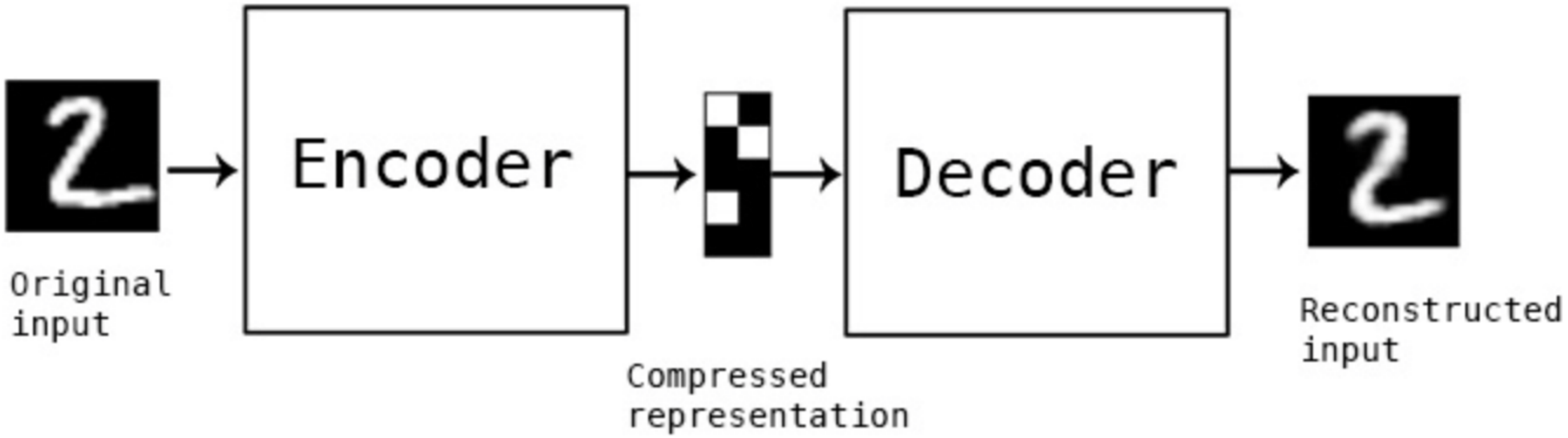
Auto Encoder
Variational Auto Encoder

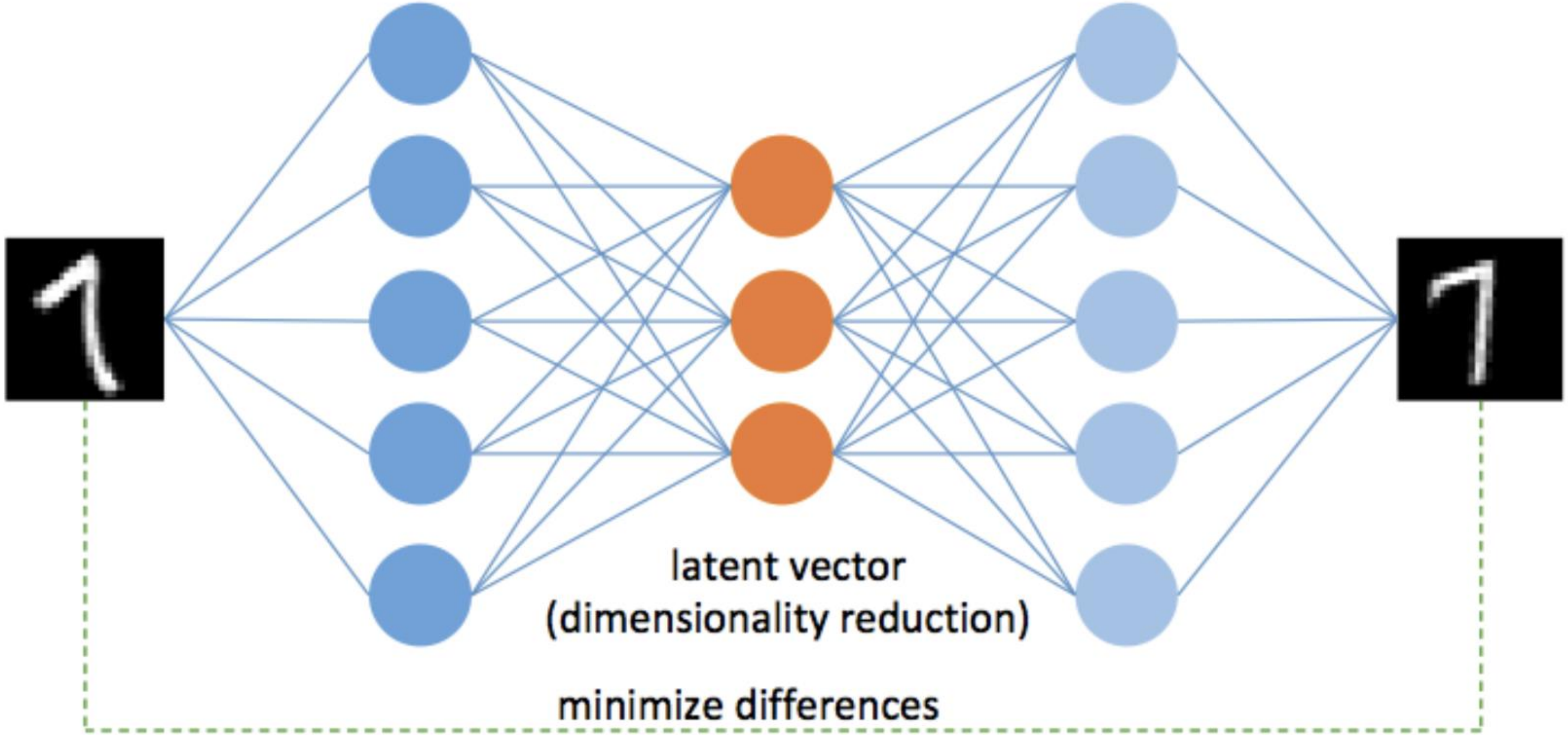
POSTECH A.I

Seungbeom Lee

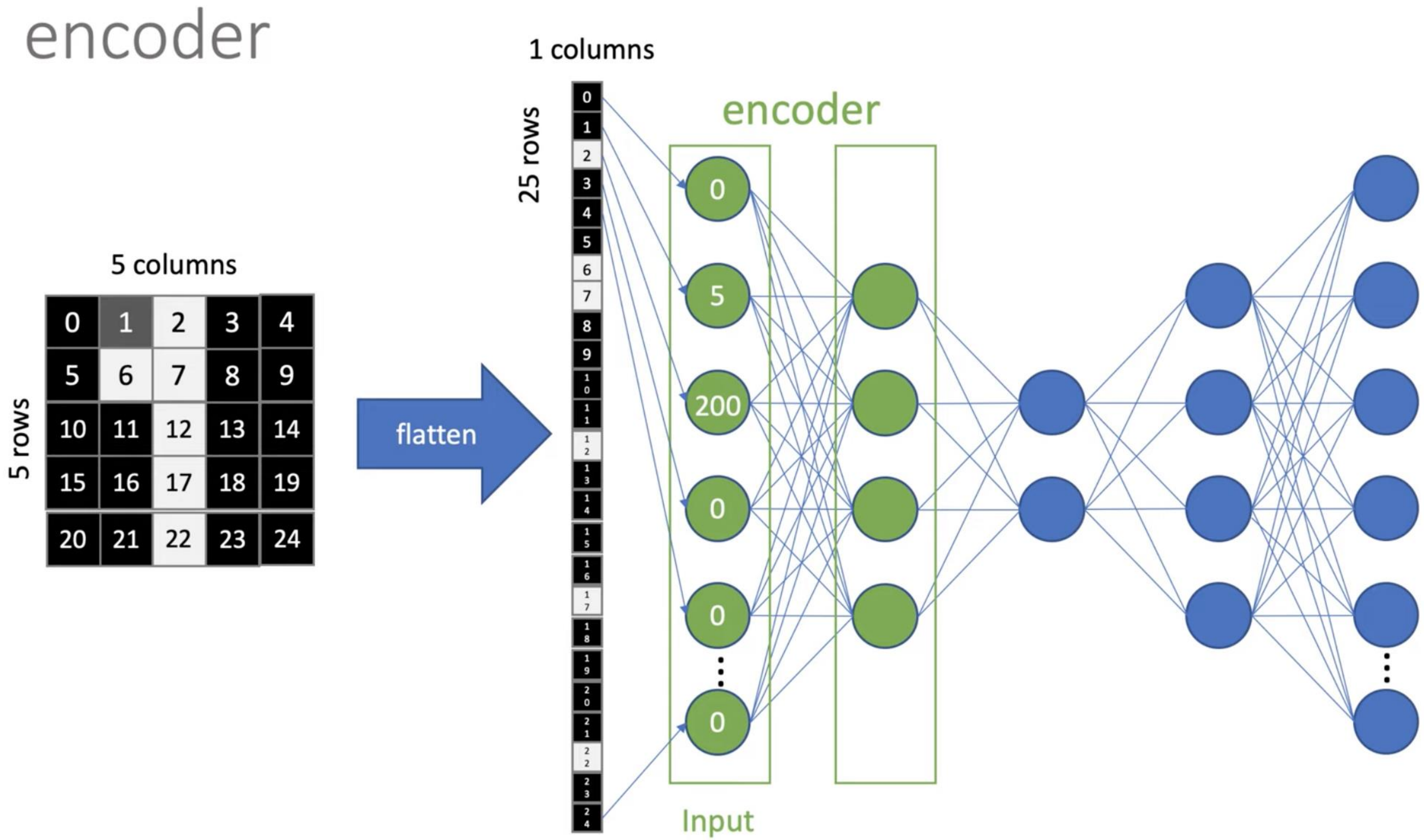
Autoencoder



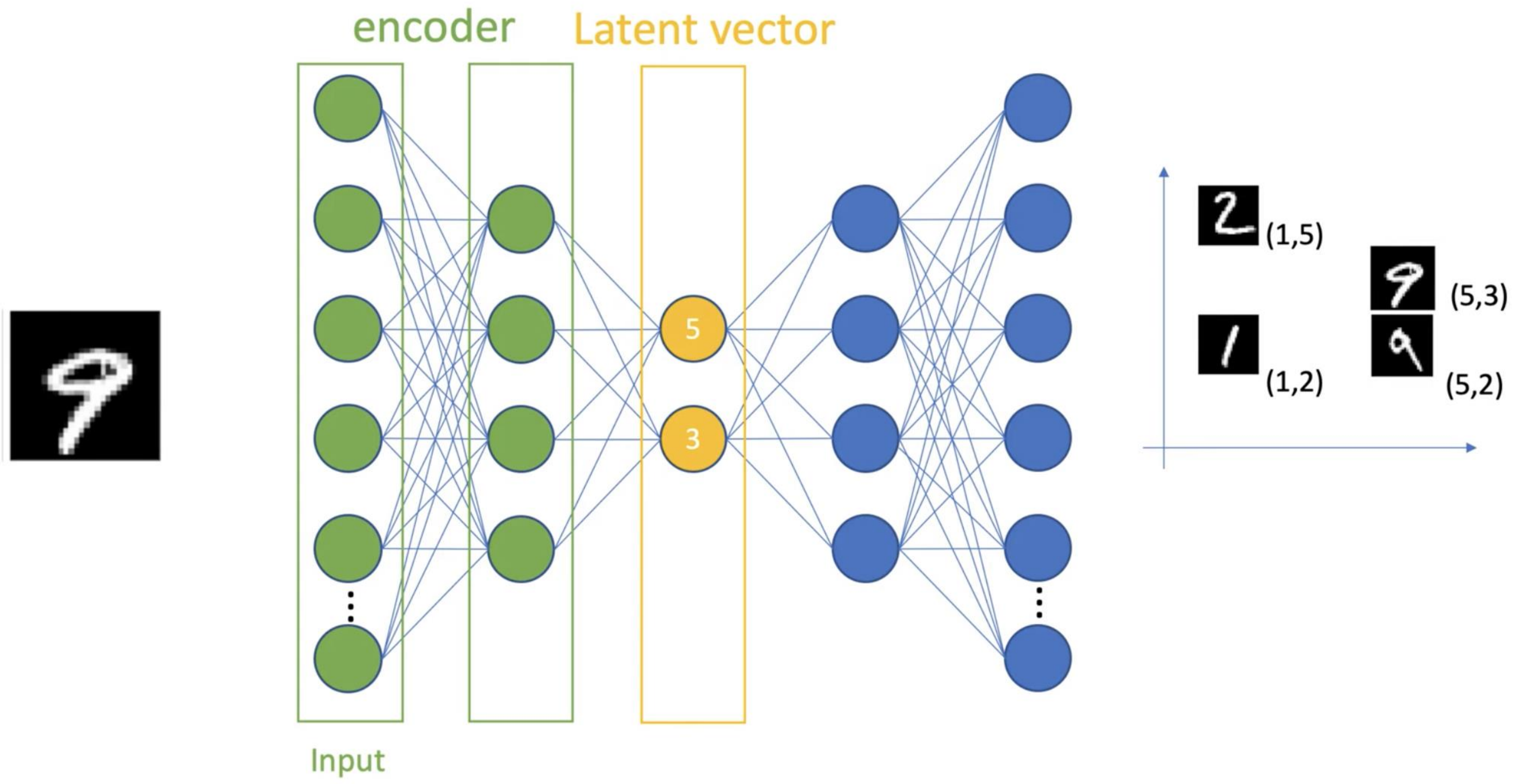
Autoencoder



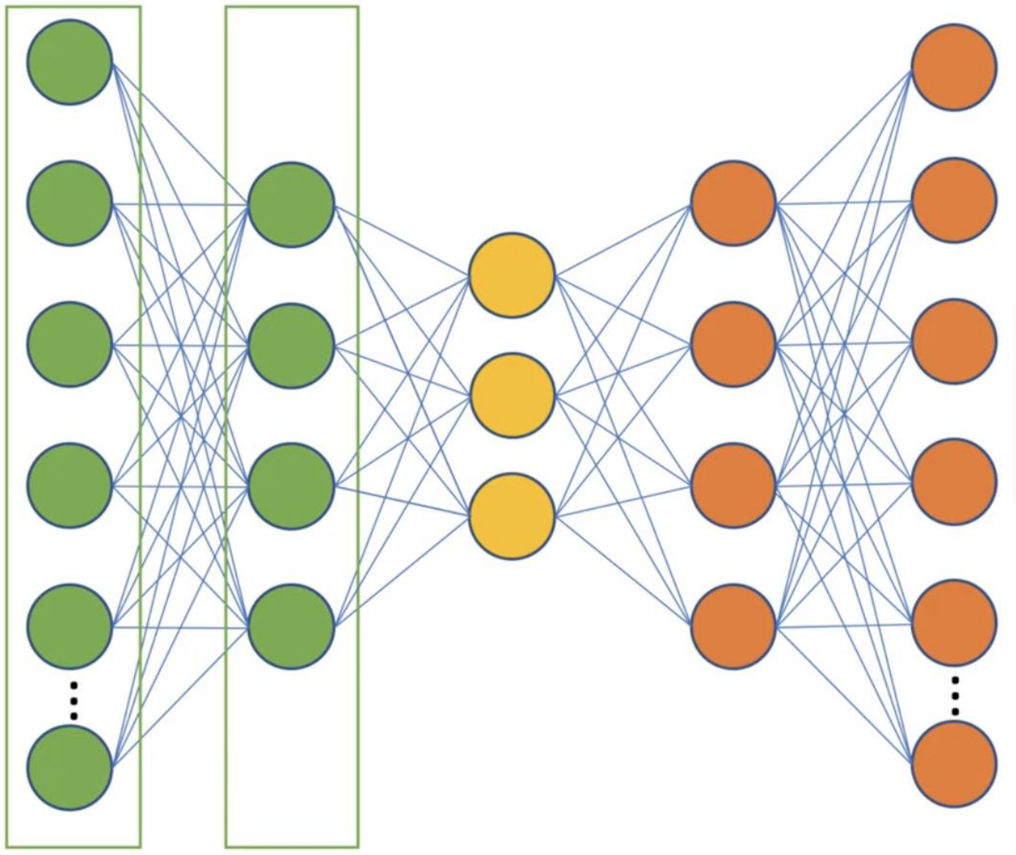
Autoencoder



Autoencoder

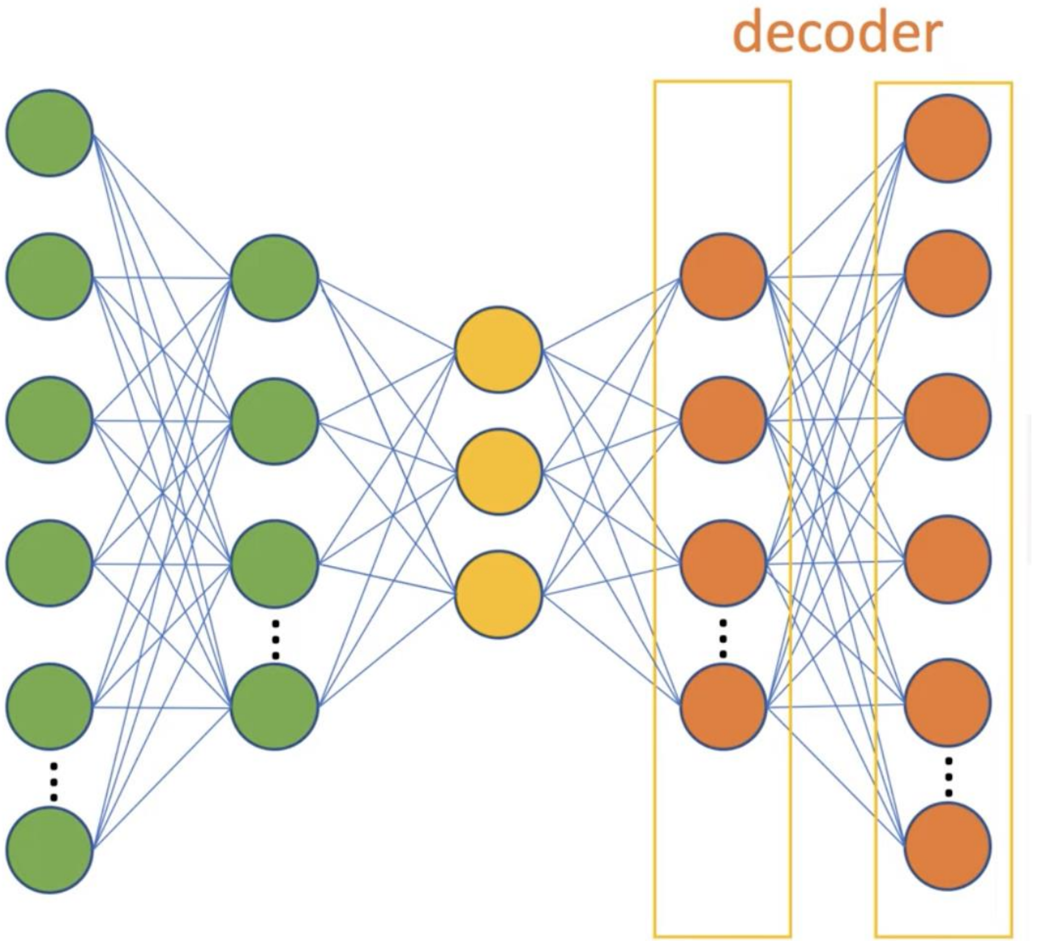


Review the



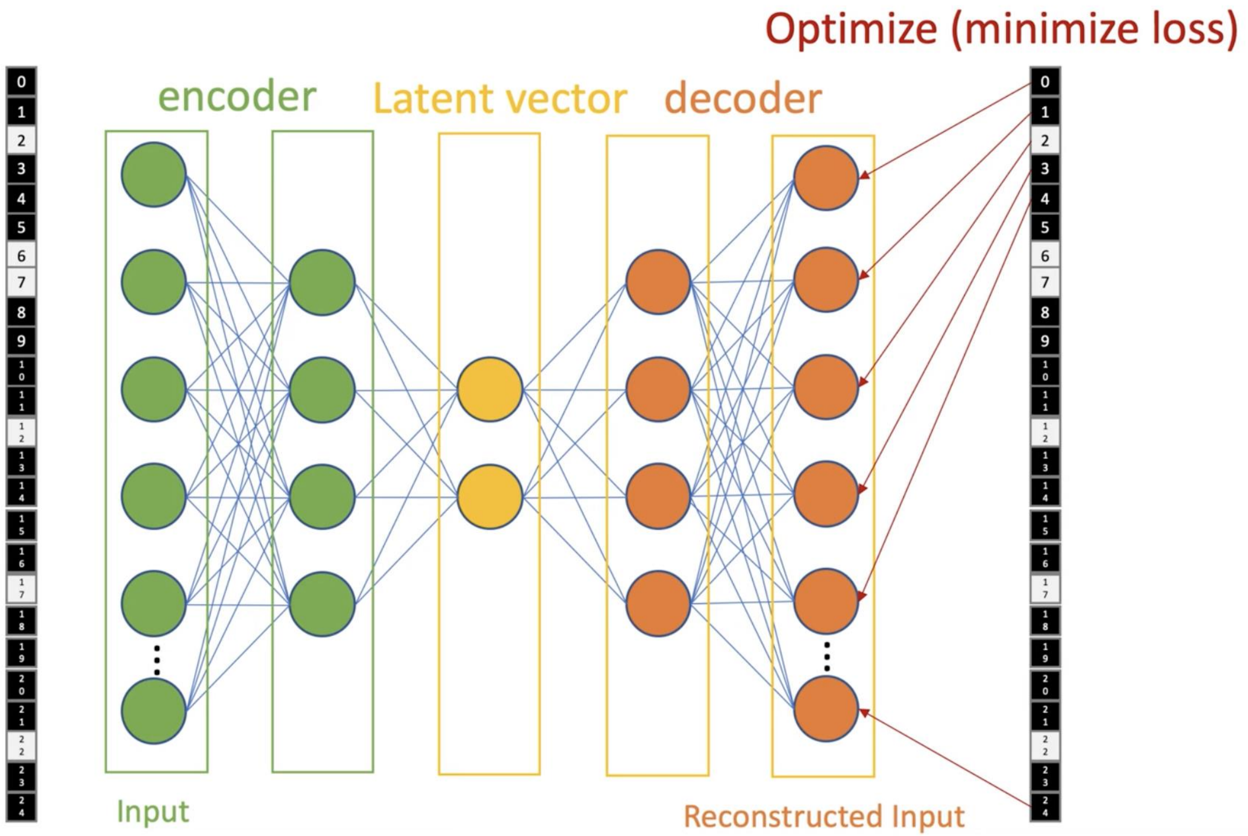
Encoding dimension = 3

```
# MNIST input 28 rows * 28 columns = 784 pixels
input_img = Input(shape=(784,))
# encoder
encoder1 = Dense(128, activation='sigmoid')(input_img)
encoder2 = Dense(3, activation='sigmoid')(encoder1)
```



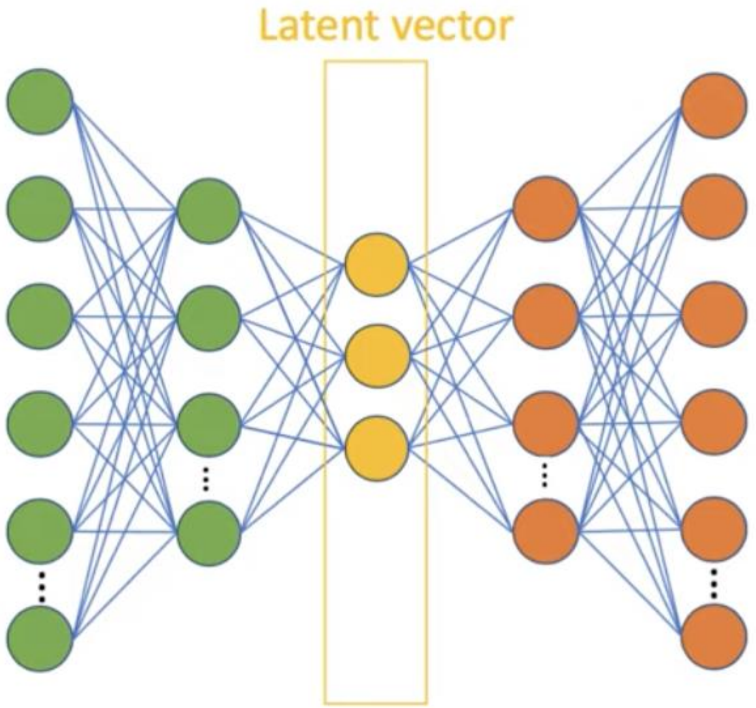

```
# decoder
decoder1 = Dense(128, activation='sigmoid')(encoder2)
decoder2 = Dense(784, activation='sigmoid')(decoder1)
```

decoder




```
autoencoder.compile(optimizer='adam', loss='binary_crossentropy')
```

```
autoencoder.fit(x_train, x_train,
                epochs=5,
                batch_size=32,
                shuffle=True,
                validation_data=(x_test, x_test))
```




```
# create encoder model
encoder = Model(inputs=input_img, outputs=encoder2)

# create decoder model
encoded_input = Input(shape=(3,))
decoder_layer1 = autoencoder.layers[-2]
decoder_layer2 = autoencoder.layers[-1]
decoder = Model(inputs=encoded_input, outputs=decoder_layer2(decoder_layer1(encoded_input)))
```

```python
# get latent vector for visualization
latent_vector = encoder.predict(x_test)


# get decoder output to visualize reconstructed image
reconstructed_imgs = decoder.predict(latent_vector)
```

It just approxiamation for the PCA

Sequence to Sequence Autoencoder

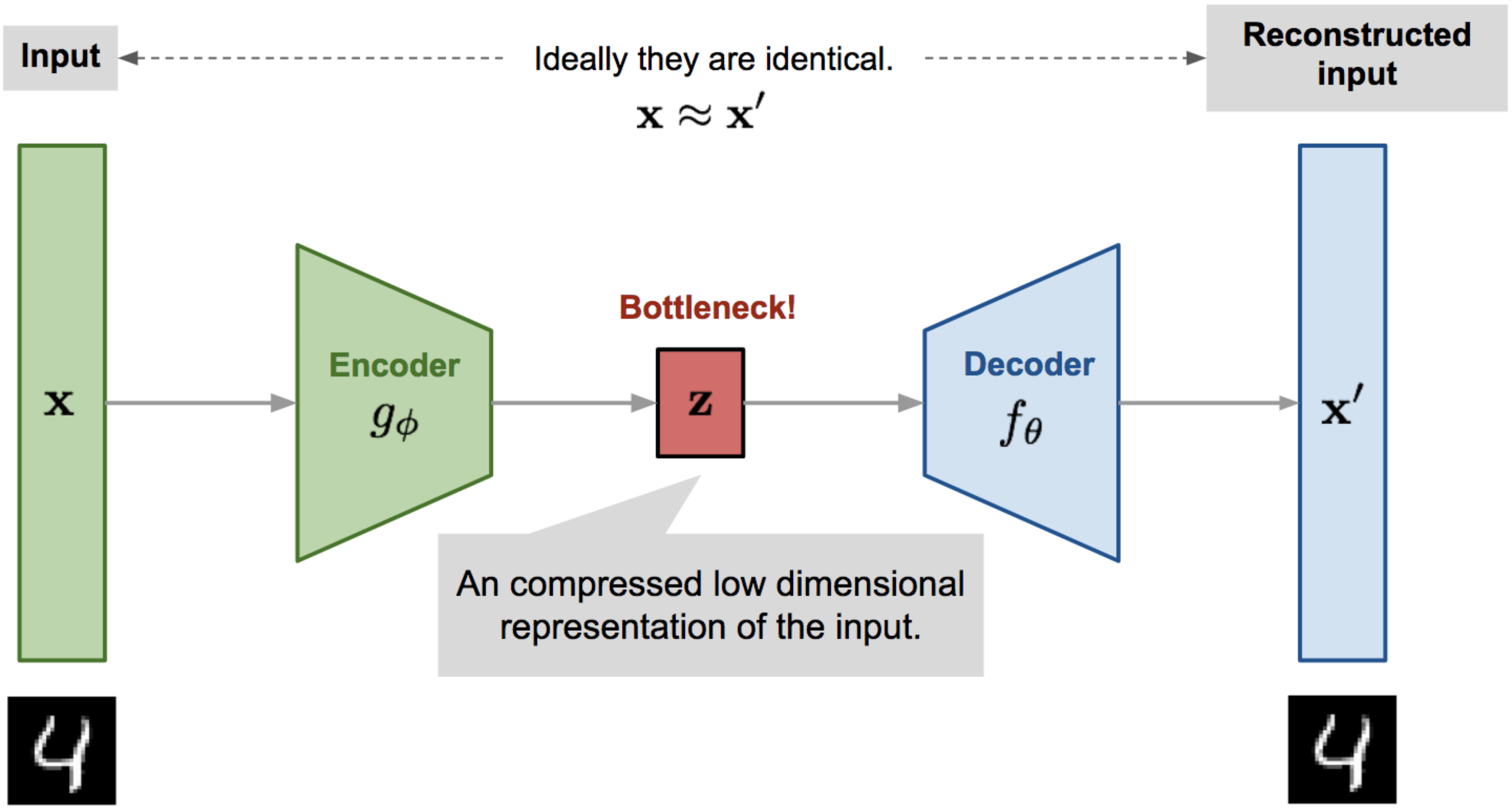
만약 데이터가 연속적인 값이라면? 가능하다!
만약 LSTM encoder를 사용하여 입력 seq를 전체 seq 대한 정보가 들어있는 단일 벡터로 바꾸고 그 벡터를 n 번 반복. (n =timestep) 그리고 이 일정한 seq를 target seq로 바꾸기위해 LSTM deocer 실행

```
from keras.layers import Input, LSTM, RepeatVector
from keras.models import Model

inputs = Input(shape=(timesteps, input_dim))
encoded = LSTM(latent_dim)(inputs)

decoded = RepeatVector(timesteps)(encoded)
decoded = LSTM(input_dim, return_sequences=True)(decoded)

sequence_autoencoder = Model(inputs, decoded)
encoder = Model(inputs, encoded)
```

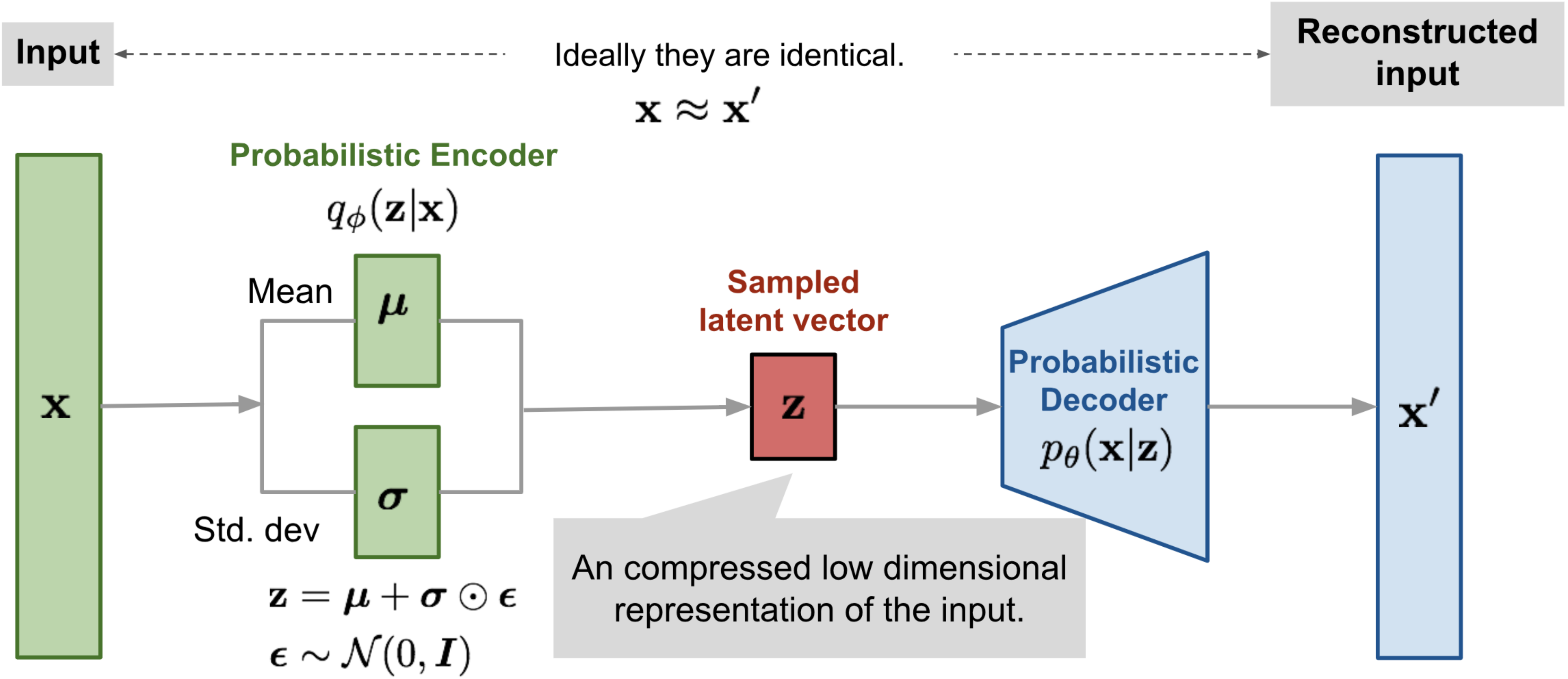
Autoencoder

- Let's practice
  - Basic Autoencoder
  - Autoencoder with CNN
  - Image denoising

Variational Auto Encoder



Variational Auto Encoder



Variational Autoencoders

Now equipped with our encoder and decoder networks, let's work out the (log) data likelihood:

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Bayes' Rule}) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad (\text{Multiply by constant}) \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Logarithms}) \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))
\end{aligned}
$$

We want to maximize the data likelihood

Decoder network gives $p_\theta(x|z)$, can compute estimate of this term through sampling. (Sampling differentiable through reparam. trick, see paper.)

This KL term (between Gaussians for encoder and z prior) has nice closed-form solution!

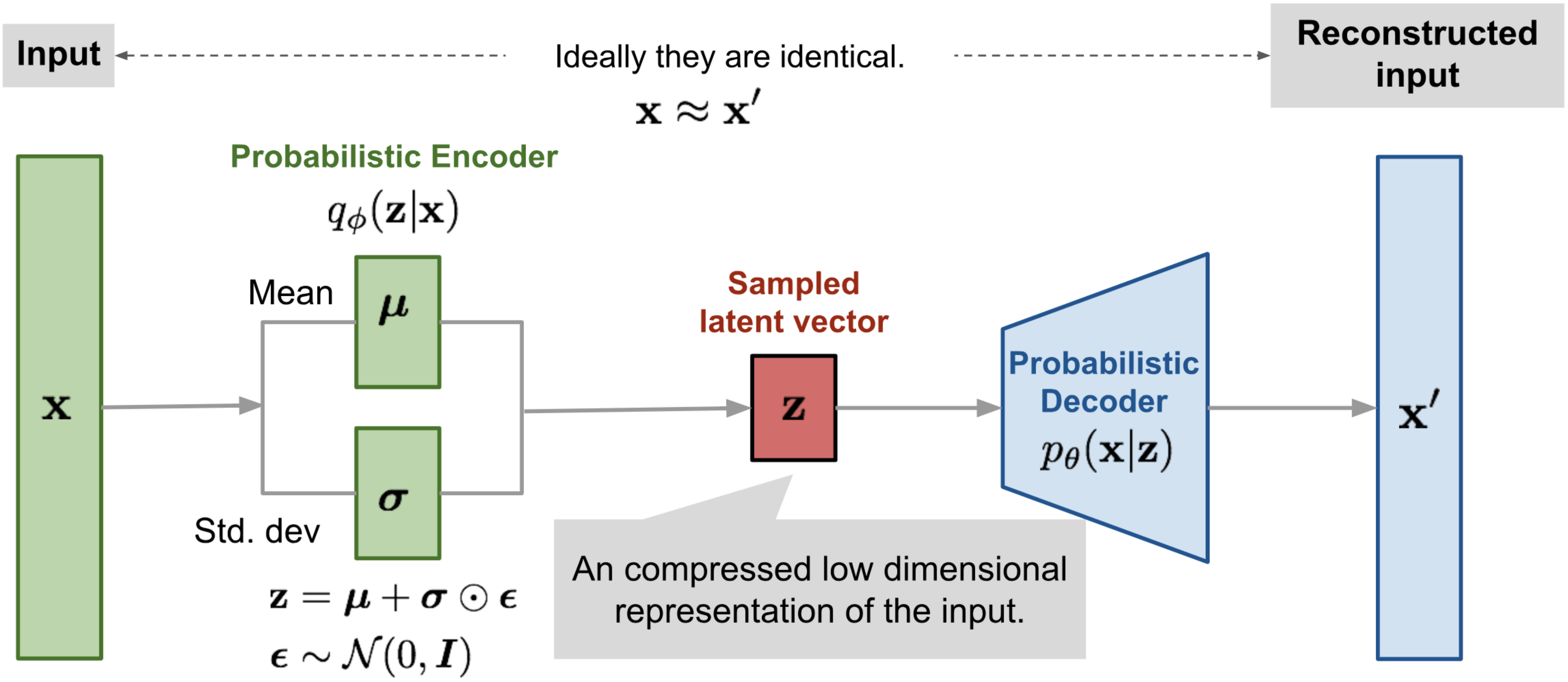
$p_\theta(z|x)$ intractable (saw earlier), can't compute this KL term :(But we know KL divergence always >= 0.

Variational Auto Encoder

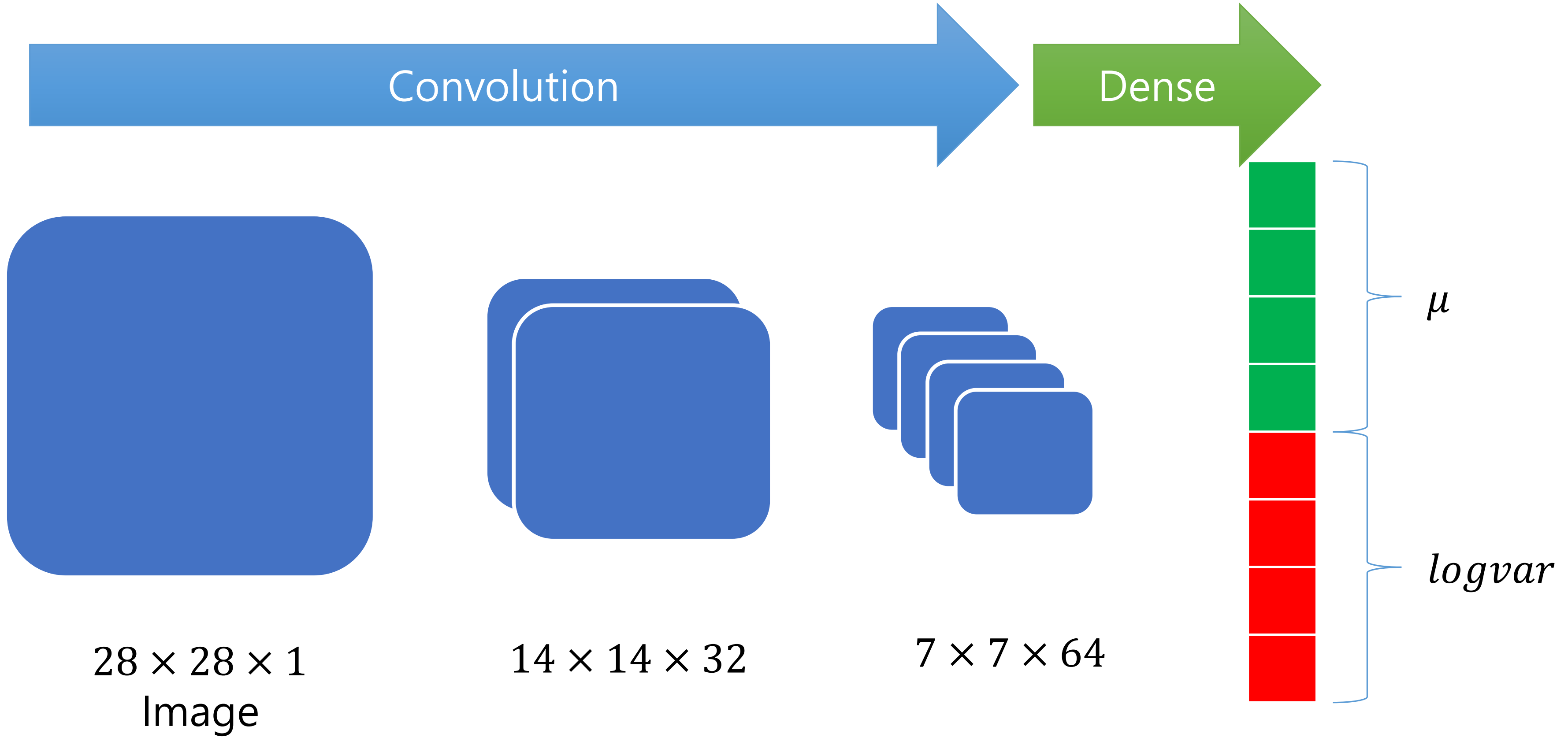
Loss function (ELBO)

$$Loss = \frac{1}{L}\sum_{i=1}^{L}(log(p_\theta(x|z^i))) + 0.5(\mu^2 + \sigma^2 - log(\sigma^2) - 1)$$

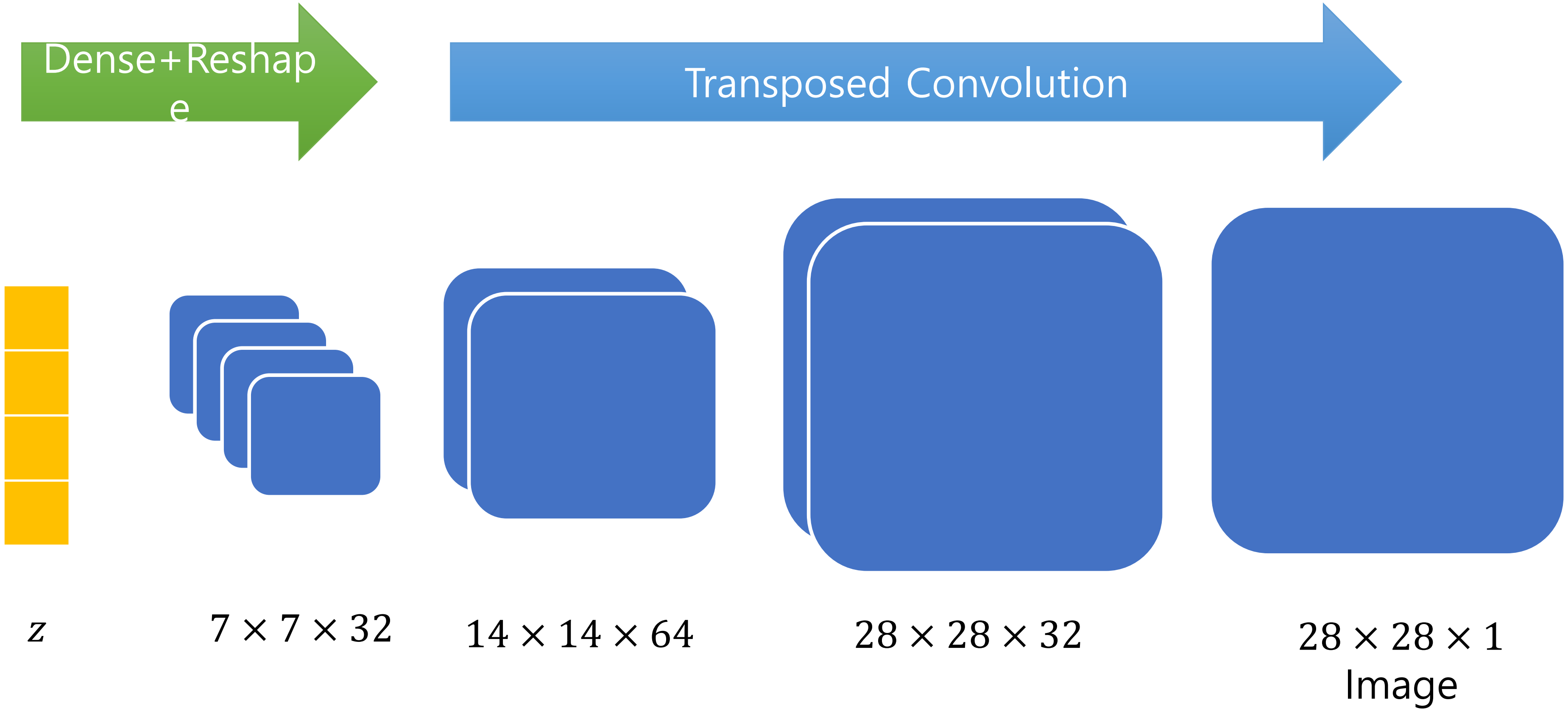
Variational Auto Encoder



Encoder structure



Decoder structure



Thank You :)

slee2020@postech.ac.kr