with tensorflow

구현을 위한 딥러닝

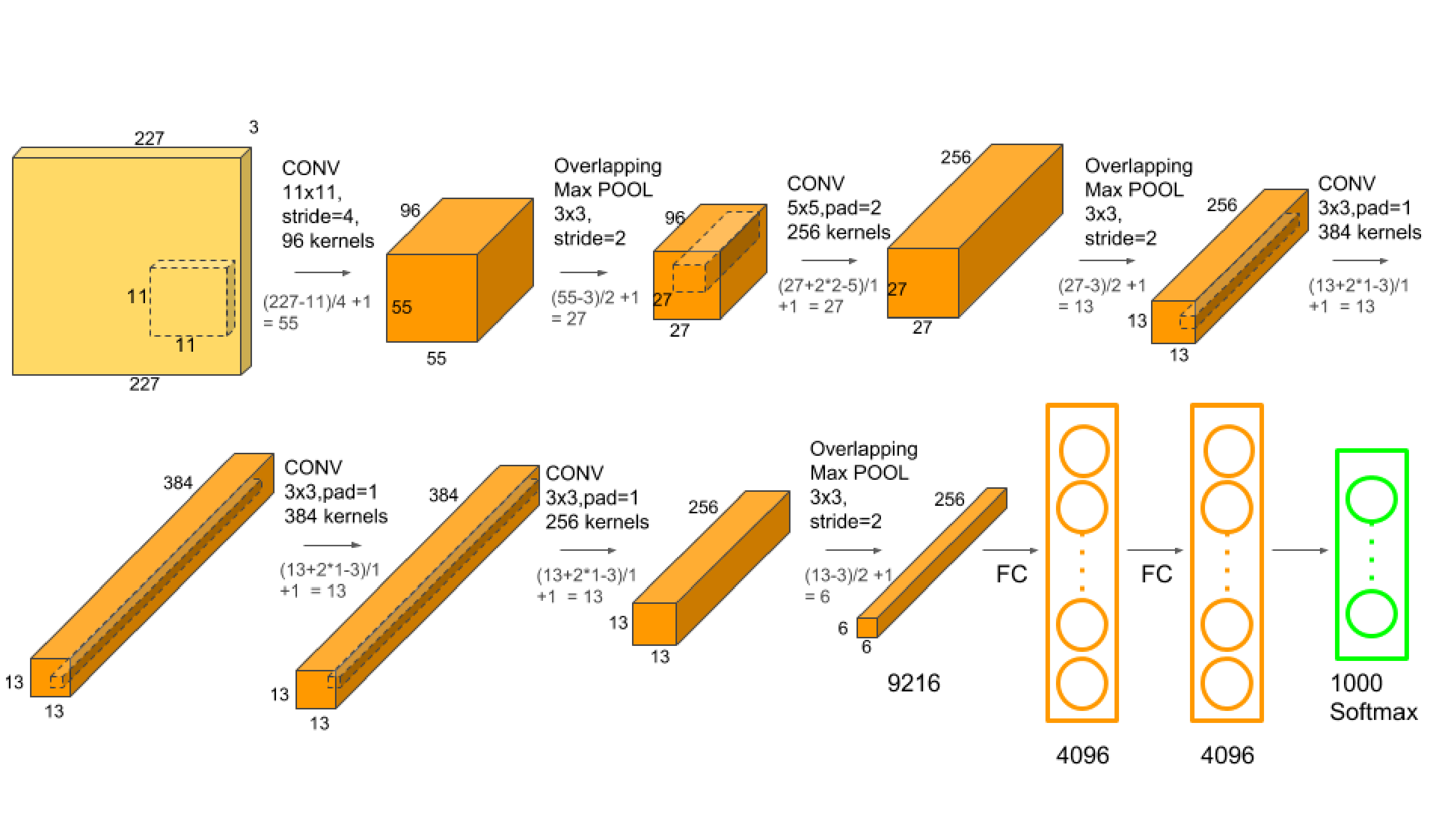
고려대학교 물리학과 한승희

mod96@naver.com

Contents

Contents

Example



227
3
11
11
227
CONV
11x11,
stride=4,
96 kernels
(227-11)/4 +1
= 55
96
55
55
Overlapping
Max POOL
3x3,
stride=2
(55-3)/2 +1
= 27
96
27
27
CONV
5x5,pad=2
256 kernels
(27+2*2-5)/1
+1 = 27
256
27
27
Overlapping
Max POOL
3x3,
stride=2
(27-3)/2 +1
= 13
256
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
256 kernels
(13+2*1-3)/1
+1 = 13
256
13
13
Overlapping
Max POOL
3x3,
stride=2
(13-3)/2 +1
= 6
256
6
6
9216
FC
4096
FC
4096
1000
Softmax

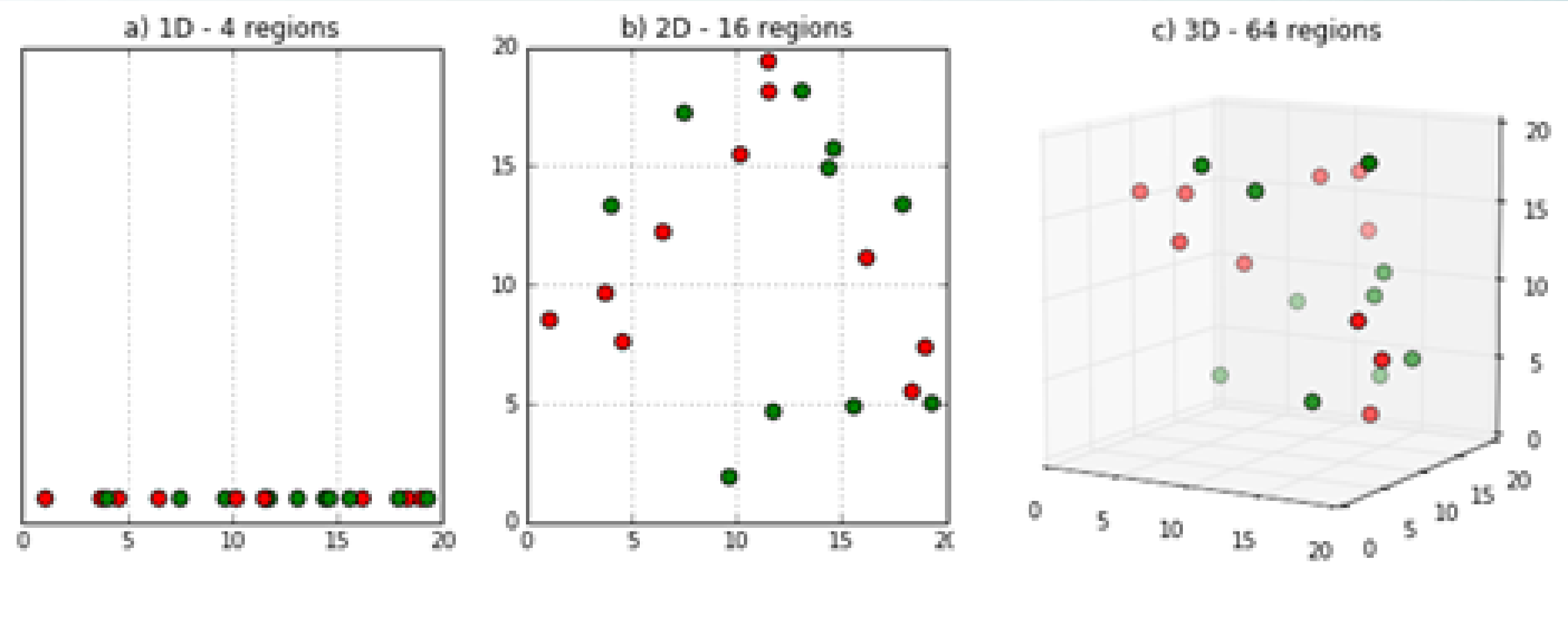
Example

Contents

Curse of Dimensionality

- Model becomes complex
- Need exponentially many data
- Metric(distance) goes wrong

Need exponentially many data



Metric goes wrong

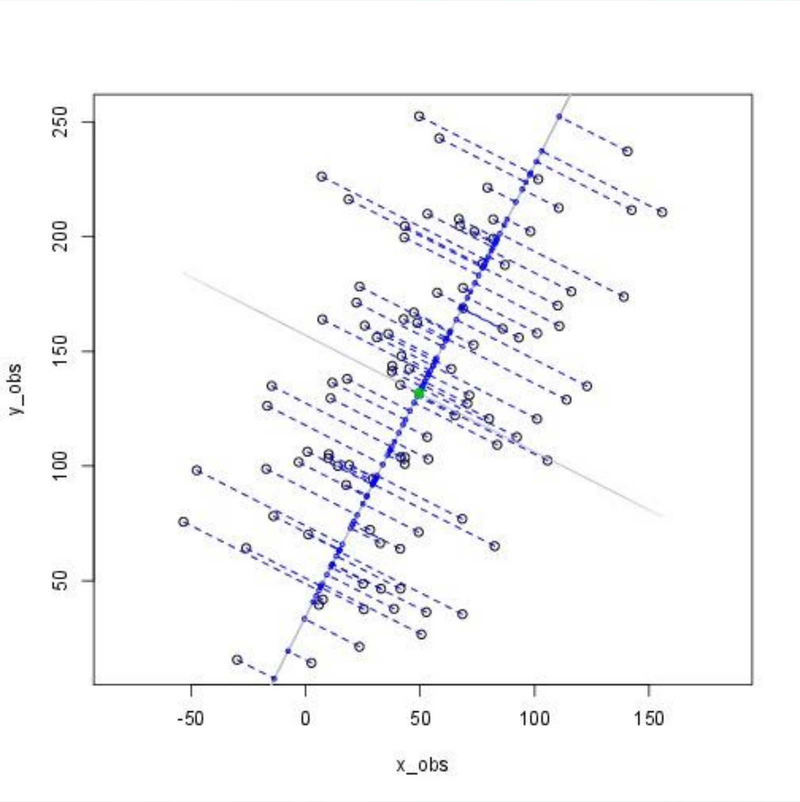
- In D-dimensional space,

Contents

Dimension Reduction

- Traditional Algorithms
  - PCA
  - LDA
  - ICA
  - CCA
  - LLE
  - MDS
  - Isomap
  - t-SNE

PCA

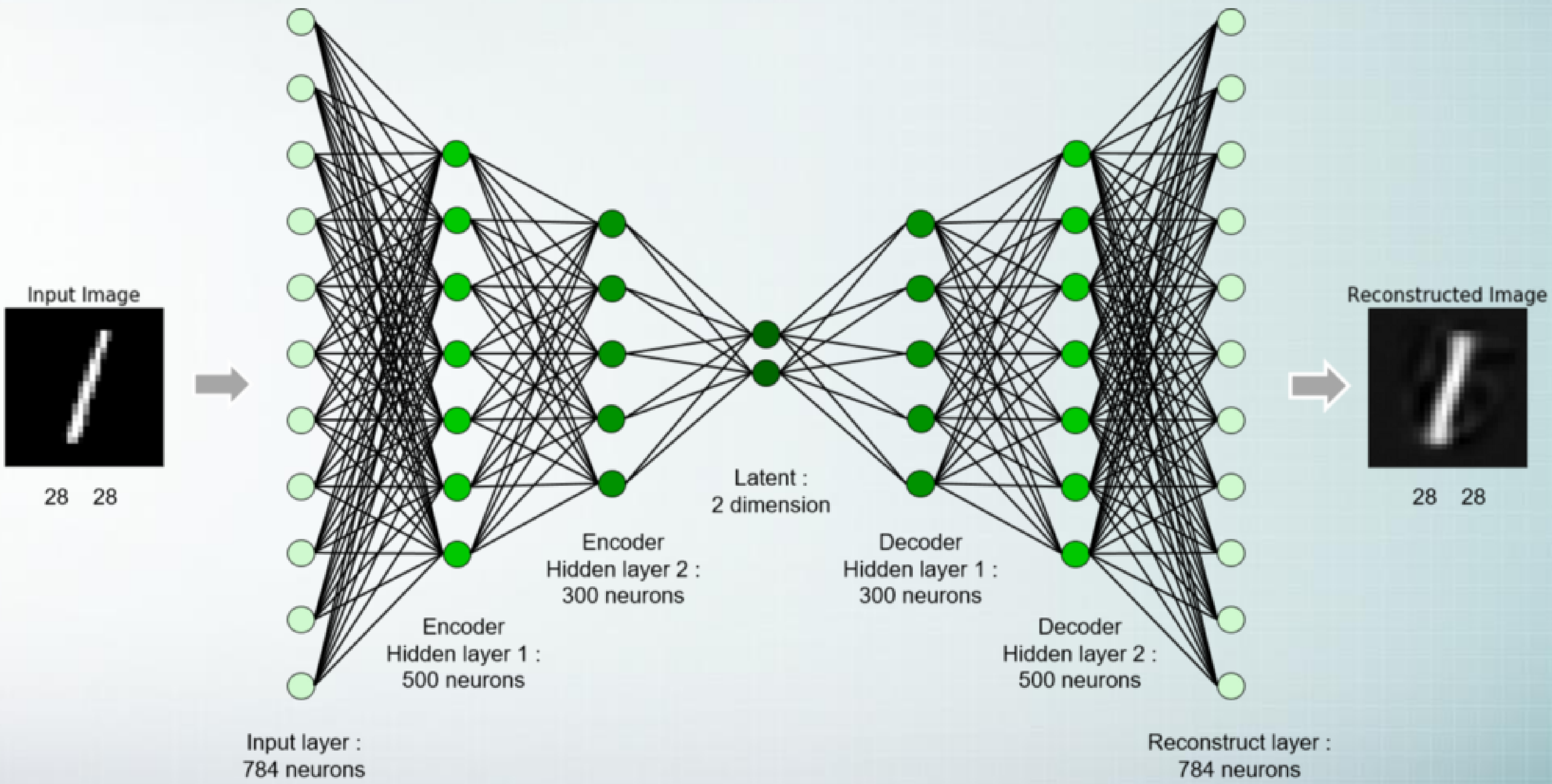


❖ PCA defines a new orthogonal coordinate system that optimally describes variance in a single dataset.

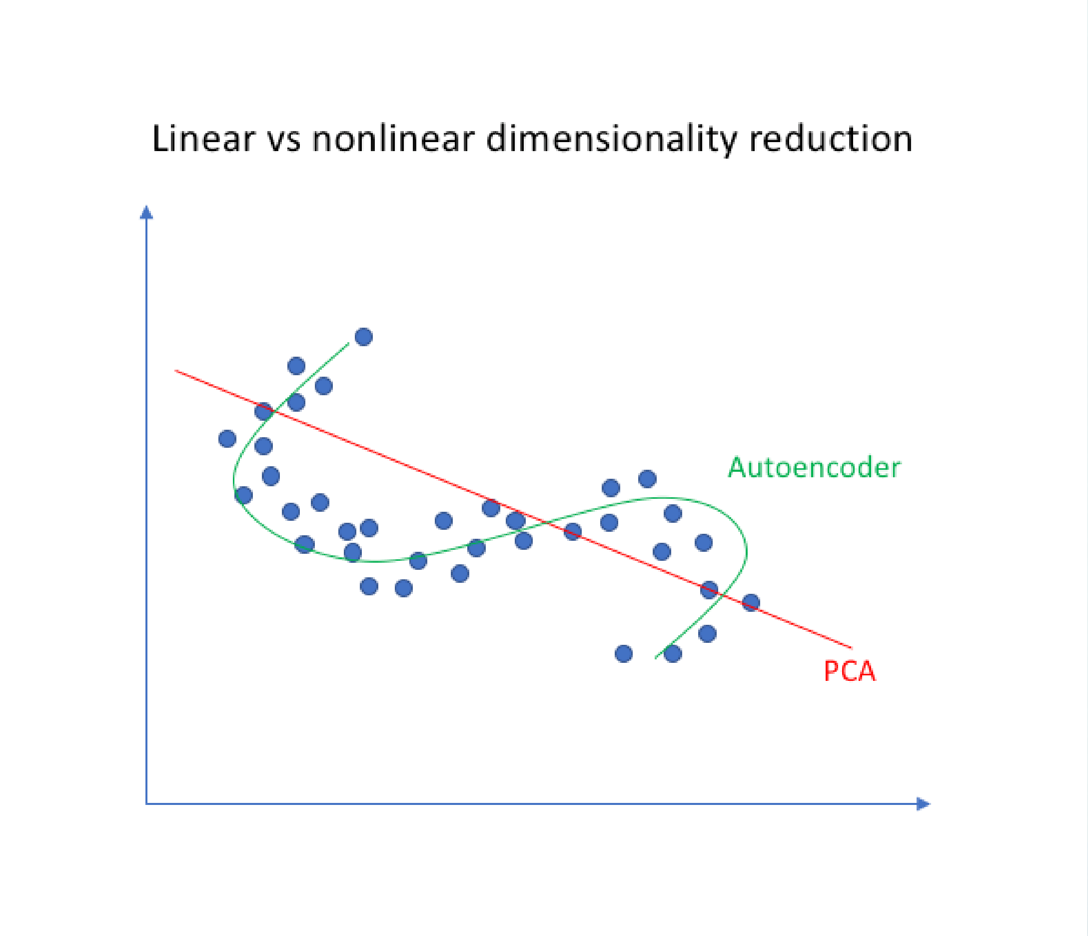
PCA

Contents

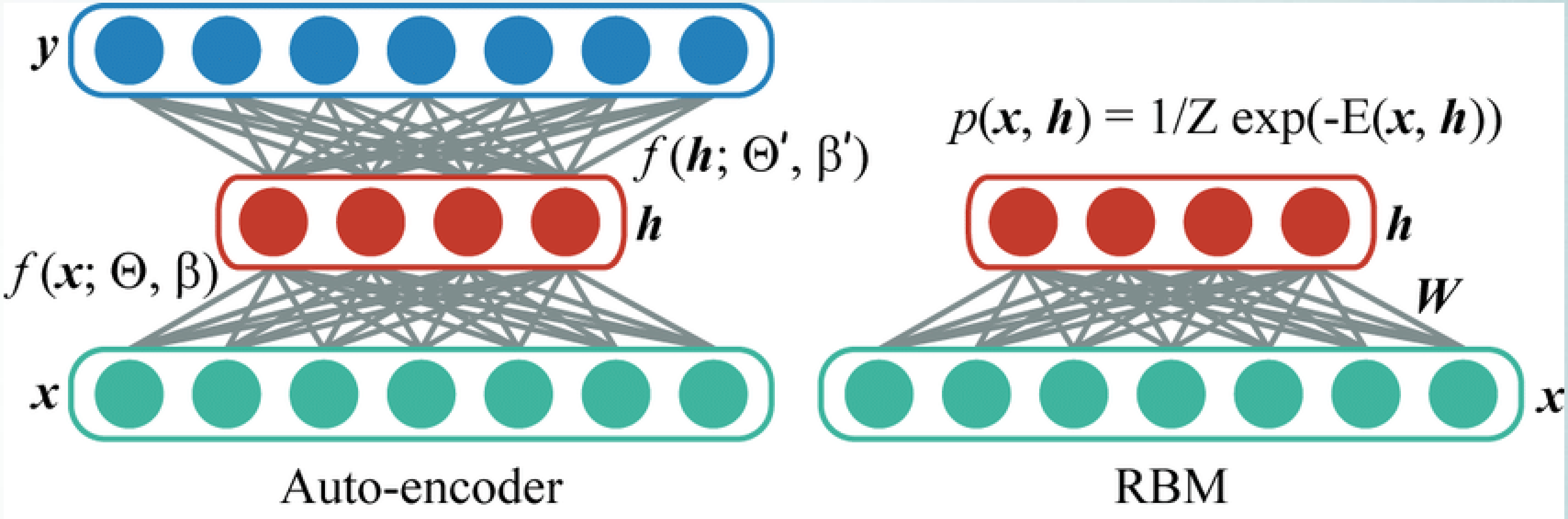
AE


Input Image
28 28
Input layer :
784 neurons
Encoder
Hidden layer 1 :
500 neurons
Encoder
Hidden layer 2 :
300 neurons
Latent :
2 dimension
Decoder
Hidden layer 1 :
300 neurons
Decoder
Hidden layer 2 :
500 neurons
Reconstruct layer :
784 neurons
Reconstructed Image
28 28

AE vs PCA

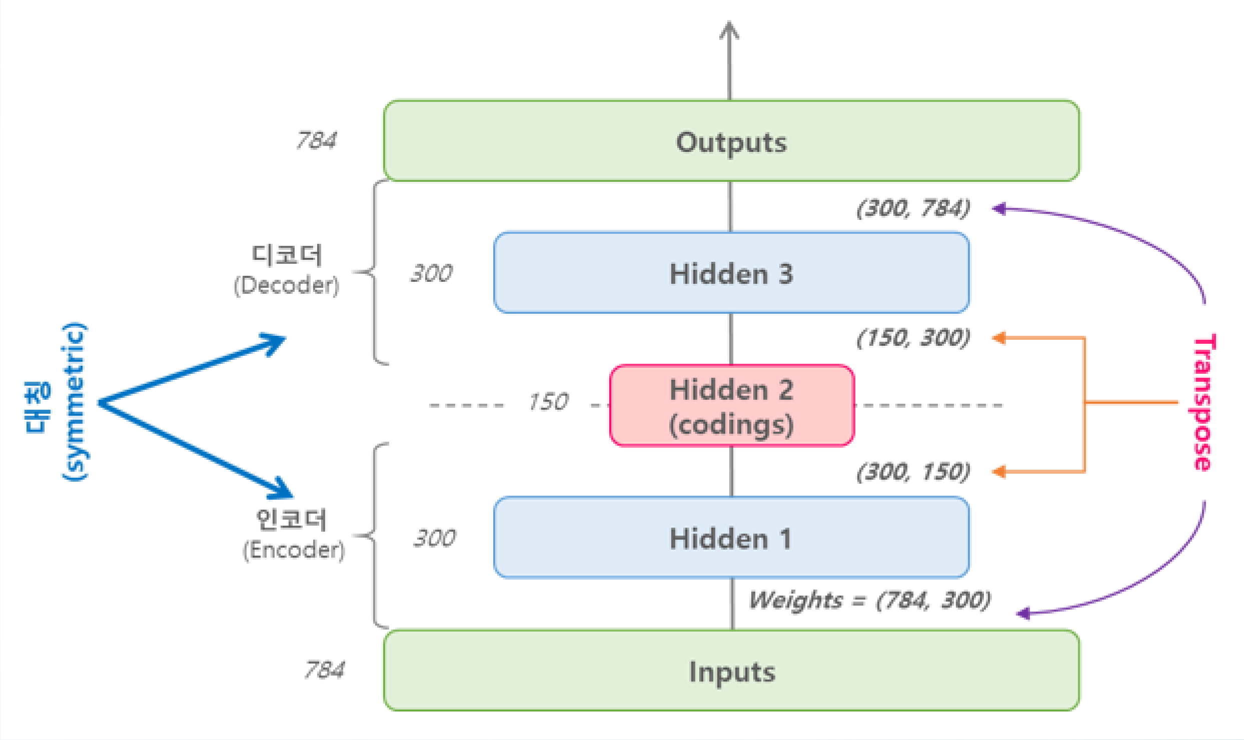

Linear vs nonlinear dimensionality reduction
Autoencoder
PCA

AE vs RBM*



* RBM : Restricted Boltzmann Machine

AE...Conv2DTranspose?



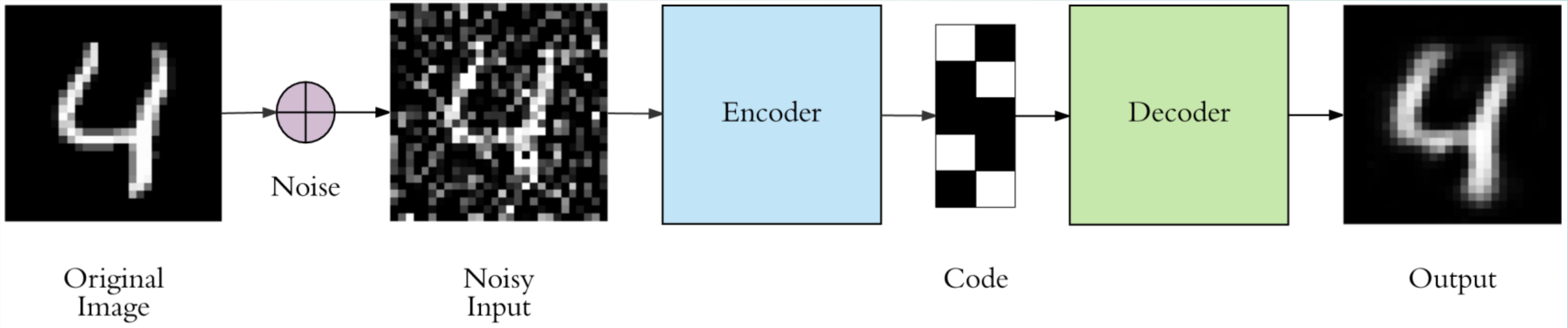
https://zzsza.github.io/data/2018/06/25/upsampling-with-transposed-convolution/

AE...Conv2DTranspose?

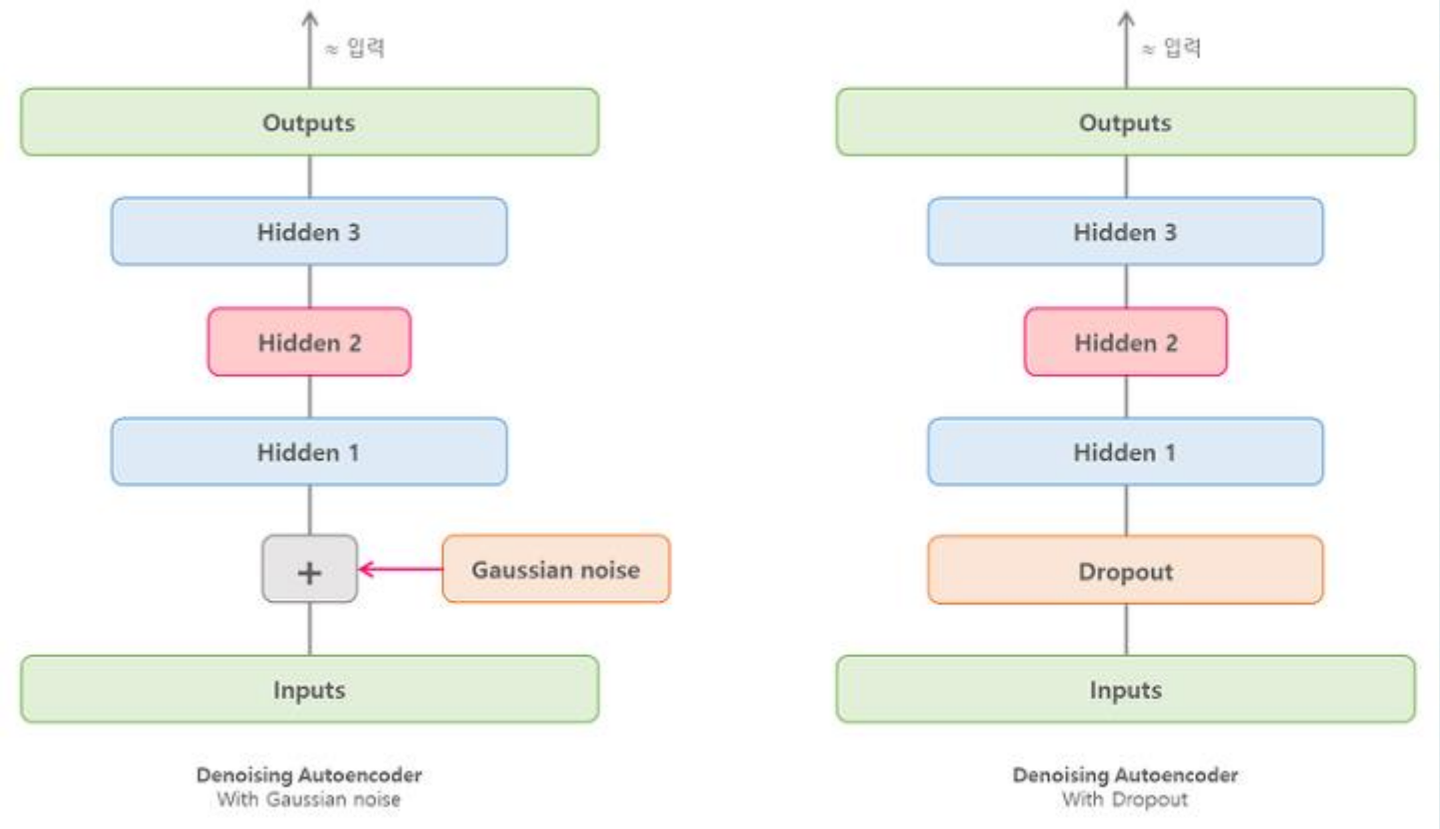
$$\begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} * \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_5 & x_6 & x_7 & x_8 \\ x_9 & x_{10} & x_{11} & x_{12} \\ x_{13} & x_{14} & x_{15} & x_{16} \end{bmatrix} = \begin{bmatrix} y_1 & y_2 \\ y_3 & y_4 \end{bmatrix}$$

$$\begin{bmatrix} w_1 & w_2 & w_3 & 0 & w_4 & w_5 & w_6 & 0 & w_7 & w_8 & w_9 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_1 & w_2 & w_3 & 0 & w_4 & w_5 & w_6 & 0 & w_7 & w_8 & w_9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_1 & w_2 & w_3 & 0 & w_4 & w_5 & w_6 & 0 & w_7 & w_8 & w_9 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_1 & w_2 & w_3 & 0 & w_4 & w_5 & w_6 & 0 & w_7 & w_8 & w_9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{15} \\ x_{16} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

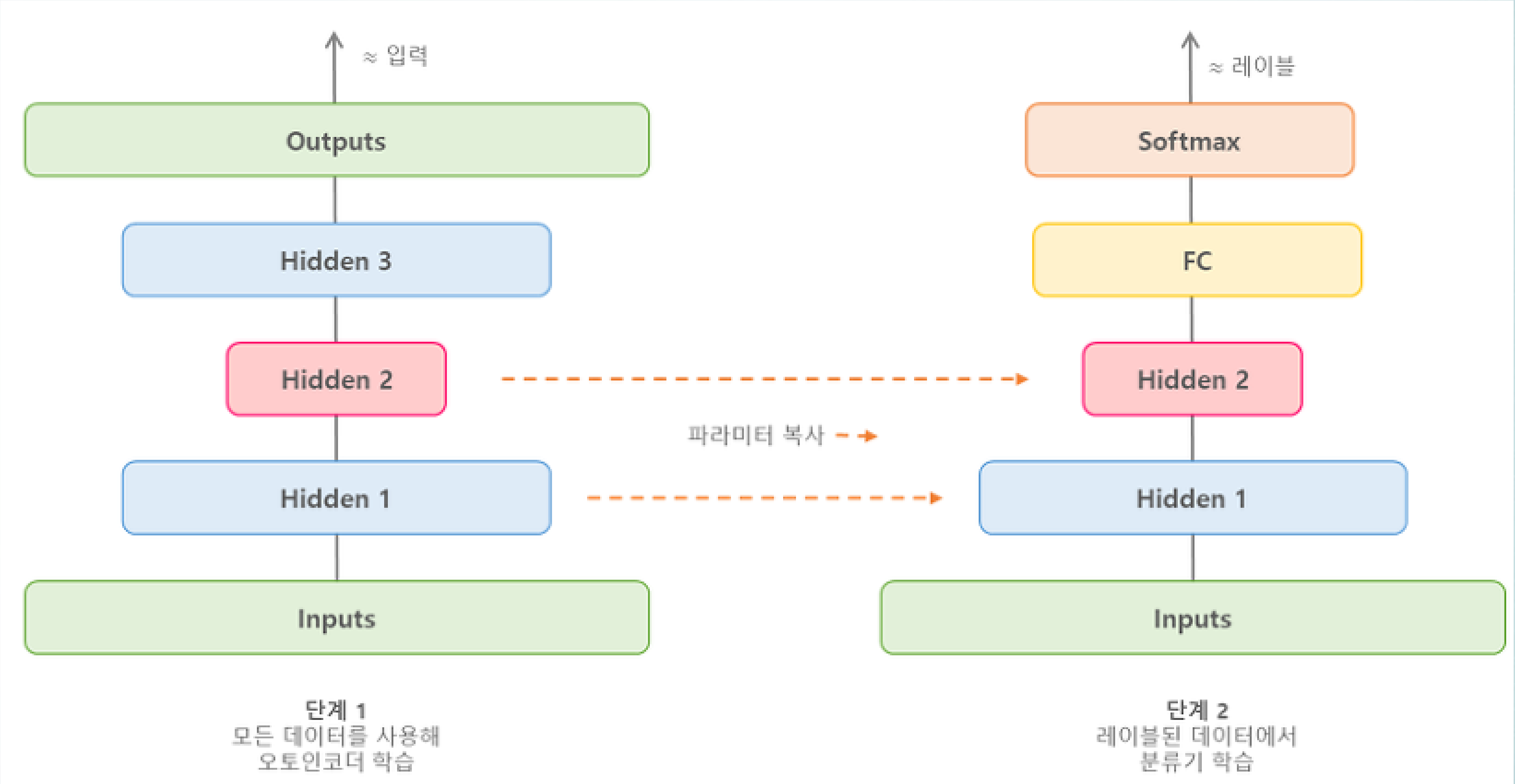
AE – Usage1 : Denoising



AE – Usage1 : Denoising

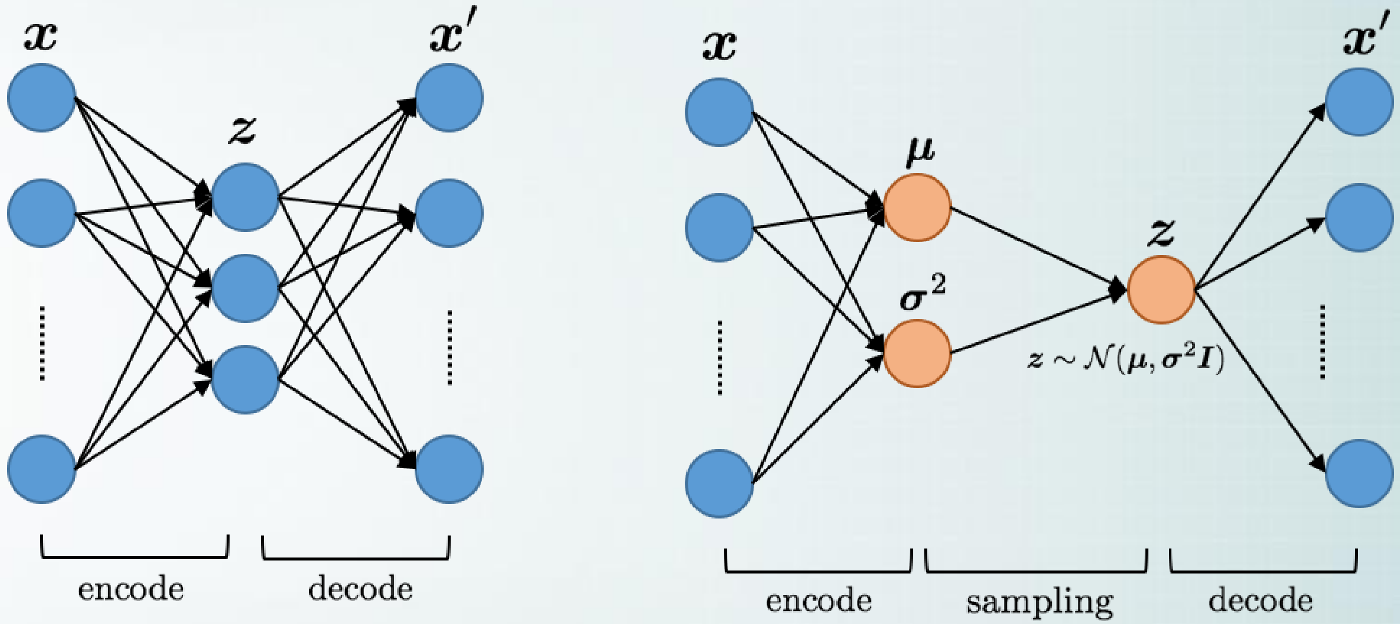


AE – Usage2 : Classification

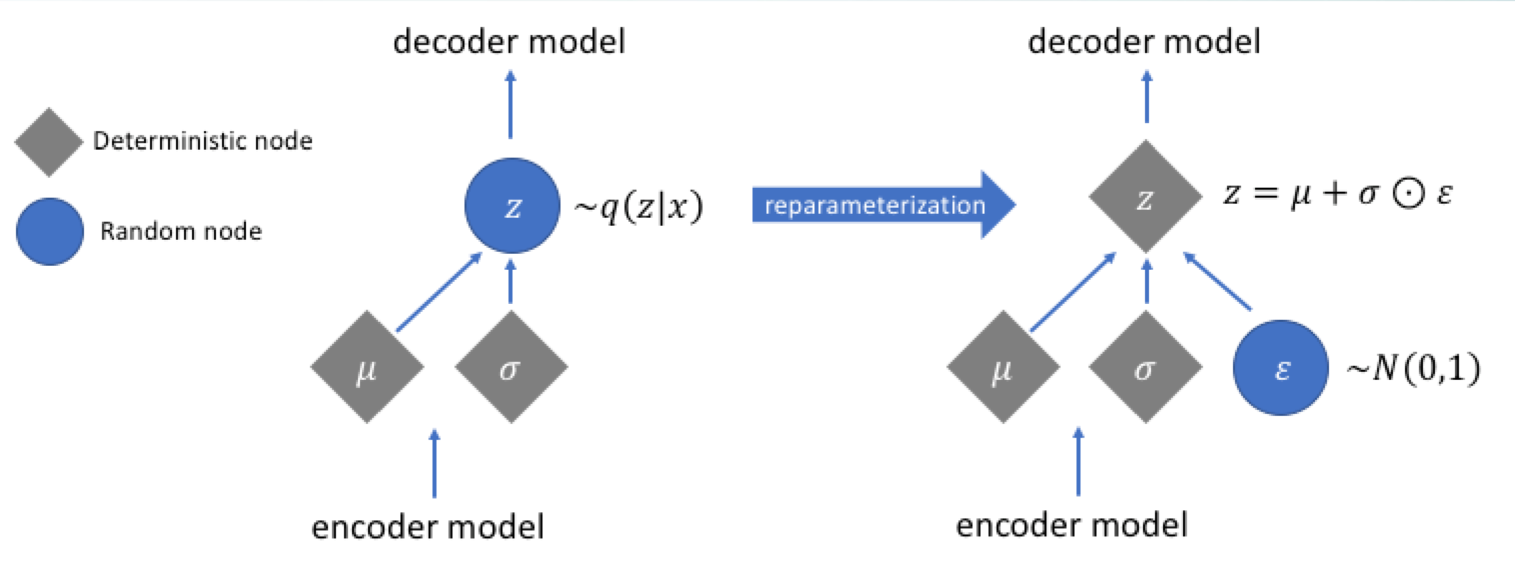


Contents

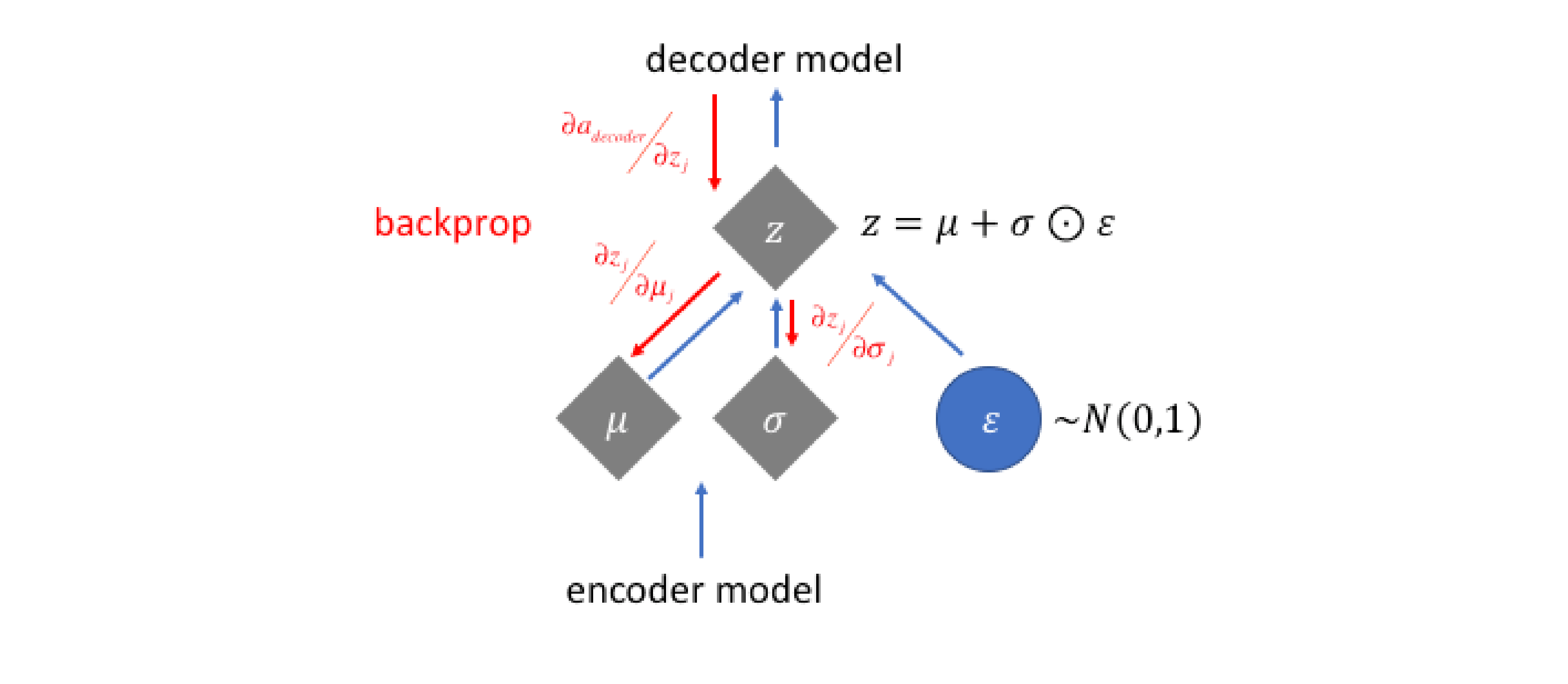
Variational Auto Encoder (VAE)



VAE – latent space sampling



VAE – latent space sampling



VAE – loss function

$$L(\theta,\phi;x^i) = D_{KL}(q_\phi(z|x^i)||p_\theta(z)) - E_{z\sim q_\phi(z|x^i)}[\log p_\theta(x^i|z)]$$

$$\approx -\frac{1}{2}\sum_{j=1}^{J}\left(1+\log(\sigma_j^i)^2-\left(\mu_j^i\right)^2-\left(\sigma_j^i\right)^2\right)+H[q_\phi(z|x^i),\,p_\theta(x^i|z)]$$

Contents

AE

- MNIST

VAE

- MNIST
- CIFAR10