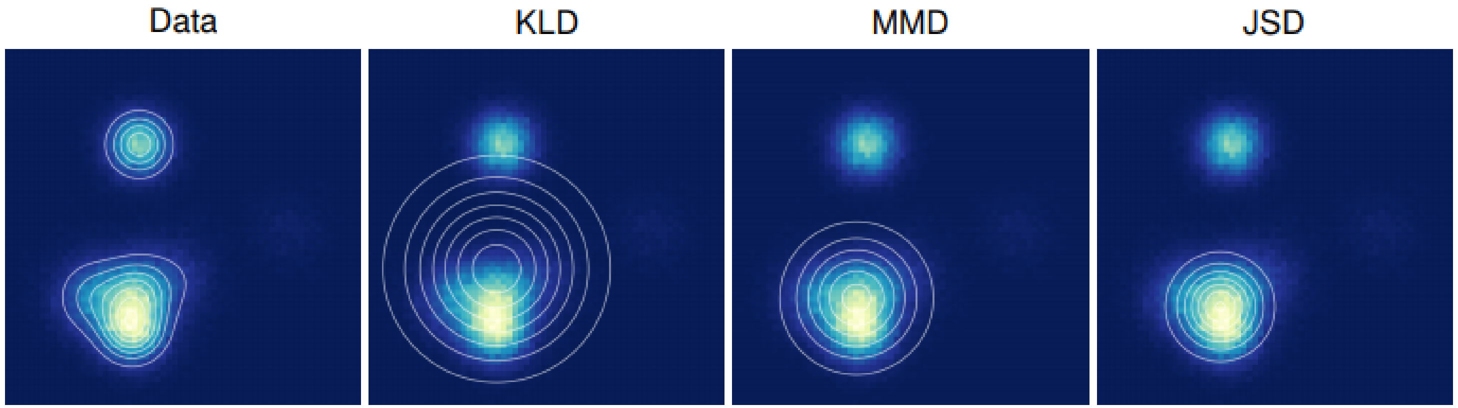


Figure 1: An isotropic Gaussian distribution was fit to data drawn from a mixture of Gaussians by either minimizing Kullback-Leibler divergence (KLD), maximum mean discrepancy (MMD), or Jensen-Shannon divergence (JSD). The different fits demonstrate different tradeoffs made by the three measures of distance between distributions.

Evaluation on Generative Model

- **생성형 모델의 평가 방법**
  - 원래 목적인 실제 $p_\theta(x)$를 잘 구했는지 확인해야함.
  - = $p(x|\theta)$를 최대화하는 θ를 잘 구했는가?
  - = 가지고 있는 샘플이 잘나오게 하는 모델을 구성했는가?

Fast campus

Evaluation on Generative Model

- **생성형 모델의 평가 방법**
  - 원래 목적인 실제 $p_\theta(x)$를 잘 구했는지 확인해야함.
  - = $p(x|\theta)$를 최대화하는 θ를 잘 구했는가?
  - = 가지고 있는 샘플이 잘나오게 하는 모델을 구성했는가?
  - 최대 가능도 방법($log - likelihood\ method$)를 통해 확인 가능
  - $log - likelihood = \log(\prod p(x|\theta))$

Fast campus

Evaluation on Generative Model

Generative Model

Auto-Encoder

Convolutional Auto-Encoder

Variational Auto-Encoder

SVHN

- **생성형 모델의 평가 방법**
  - 원래 목적인 실제 $p_\theta(x)$를 잘 구했는지 확인해야함.
  - = $p(x|\theta)$를 최대화하는 θ를 잘 구했는가?
  - = 가지고 있는 샘플이 잘나오게 하는 모델을 구성했는가?

  - 최대 가능도 방법$(log - likelihood\ method)$를 통해 확인 가능
  - $log - likelihood = \log(\prod p(x|\theta))$

  - **하지만 이미지 분야에서는 높은 Variance를 가지고 있는 High-Dimension Space를 다루고 있기 때문에, 구하기 쉽지 않음.**

Evaluation on Generative Model

- **생성형 모델의 평가 방법**
  - 초기 GAN(Generative Adversarial Network) 논문에서의 평가 방법
  - Likelihood를 구하기 어려우므로, Parzen Window 방법을 활용

Model	MNIST	TFD
DBN [3]	138 ± 2	1909 ± 66
Stacked CAE [3]	121 ± 1.6	$\mathbf{2110 \pm 50}$
Deep GSN [6]	214 ± 1.1	1890 ± 29
Adversarial nets	$\mathbf{225 \pm 2}$	$\mathbf{2057 \pm 26}$

Table 1: Parzen window-based log-likelihood estimates. The reported numbers on MNIST are the mean log-likelihood of samples on test set, with the standard error of the mean computed across examples. On TFD, we computed the standard error across folds of the dataset, with a different σ chosen using the validation set of each fold. On TFD, σ was cross validated on each fold and mean log-likelihood on each fold were computed. For MNIST we compare against other models of the real-valued (rather than binary) version of dataset.

Evaluation on Generative Model

- **생성형 모델의 평가 방법**
  - 하지만 이러한 방법이 완벽하지 않음 (오히려 사용 권장하지 않음)

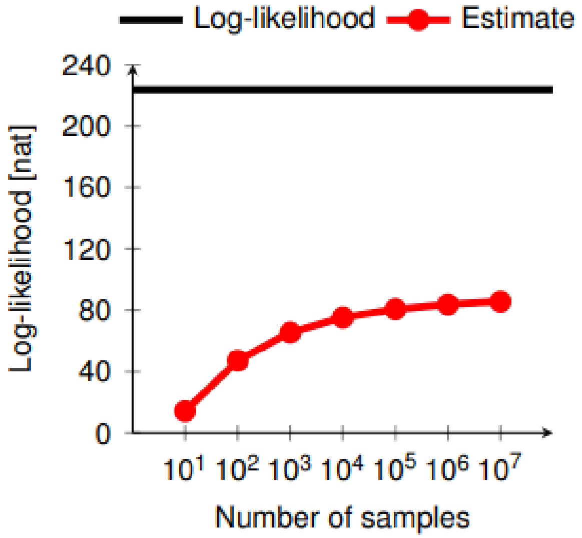


Figure 3: Parzen window estimates for a Gaussian evaluated on 6 by 6 pixel image patches from the CIFAR-10 dataset. Even for small patches and a very large number of samples, the Parzen window estimate is far from the true log-likelihood.

Model	Parzen est. [nat]
Stacked CAE	121
DBN	138
GMMN	147
Deep GSN	214
Diffusion	220
GAN	225
True distribution	**243**
GMMN + AE	282
k-means	313

Table 1: Using Parzen window estimates to evaluate various models trained on MNIST, samples from the true distribution perform worse than samples from a simple model trained with k-means.

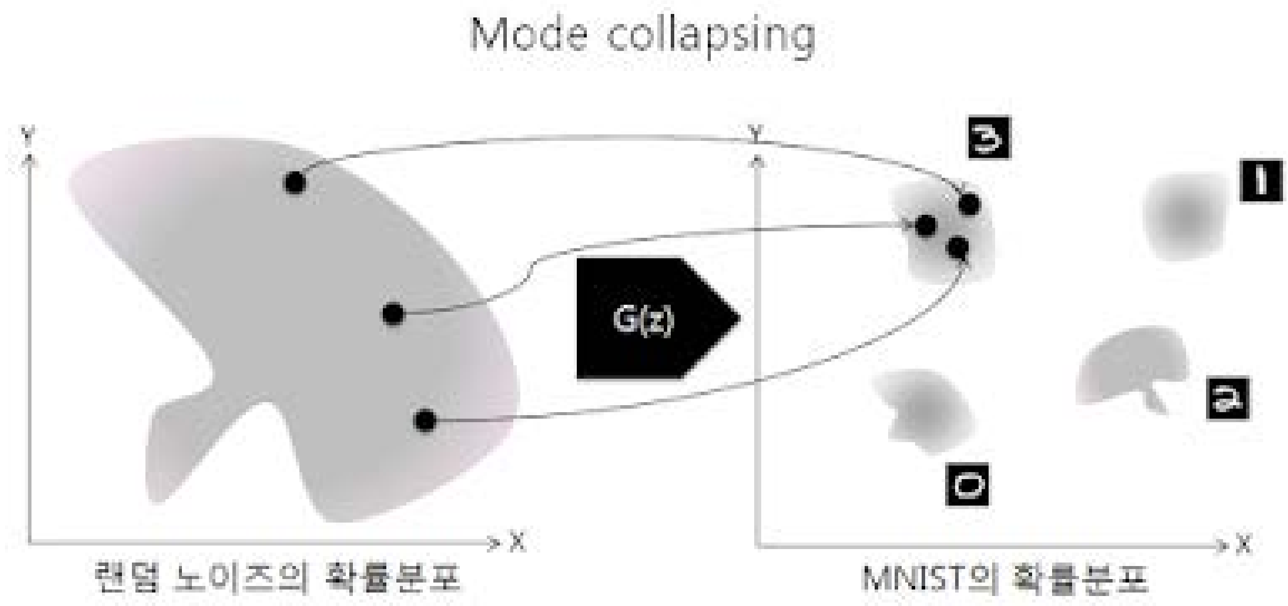
Evaluation on Generative Model

Generative Model
Auto-Encoder
Convolutional Auto-Encoder
Variational Auto-Encoder
SVHN

- **생성형 모델의 평가 방법**
  - 남은 것은 가장 쉽게 샘플들을 확인하는 것

Evaluation on Generative Model

- **생성형 모델의 평가 방법**
  - 남은 것은 가장 쉽게 샘플들을 확인하는 것
  - **하지만 주의해야할 점이 많음**
    - 학습 이미지(Training Data) 중 하나를 출력한다면?
    - 학습 이미지 중 가장 많이 나오는 이미지들의 평균을 보여준다면? (Mode Collapse)



http://dl-ai.blogspot.com/2017/08/gan-problems.html

실 습