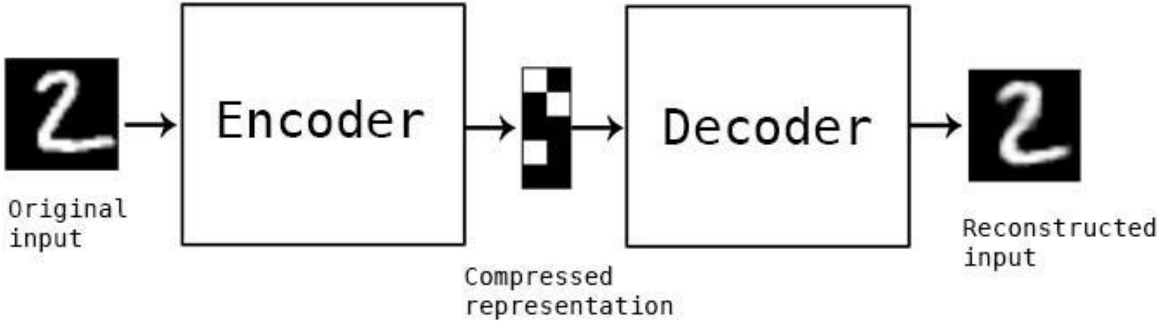
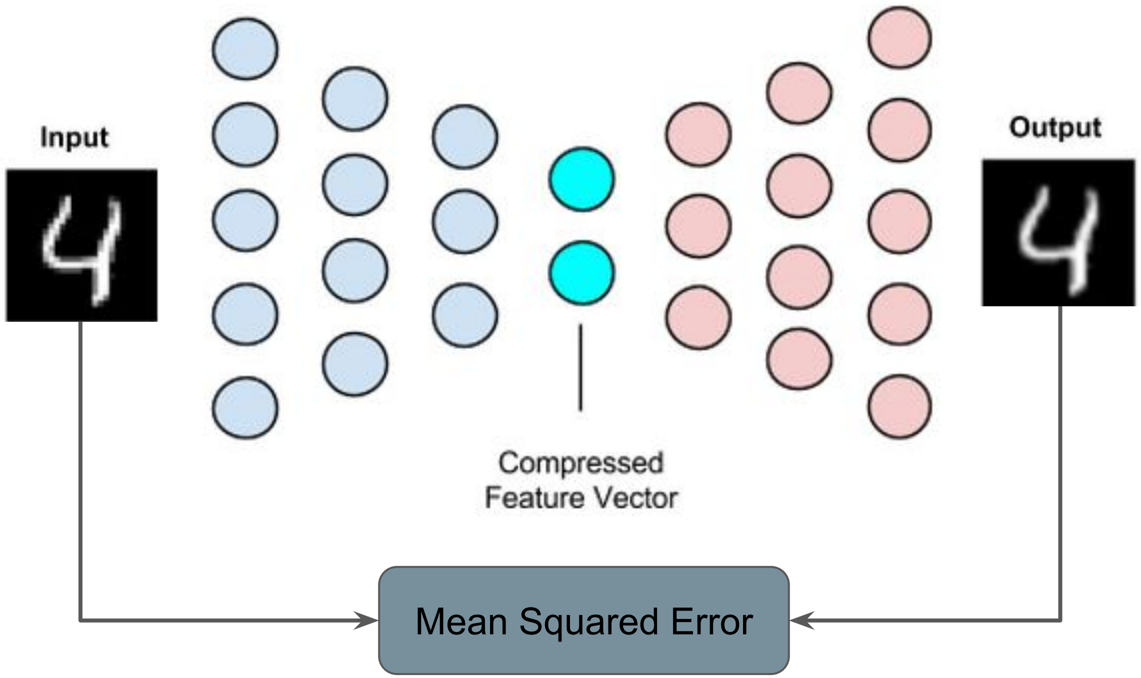
Machine Learning

Lecture 6

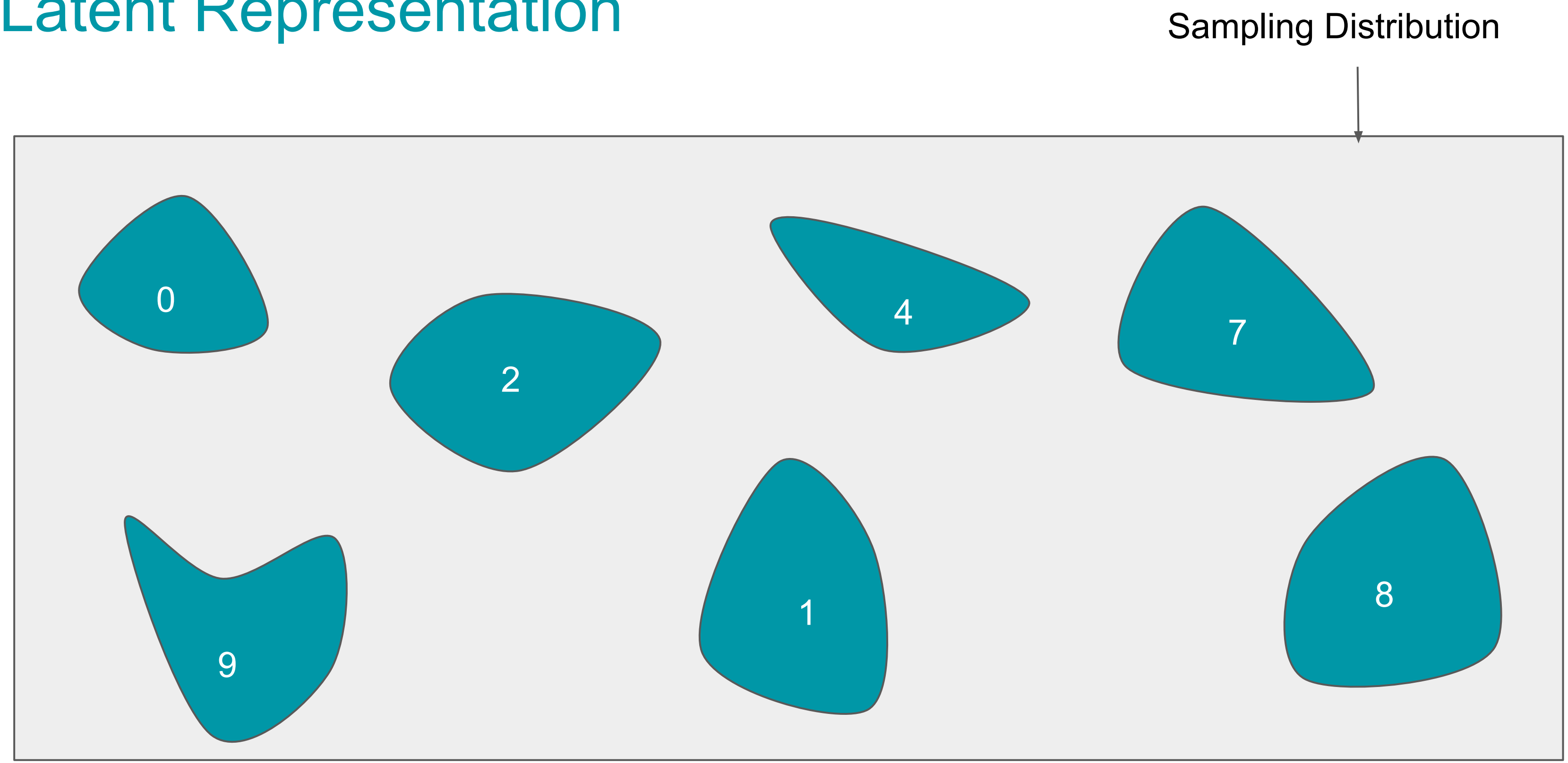
Autoencoders



Deep Autoencoders

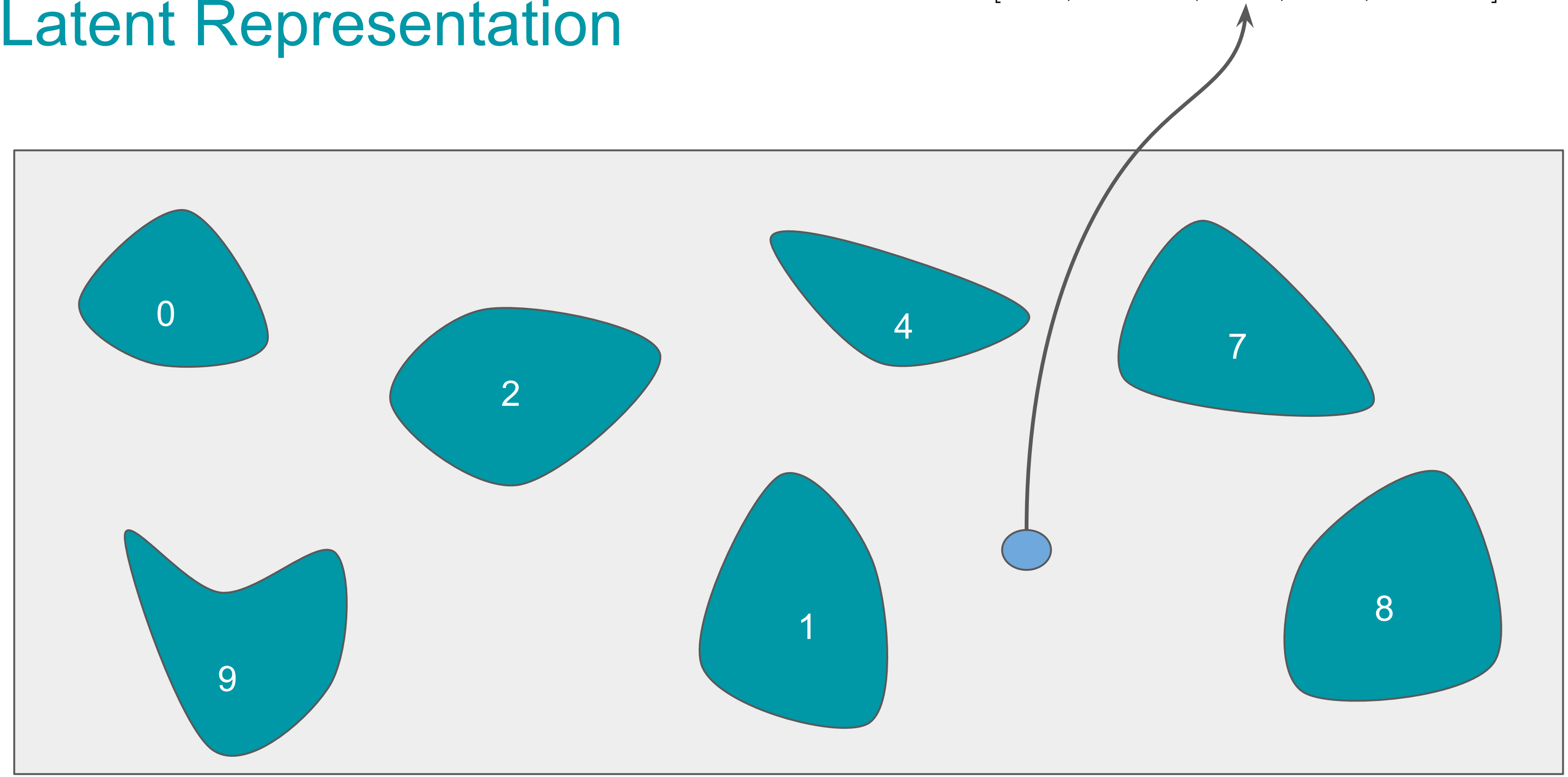


Latent Representation



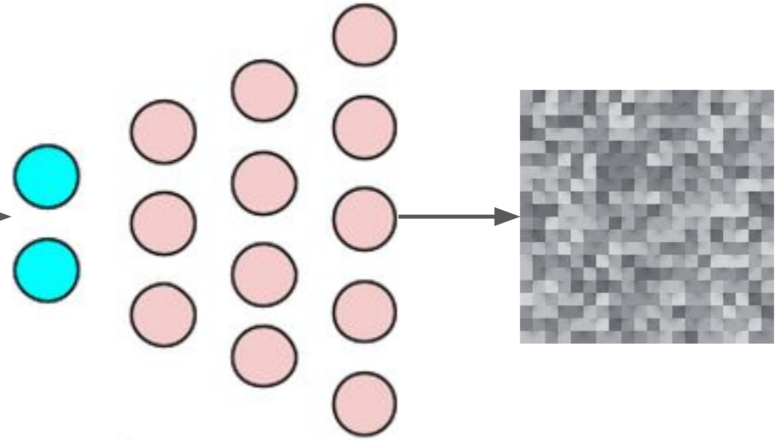
Latent Representation

$$[1.23, \; -5.43, \; 4.67, \; 0.98, \; -4.33]$$

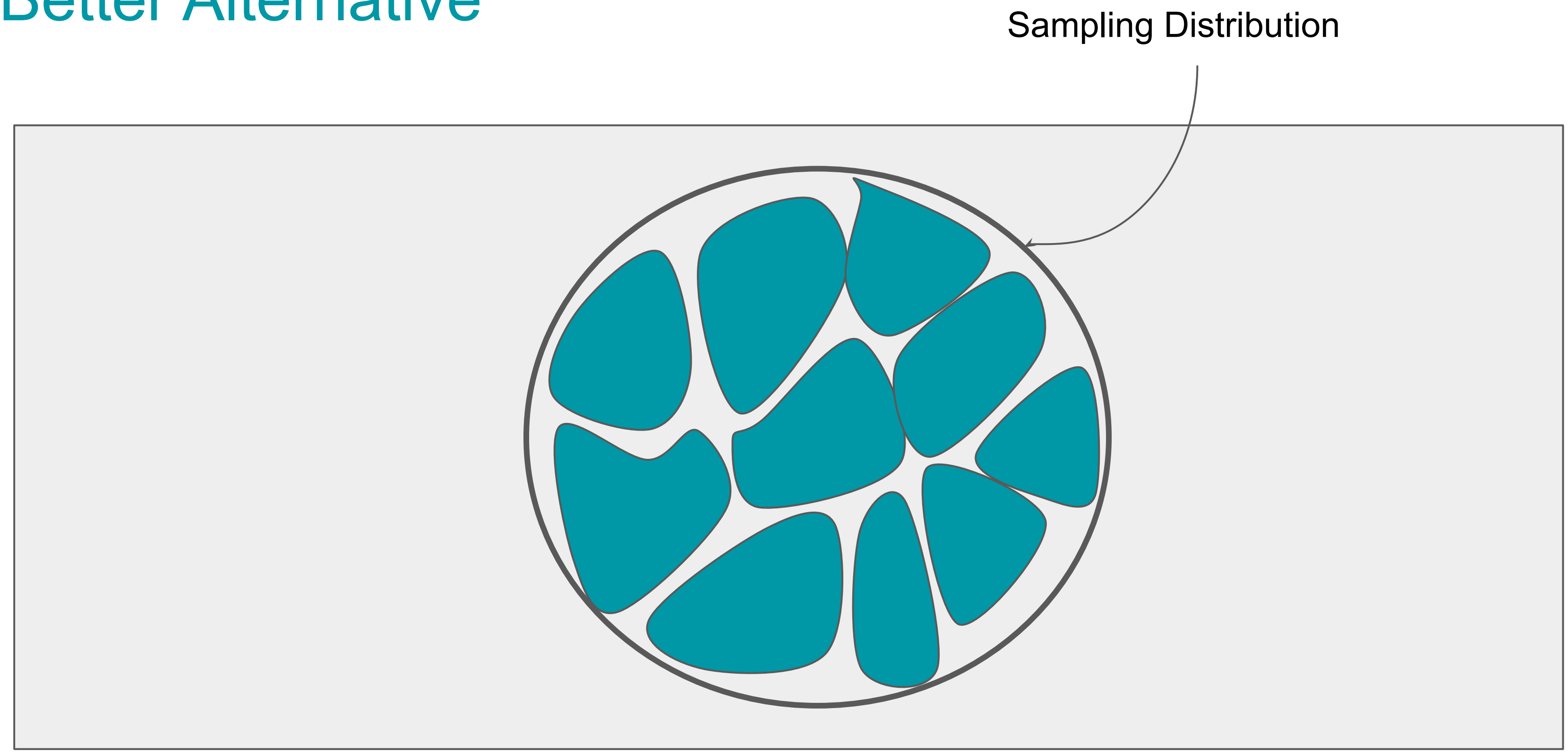


Autoencoders

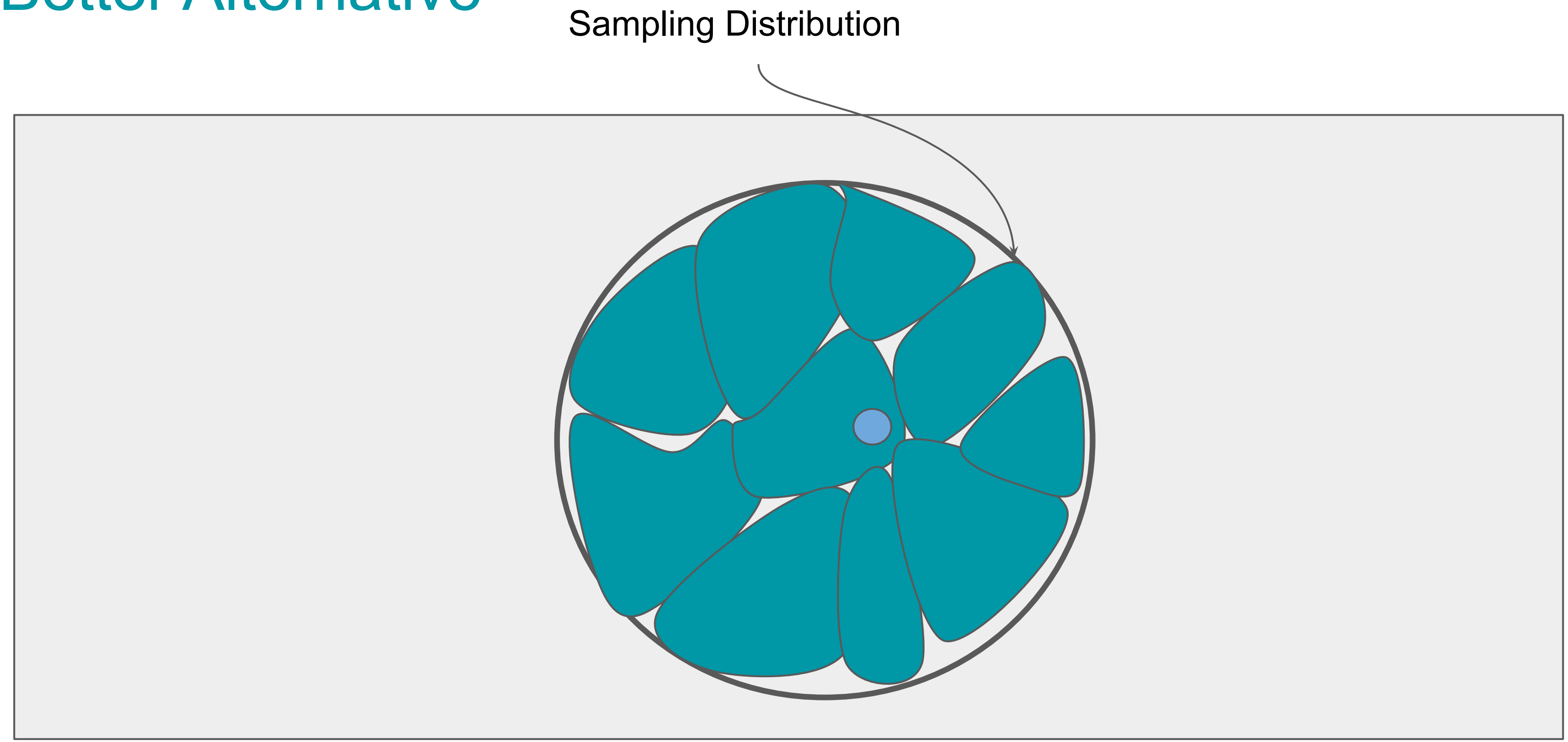
$$[1.23, \ -5.43, 4.67, 0.98, \ -4.33]$$



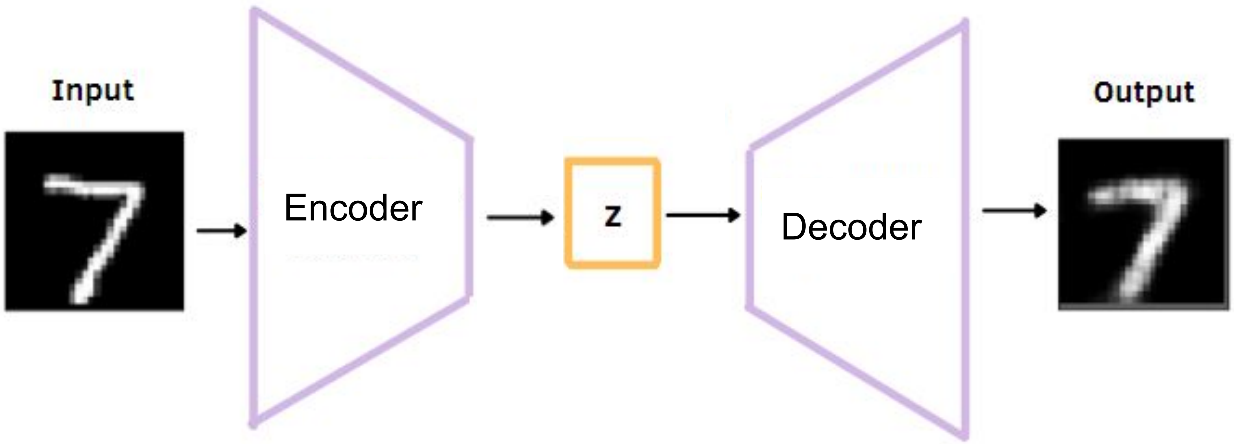
Better Alternative



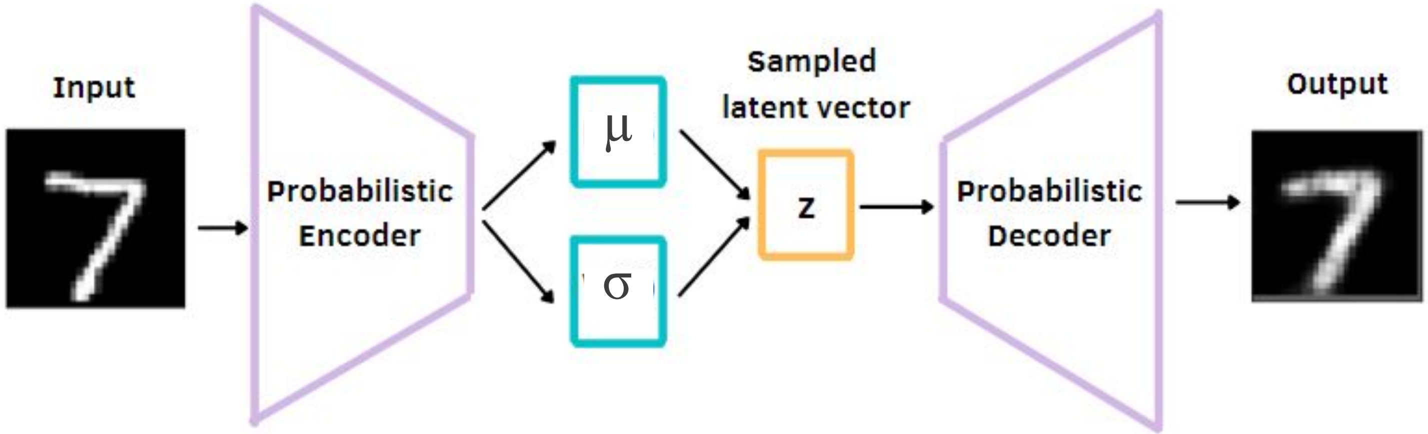
Better Alternative



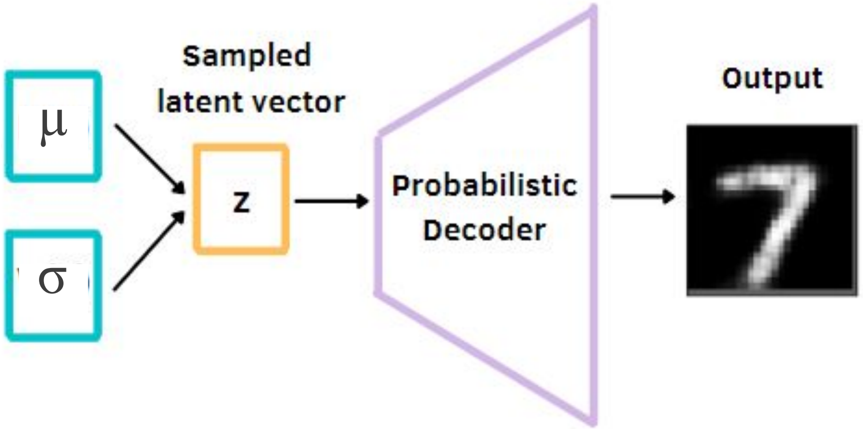
Autoencoders



Variational Autoencoders (VAEs)



Generating with VAEs

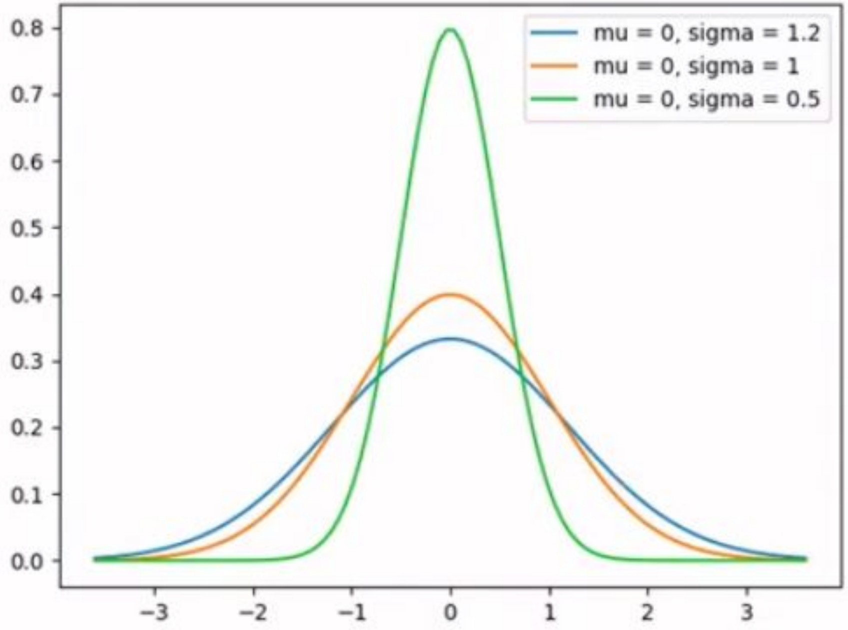


Few Definitions

- Gaussian Distribution
  - $z \sim N(\mu, \sigma)$.
    - μ - mean
    - σ - standard deviation

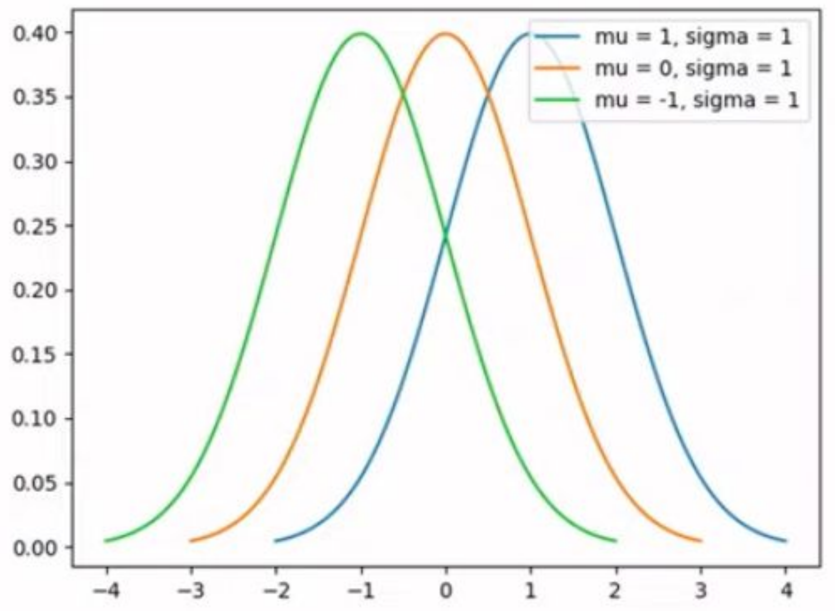
Few Definitions

- Gaussian Distribution
  - $z \sim N(\mu, \sigma)$.
    - μ - mean
    - σ - standard deviation



Few Definitions

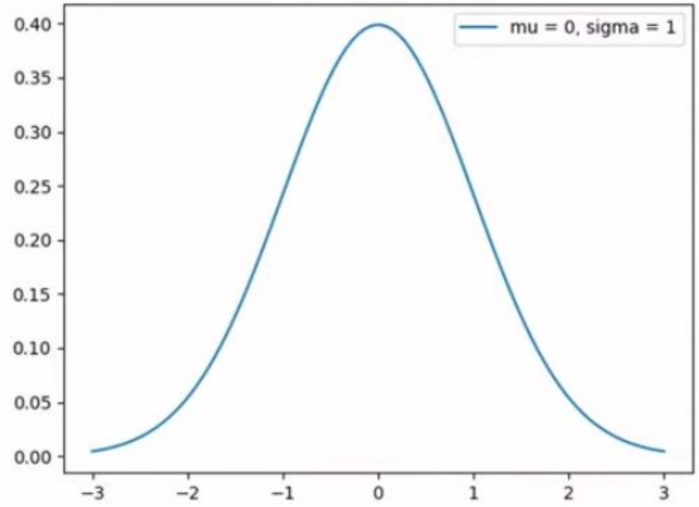
- Gaussian Distribution
  - $z \sim N(\mu, \sigma)$.
    - μ - mean
    - σ - standard deviation



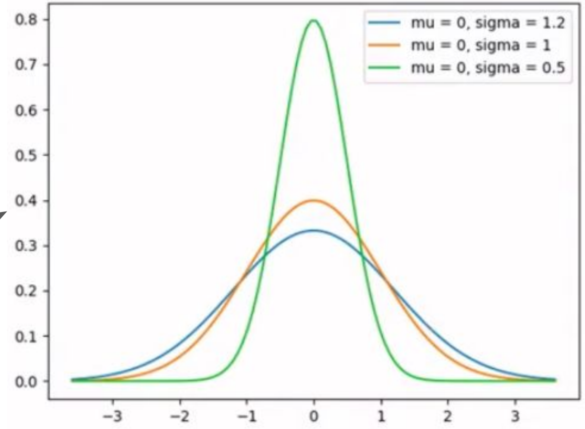
Reparameterization Trick

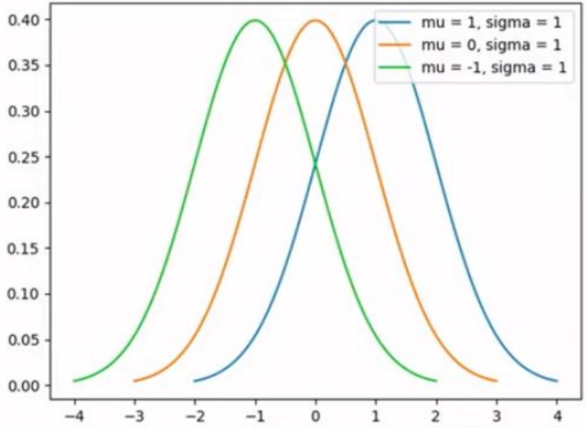
- We want to use gradient descent to learn the model's parameters.
- Given z drawn from $q_\theta(z|x)$, how do we take derivatives of (a function of) z w.r.t. θ?
- We can reparameterize: $z = \mu + \sigma \odot \epsilon$
- $\epsilon \sim N(0, 1)$
- Now we can take derivatives of (functions of) z w.r.t. μ and σ.
- Output of $q_\theta(z|x)$ is vector of μ's and vector of σ's.

Reparameterization Trick

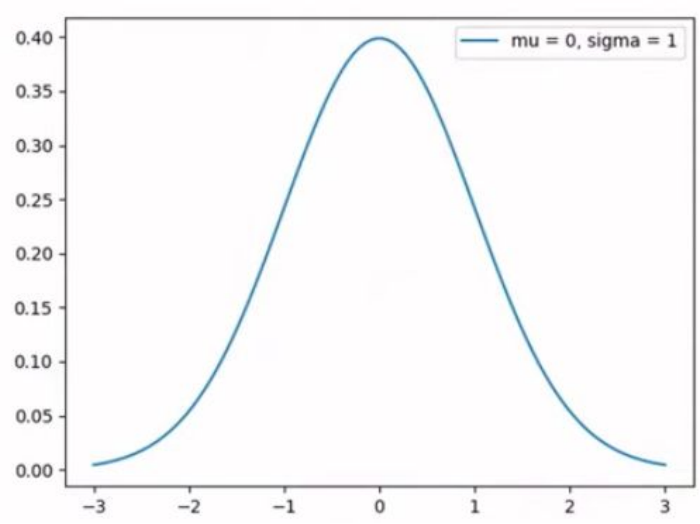


If (μ, σ)
are known





Reparameterization Trick



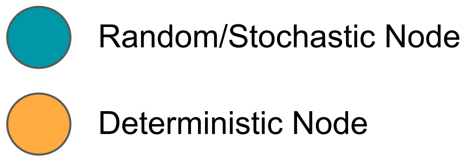
If (μ, σ) are known,

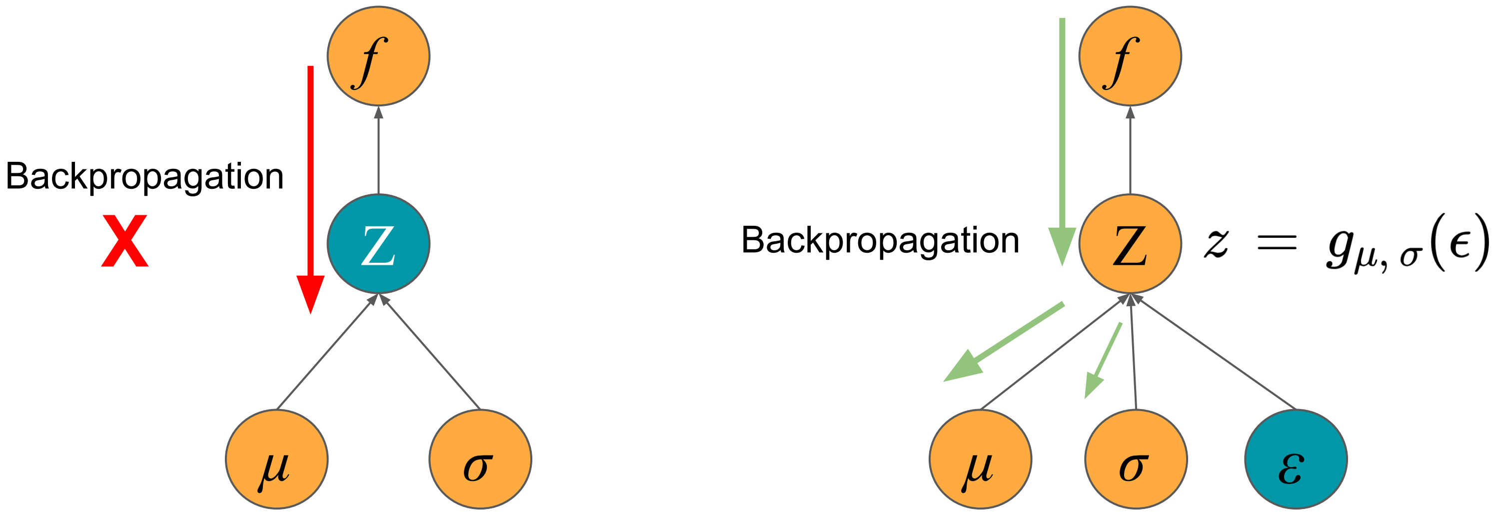
$$g_{\mu,\sigma}(\epsilon) = \mu + \sigma \odot \epsilon$$

$$\epsilon \sim N(0,1)$$

$$z = g_{\mu,\sigma}(\epsilon)$$

Reparameterization Trick





Loss Functions

Reconstruction Loss $\rightarrow$ **Binary Cross Entropy Loss**

Input

Output

$$-\frac{1}{N}\sum_{i=1}^{N} y_i . \log\left(p(y_i)\right) + (1-y_i) . \log\left(1-p(y_i)\right)$$

Latent Loss $\rightarrow$ **KL Divergence Loss**

$$D_{KL}(P||Q) = \int P(x)\log\frac{P(x)}{Q(x)}dx$$