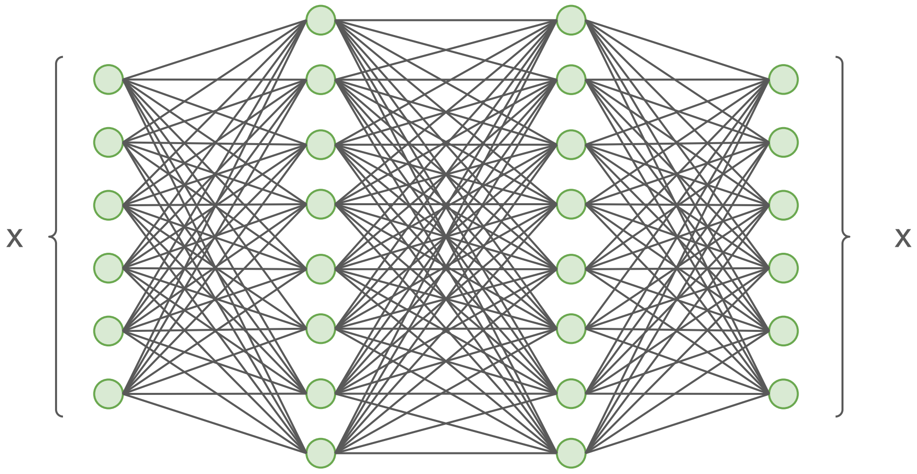
Variational AutoEncoders

Thanos Tagaris

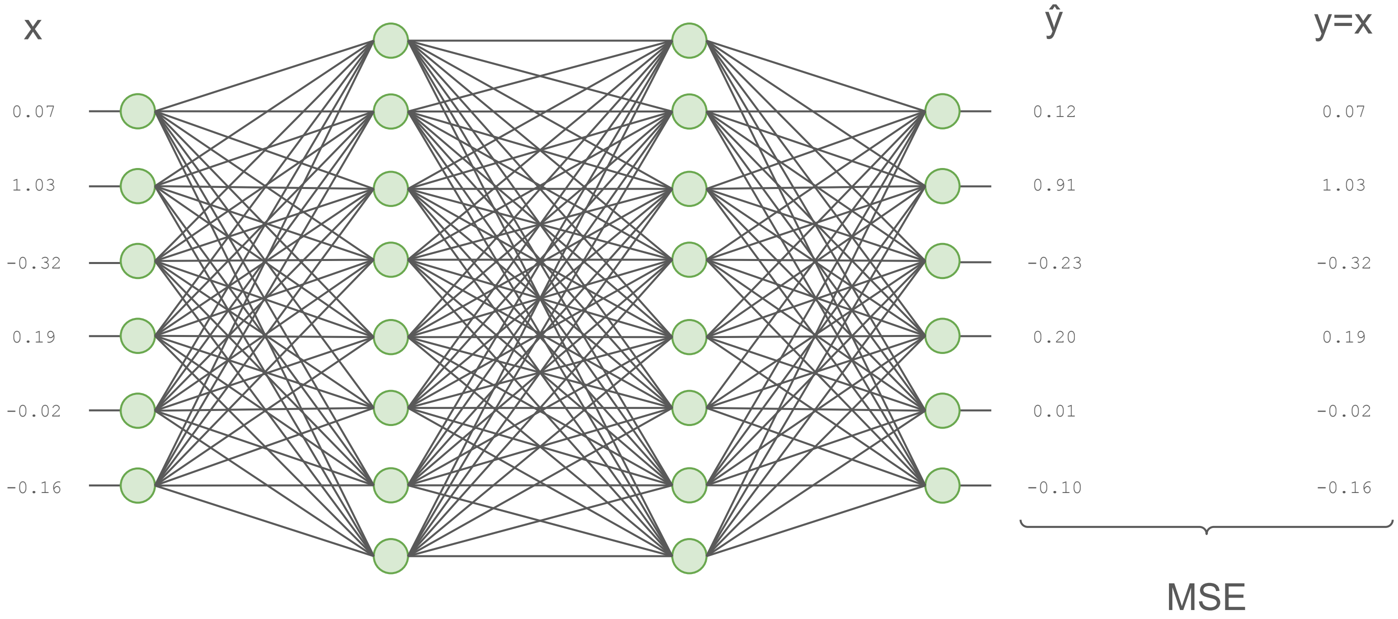
Generative Neural Networks: AutoEncoders

AutoEncoders (AE)

- A Neural Network architecture that has the same shape for its input and output
- Trained in an unsupervised manner
- This leads to a generative training

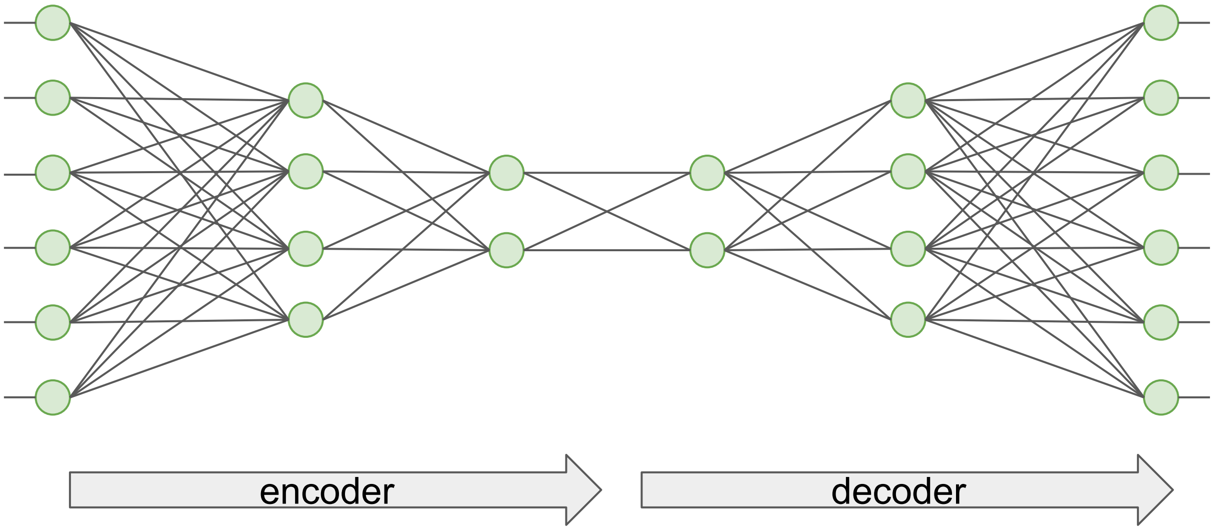


AutoEncoders (AE)



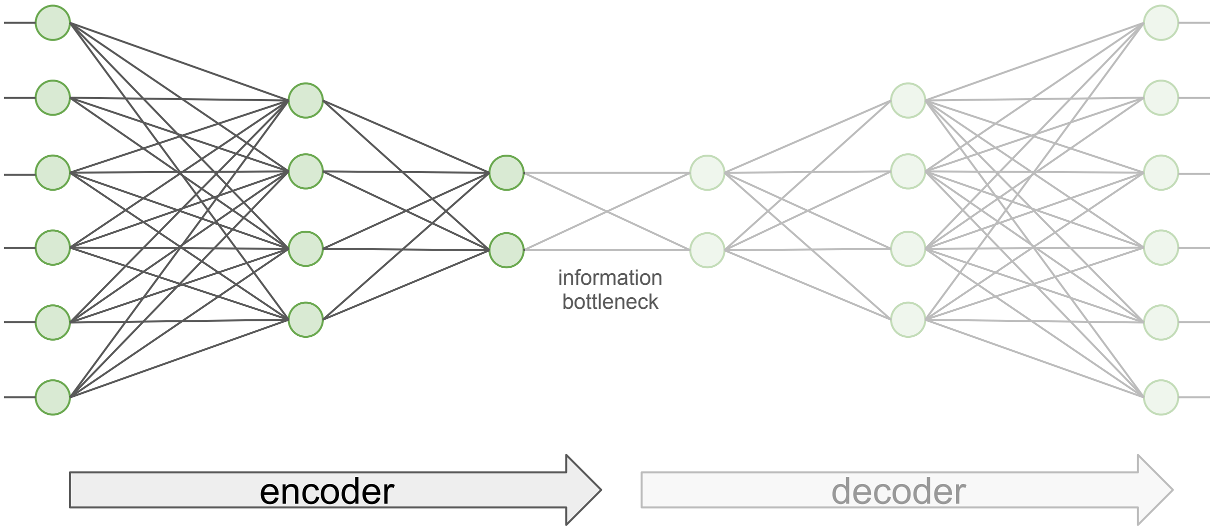
AutoEncoders (AE)

- One thing was inaccurate in the previous depictions
- Autoencoders need to *shrink* the input dimensions to *compress* the input information
- Else they could just learn to copy the information from input directly to output



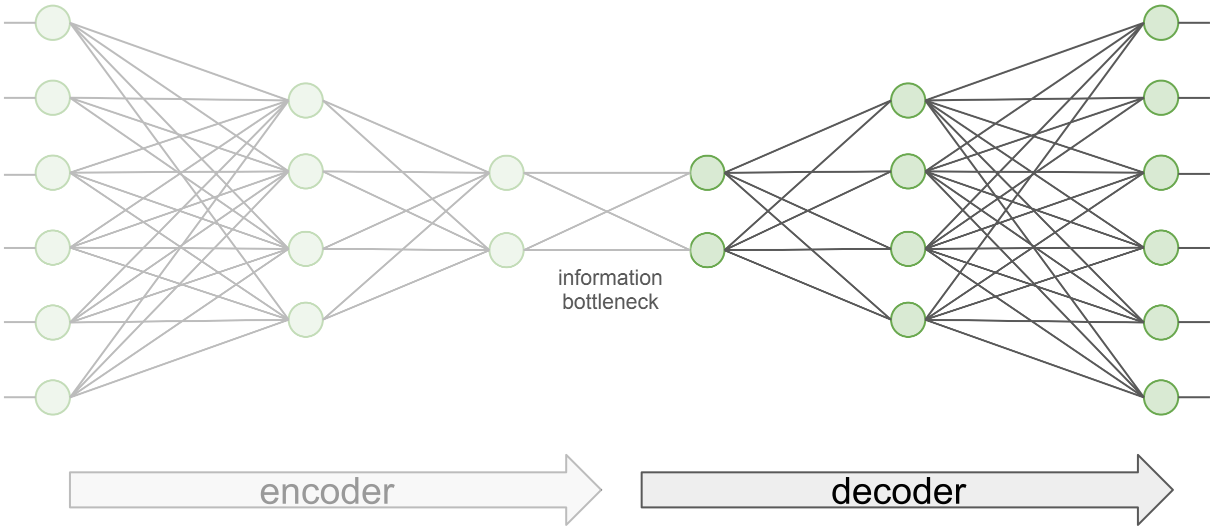
Encoding

- The task of the autoencoder is to essentially *reconstruct* the original input
- The encoder compresses the most **useful information** on how to do this
- Due to the information shrinkage, it can’t capture all the details.



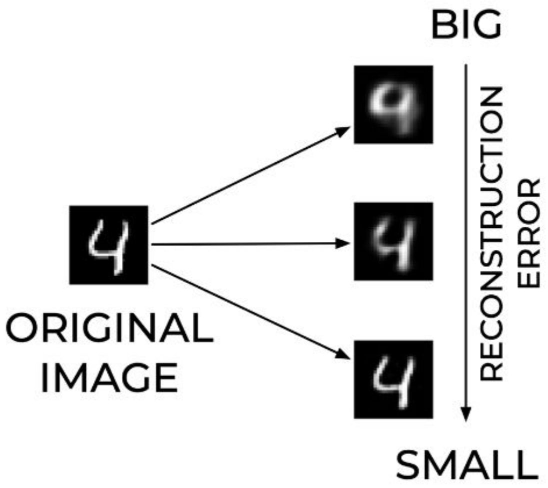
Decoding

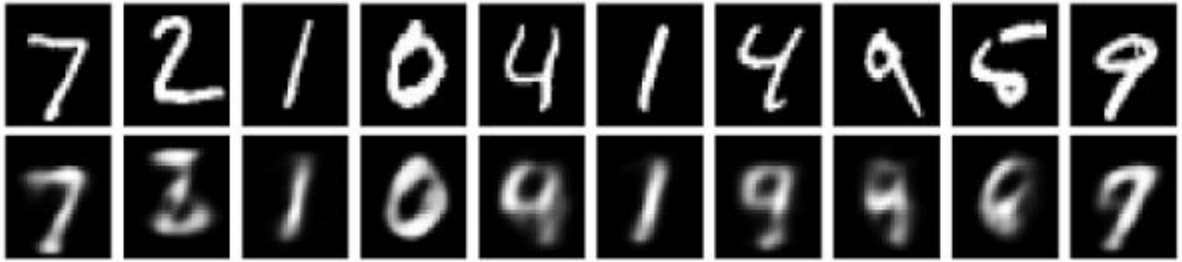
- The decoder takes the compressed vector as its input and needs to fill in all the missing details.
- To do this it needs to understand some fundamental properties about how the input data's underlying distribution.
- This is what makes it generative.



Input Reconstruction

- An autoencoder can work on any modality (images, audio, text, etc.)
- Again, to be able to reconstruct its input, it needs to learn the distribution of its inputs
- … it needs to be generative

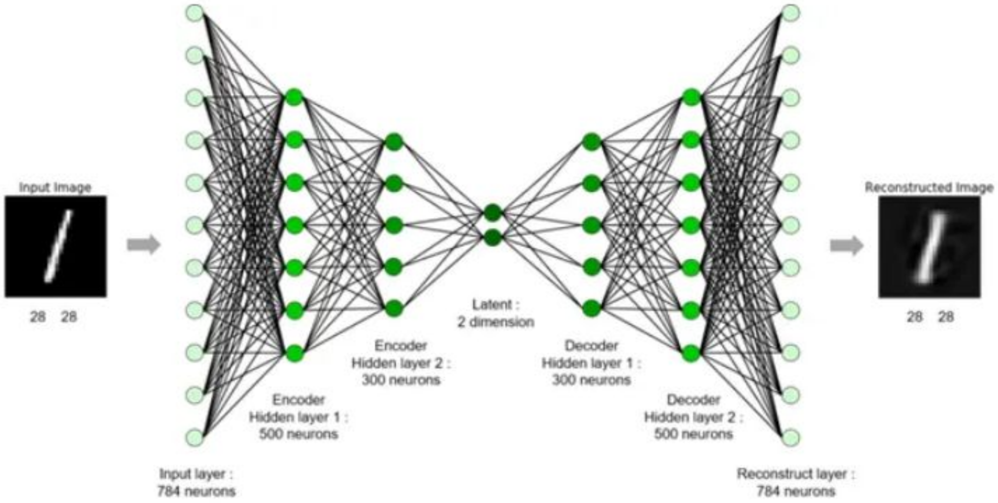


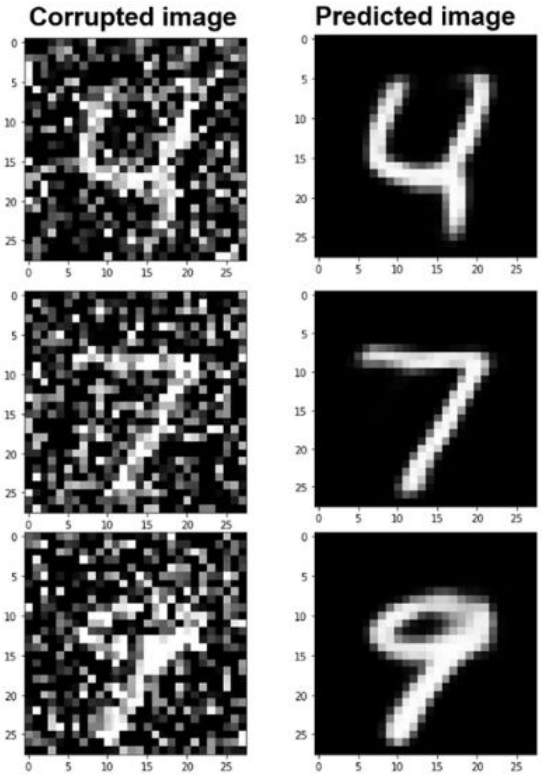


Demo

Build a convolutional autoencoder on the MNIST dataset

Denoising Autoencoders

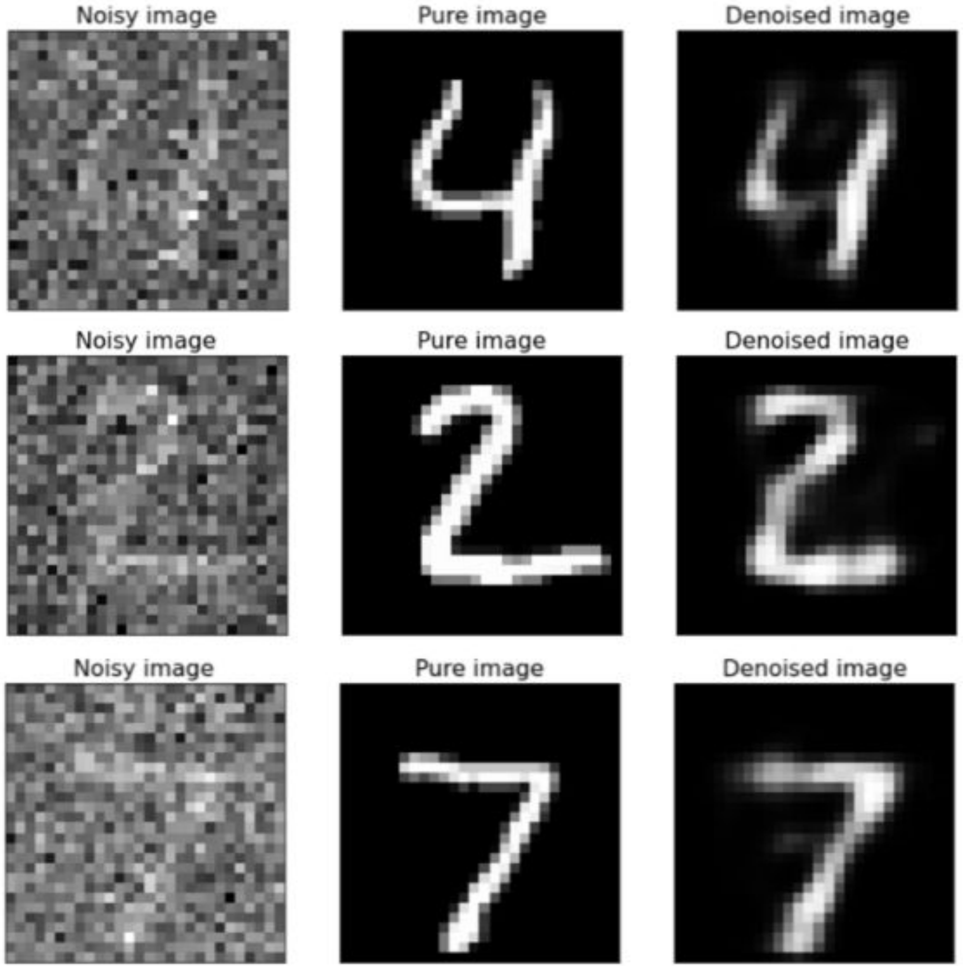




Denoising Autoencoders

even harder examples…

This process forces the autoencoder to look beyond the input noise and learn the true underlying distribution

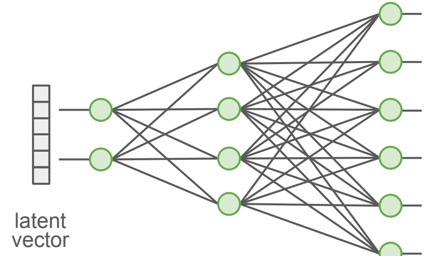


Demo

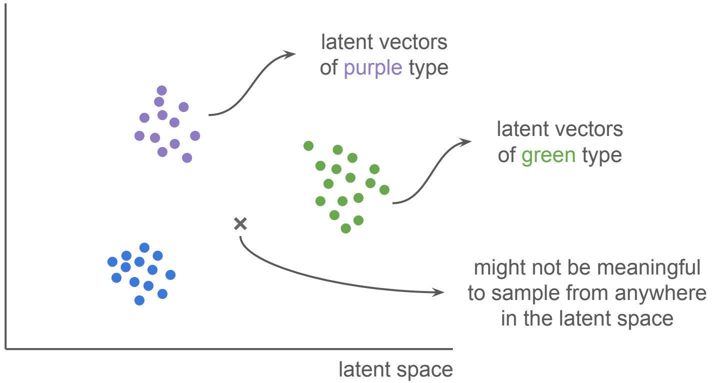
Try to train the previously created AutoEncoder for denoising

Using Autoencoders for Generative tasks

- So, we know that the decoding part of the AE is generative.
- What do we need to do to make it generate something?
  - ➢ supply it with a latent vector

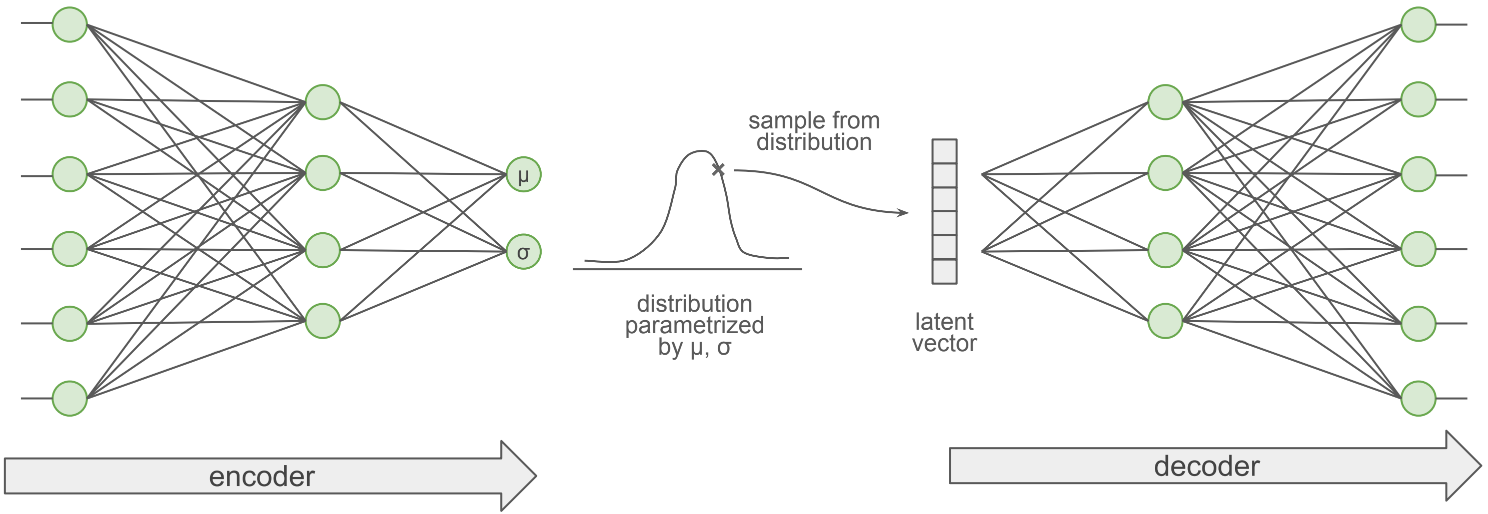


- How do we know what values to choose in the latent vector?
  - ➢ we can't
  - ➢ can we randomly choose a vector?



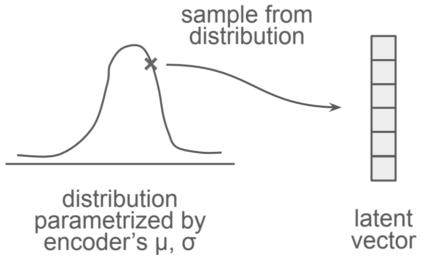
Variational AutoEncoders (VAE)

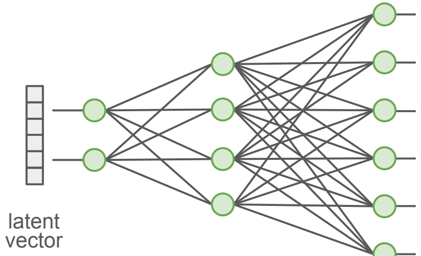
- Idea: instead of learning the latent vectors deterministically, learn the distribution from which they are sampled.
- Encoder will learn the parameters of this distribution (e.g. μ, σ in normal distribution)
- Decoder will reconstruct output from a latent vector, sampled from this distribution

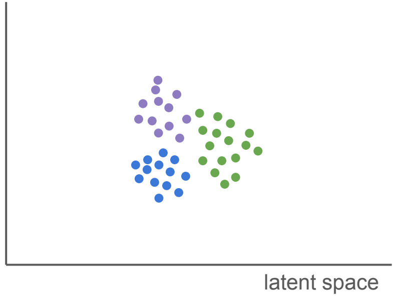


Using VAEs for Generative tasks

- So, how does a VAE help us with our generative tasks?
  - ➢ In deterministic AEs we didn't know how to select the latent vector
  - ➢ In VAEs, we know the distribution from which latent vectors can be sampled
- Generative process:
  - ➢ Step 1: sample a latent vector from the distribution
  - ➢ Step 2: supply latent vector to the decoder
- VAE's training objective leads them to have a more continuous latent space

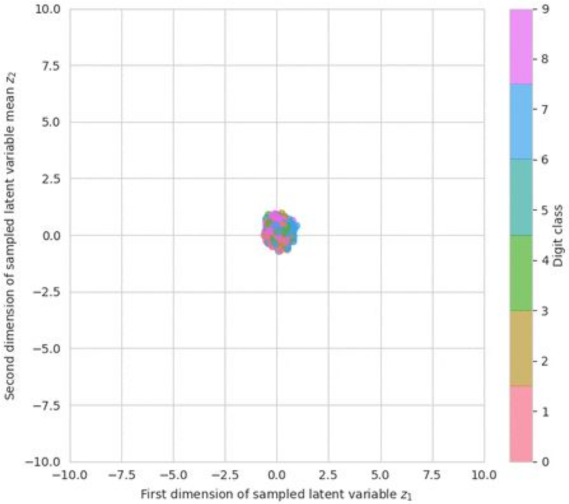


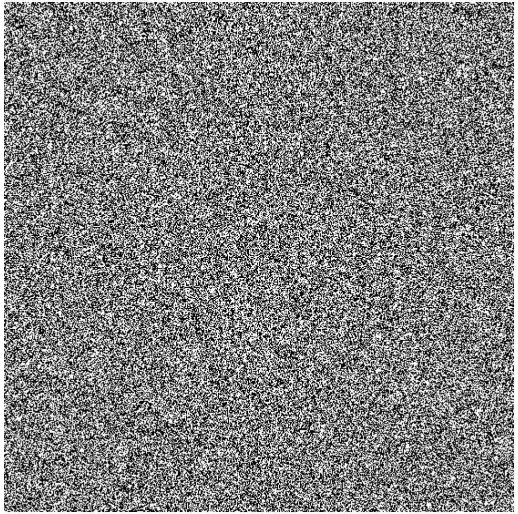




VAE: training details

- Trained with ELBO loss function. Two terms:
    - Reconstruction loss
    - Regularization term to constrain latent distribution to “follow” a prior (e.g. gaussian)





Evidence Lower Bound

$$\begin{aligned}
\log p_\theta(x^{(i)}) &\geq \mathcal{L}(\theta, \phi; x^{(i)}) \\
&= E_{q_\phi(z|x^{(i)})}[-\log q_\phi(z|x) + \log p_\theta(x|z)] \\
&= \underbrace{-D_{KL}(q_\phi(z|x^{(i)}) || p_\theta(z))}_{\text{KL diversion to keep distribution close to a prior distribution}} + \underbrace{E_{q_\phi(z|x^{(i)})}[\log p_\theta(x|z)]}_{\text{likelihood}}
\end{aligned}$$