

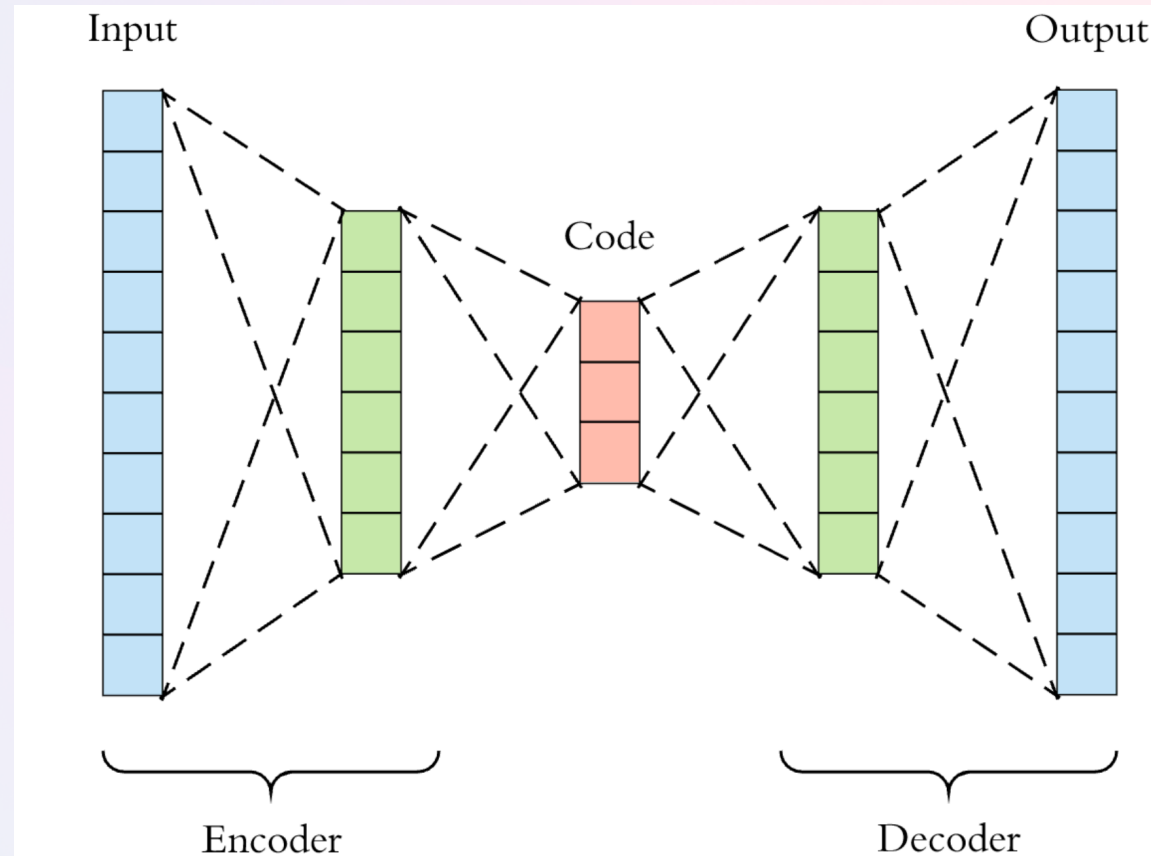
# Autoencoders

David I. Inouye

# Outline

- Autoencoder basics
- Autoencoder constraints and formulations
  - Undercomplete autoencoder
  - Sparse autoencoder
  - Denoising autoencoders
- Probabilistic autoencoders
- Variational Autoencoder (VAE)
  - Definition and objective derivation
  - Evidence Lower Bound (ELBO)
  - Reparameterization trick

# Autoencoders Map an Input to a Latent Code (**Encoder**) and Map This Latent Code Back to the Input (**Decoder**)



<https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368>

# The Optimization Problem Is to Fit the Encoder and Decoder Simultaneously to Reconstruct Output

- More formally, the autoencoder objective is:

$$\min_{f,g} \mathbb{E}[L(x, \tilde{x})] \implies \min_{f,g} \mathbb{E}[L(x, g(f(x)))]$$

- One example is using Mean Squared Error loss:

$$\min_{f,g} \mathbb{E}[\|x - g(f(x))\|_2^2]$$

# If There Are No Constraints on the Encoder and Decoder Than the Identity Function Works Perfectly...

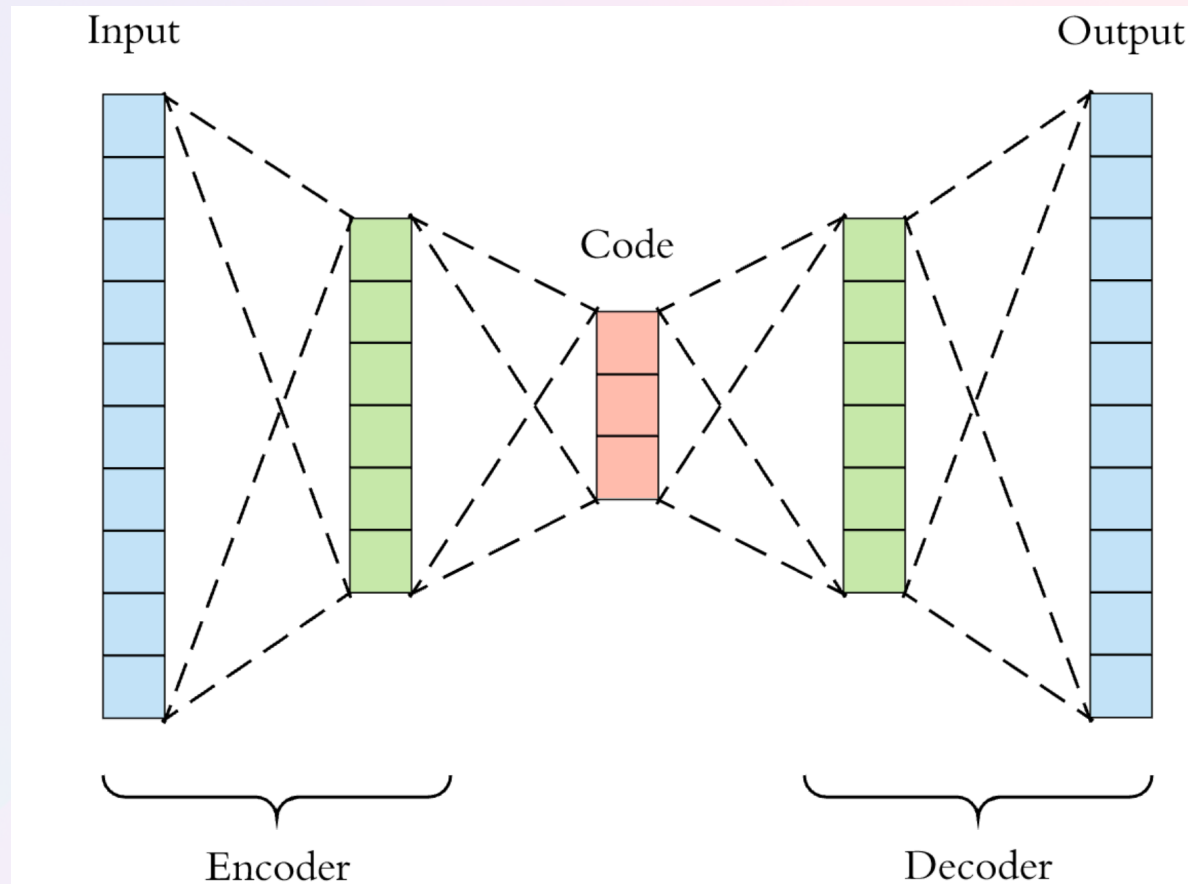
- Suppose  $f(x) = x$  and  $g(x) = x$ .
- Then we know that:

$$\min_{f,g} \mathbb{E}[\|x - g(f(x))\|_2^2] = \min_{f,g} \mathbb{E}[\|x - x\|_2^2] = 0$$

- And since all terms are positive, this is the global minimum.
- Trivial/useless... What can we do?

# Adding Constraints to F, G or Z Can Often Produce Interesting Properties of Z

- **Undercomplete autoencoders** assume that the latent space has lower dimension, i.e.,  $k < d$ .



# The **Undercomplete** and **Linear** Autoencoder Is Closely Related to PCA

- **Formally**

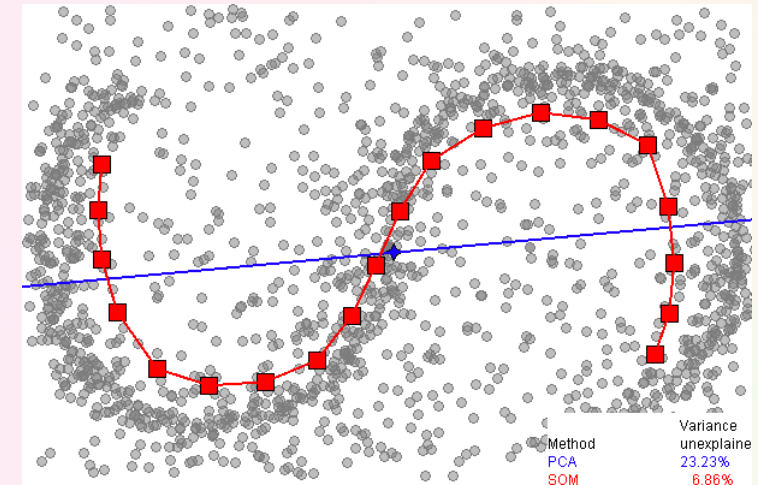
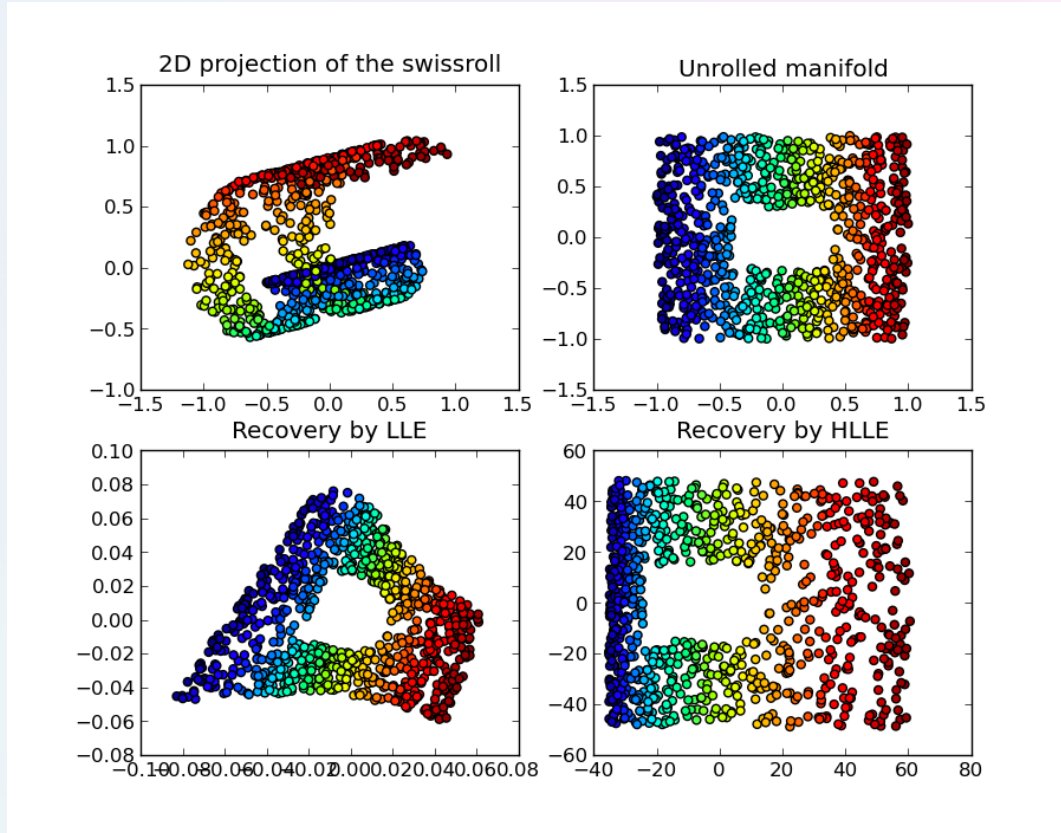
- Let  $z = f(x) = Ax + b, z \in \mathbb{R}^k$

- Let  $\tilde{x} = g(z) = Bz + c$

- Let  $L(x, \tilde{x}) = \mathbb{E}[\|x - \tilde{x}\|_2^2]$

- One solution can be derived from PCA though other (closely-related) solutions exist.
- Autoencoders are “non-linear” PCA.

# Why Might We Want a Non-Linear Autoencoder? Non-Linear Dimensionality Reduction



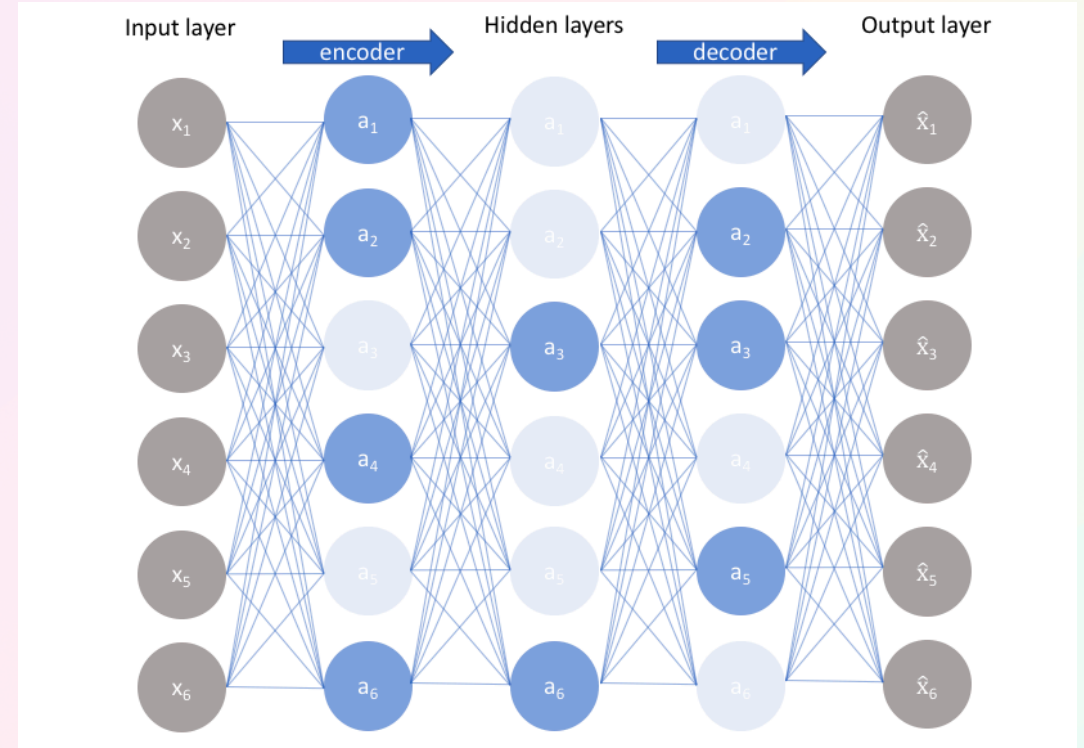
[https://en.wikipedia.org/wiki/Nonlinear\\_dimensionality\\_reduction](https://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction)

# Sparse Autoencoders Add a Penalty That the Latent Space Is Sparse

- Add a regularization term to latent variables:

$$\min_{f,g} \mathbb{E}[L(x, g(f(x))) + \lambda \|f(x)\|_1]$$

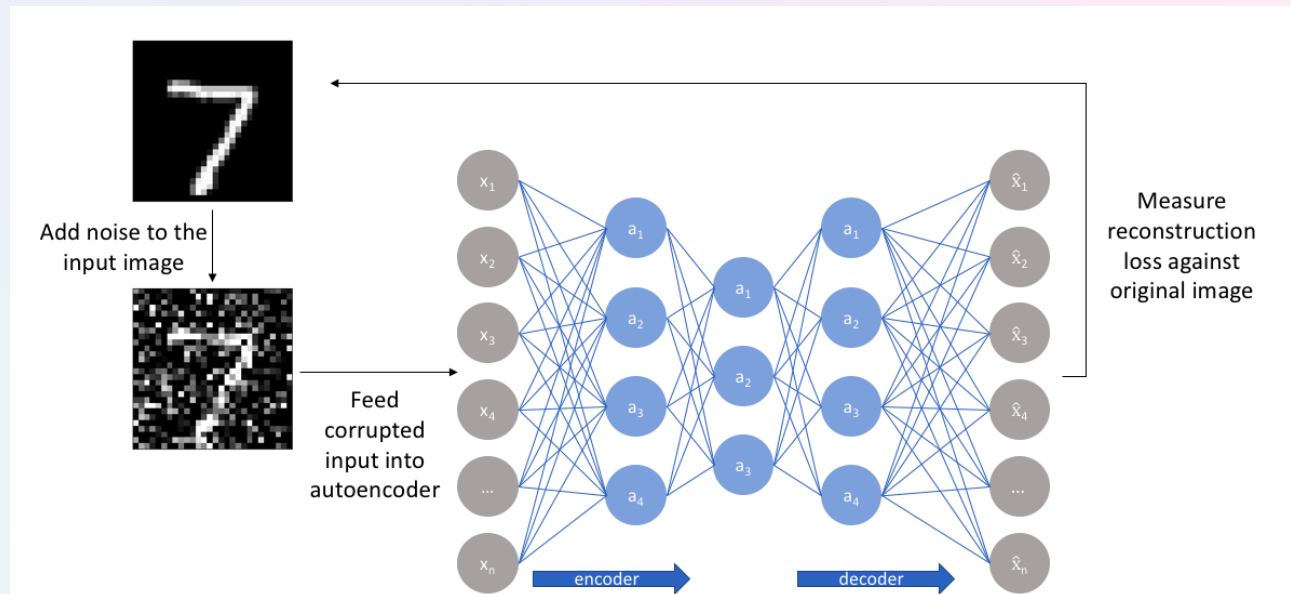
- This creates **data-dependent** sparsity.



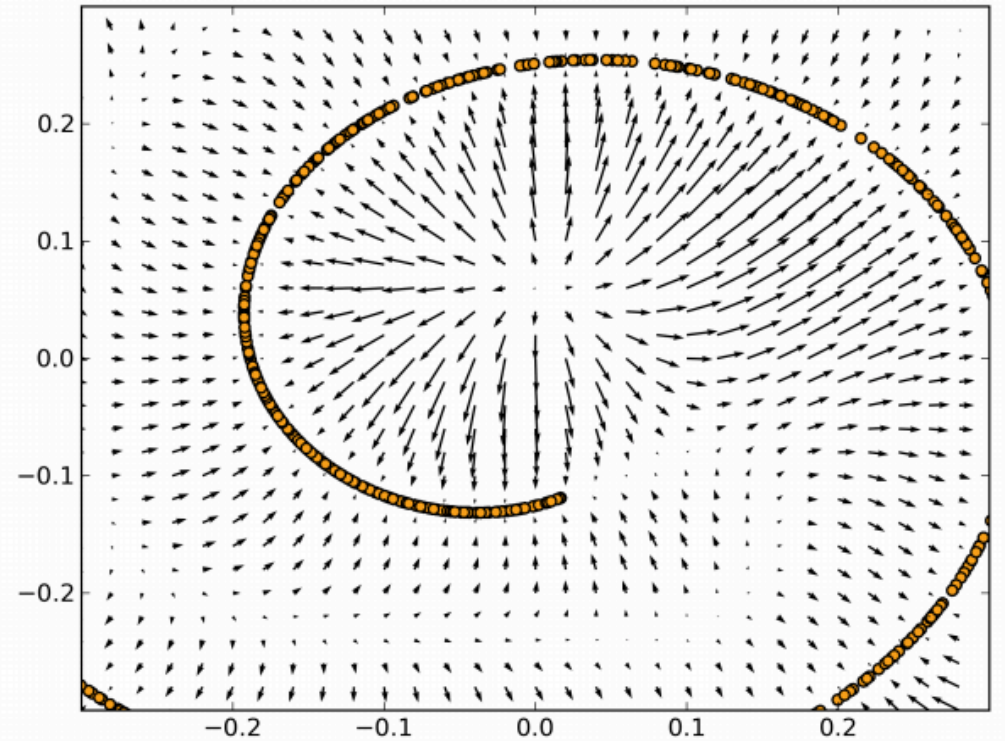
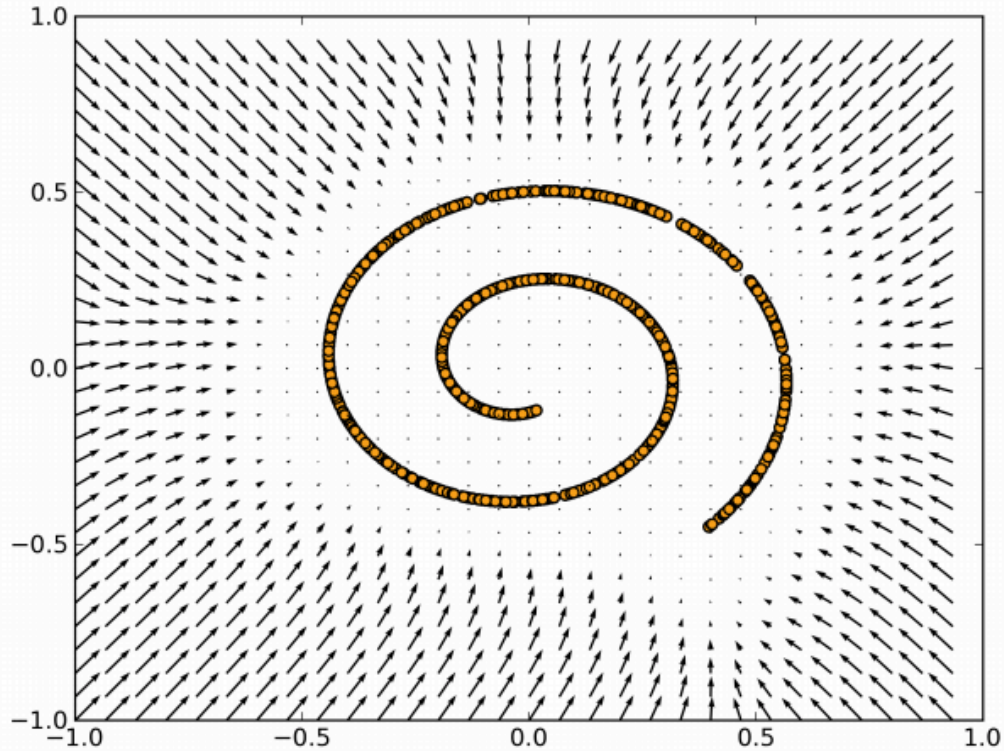
# Denoising Autoencoders Force Functions to Learn to Remove Noise Rather Than Copy the Input

- Add noise to the input so that copying input is not possible:

$$\min_{f,g} \mathbb{E}_{x,\epsilon} [L(x, g(f(x + \epsilon)))], \quad \text{where } \epsilon \sim \mathcal{N}(\mu, \sigma I)$$

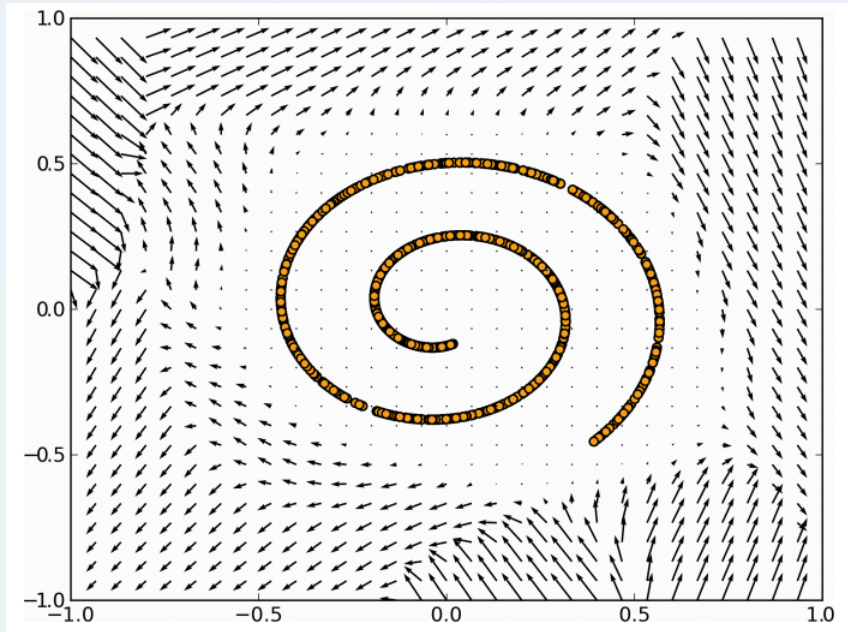


# Denoising Autoencoders Can Be Shown to Learn the Structure of the Distribution



# Denoising Autoencoders Can Be Shown to Learn the Structure of the Distribution

But the vector field far from the training data is not trained well. (Diffusion models fix this problem.)



<https://arxiv.org/pdf/1211.4246.pdf>

# Autoencoders Can Also Use **Non-Deterministic** or **Probabilistic Mappings**

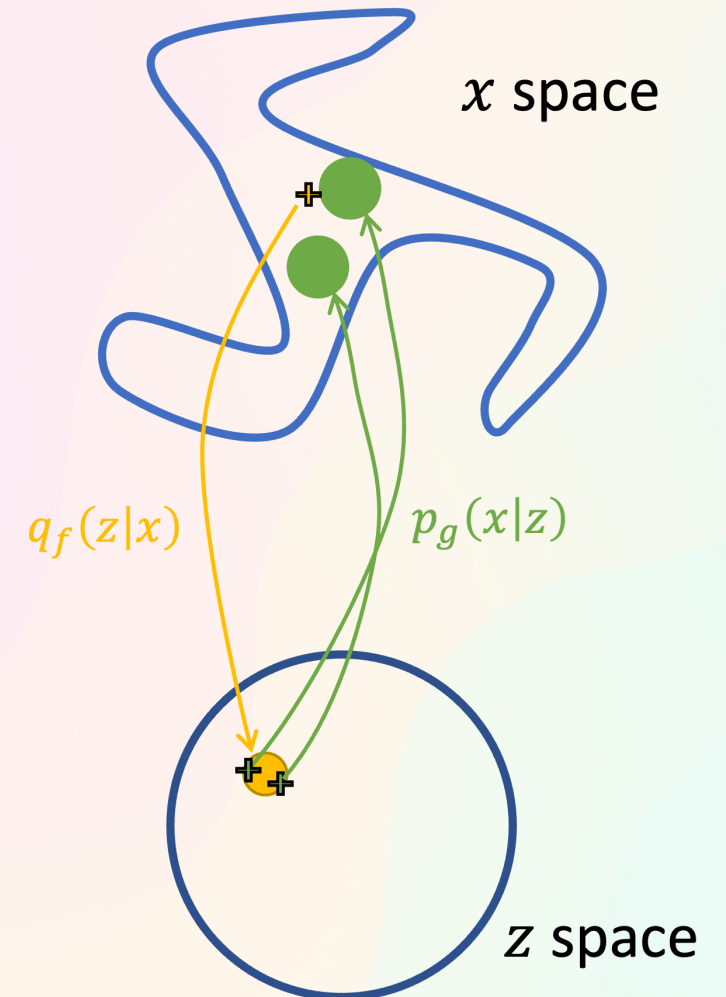
- The outputs are distributions instead of a points.
- Encoder/decoder output the parameters of distribution.
- **Probabilistic mappings**

- Replace encoder  $f(x)$  with

$$q_f(z|x) = \mathcal{N}(z; \mu = f(x), \Sigma = I)$$

- Replace decoder  $g(z)$  with

$$p_g(x|z) = \mathcal{N}(x; \mu = g(z), \Sigma = I)$$



# Vanilla Probabilistic Autoencoder Could Minimize Expected Negative Log Likelihood of Training Data

$$\min_{f,g} E_{p_{\text{data}}(x)} [E_{q_f(z|x)} [-\log p_g(x|z)]]$$

## Key Facts for Derivation:

1.  $p_g(x_i|z_i^l) = N(x; \mu_i^l = g(z_i^l), \Sigma = I)$

2.  $q_f(z_i^l|x_i) = N(z; \mu_i = f(x_i), \Sigma = I)$

- **Sampling/Reparametrization trick:** If  $\epsilon \sim N(0, I)$  and  $z_l = \mu + \epsilon$ , then  $z_l \sim N(\mu, I)$ .

# Vanilla Probabilistic Autoencoder Could Minimize Expected Negative Log Likelihood of Training Data

$$\begin{aligned} & \hat{E}_{p_{\text{data}}(x)}[\hat{E}_{q_f(z|x)}[-\log p_g(x|z)]] \quad (\text{Empirical expectation}) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{l=1}^m -\log p_g(x_i|z_i^l) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{l=1}^m -\log \exp\left(-\frac{1}{2}\|x_i - \mu_i^l\|_2^2 - \frac{d}{2}\log 2\pi\right) \quad (\text{Definition of } p_g) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{l=1}^m \frac{1}{2}\|x_i - \mu_i^l\|_2^2 + c \quad (c \text{ is constant}) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{l=1}^m \frac{1}{2}\|x_i - g(z_i^l)\|_2^2 + c(\mu_i^l \text{ in terms of } z_i^l) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{l=1}^m \frac{1}{2}\|x_i - g(f(x_i) + \epsilon_i^l)\|_2^2 + c \quad (\text{Definition and Reparametrization Fact (1,3)}) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{1}{2}\|x_i - g(f(x_i) + \epsilon_i)\|_2^2 + c \quad (\text{let } m = 1, \text{ where } \epsilon_i \sim N(0, I)) \end{aligned}$$

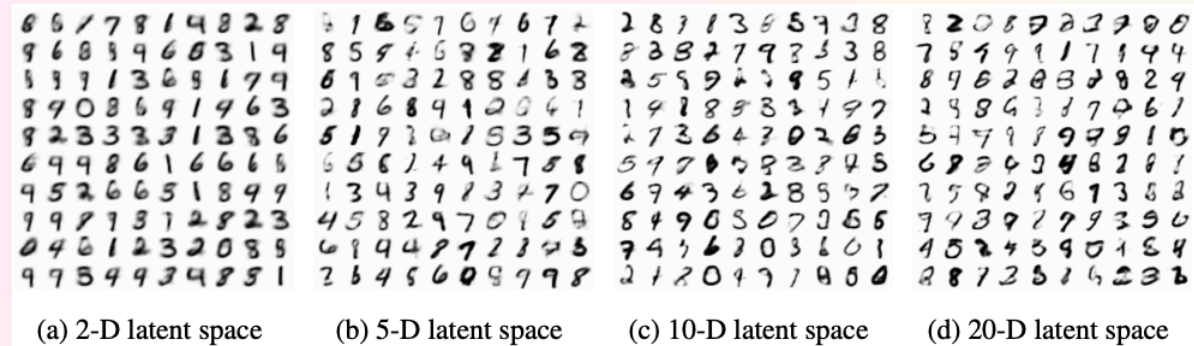
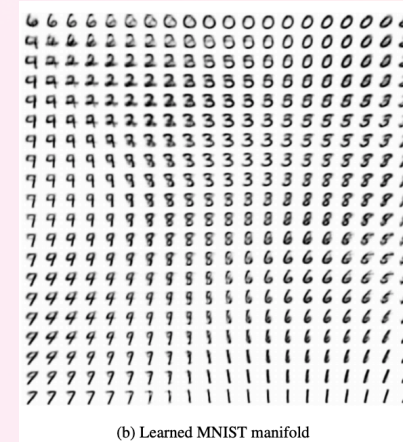
# Comparison Between Autoencoders

- MSE autoencoder (AE)
  - $\min_{f,g} \frac{1}{n} \sum_i \|x_i - g(f(x_i))\|_2^2$
- Sparse autoencoder
  - $\min_{f,g} \frac{1}{n} \sum_i \|x_i - g(f(x_i))\|_2^2 + \lambda \|f(x_i)\|_1$
- Gaussian denoising autoencoder (DAE)
  - $\min_{f,g} \frac{1}{n} \sum_i \|x_i - g(f(x_i + \epsilon_i))\|_2^2, \quad \epsilon_i \sim \mathcal{N}(\mu, \sigma I)$
- Vanilla Gaussian probabilistic autoencoder
  - $\min_{f,g} \frac{1}{n} \sum_i \|x_i - g(f(x_i) + \epsilon_i)\|_2^2, \quad \epsilon_i \sim \mathcal{N}(0, I)$
- Regularized Gaussian probabilistic autoencoder
  - $\min_{f,g} \frac{1}{n} \sum_i \|x_i - g(f(x_i) + \epsilon_i)\|_2^2 + \lambda \|f(x_i)\|_2^2, \quad \epsilon_i \sim \mathcal{N}(0, I)$
- **This is a special case of Gaussian variational autoencoders (VAE) called Constant Variance VAEs.**

Ghosh, P., Sajjadi, M. S., Vergari, A., Black, M., & Schölkopf, B. (2019). From variational to deterministic autoencoders. arXiv preprint arXiv:1903.12436.

# Variational Autoencoders (VAE) Are One of the Most Common Probabilistic Autoencoders

- Method produces both:
  - Probabilistic encoder/decoder for dimensionality reduction/compression.
  - **Generative model** for the data (AEs don't provide this).
- **Generative model** can produce fake data.



Figures and reference from Kingma, D. P. and M. Welling (2014). "Auto-Encoding Variational Bayes". *International Conference on Learning Representations*.

# VAEs Have Inference and Generative Networks With an Assumed Prior Distribution on $Z$

- **Generative model**

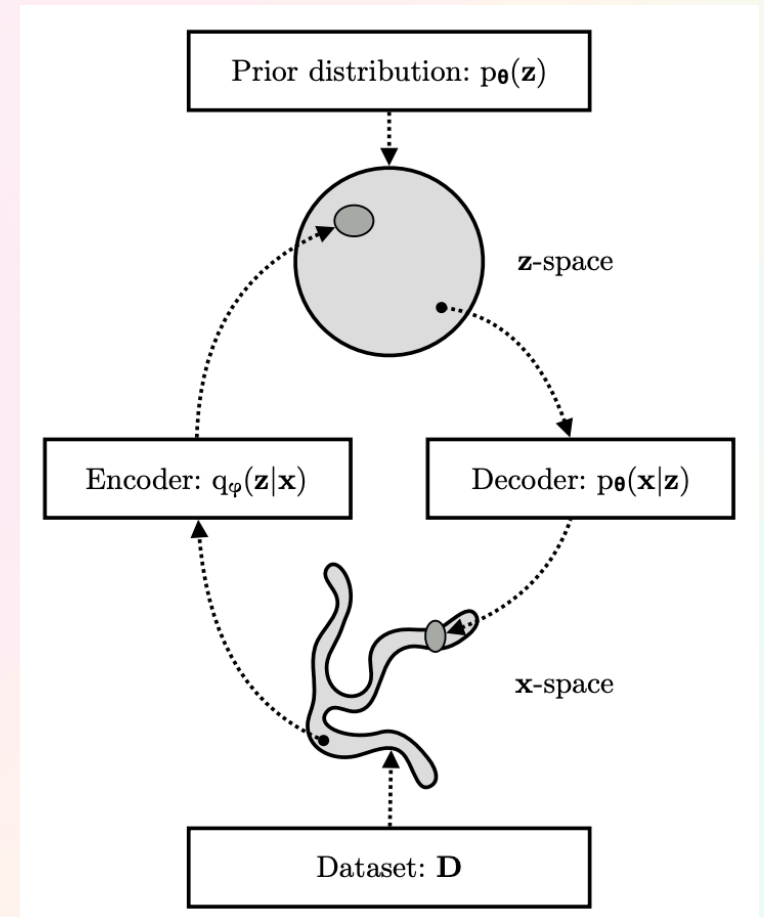
- $z \sim p_g(z) = \mathcal{N}(0, I)$
- $x \sim p_g(x|z)$

- **MLE is intractable**

- $\log p(x; g) = \log \int_z p_g(z) p_g(x|z) dz$

- **Add encoder/inference model to help**

- $x \sim p_{\text{data}}(x)$
- $z \sim q_f(z|x)$



Kingma, D. P., & Welling, M. (2019). An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4), 307-392.

# Derivation of VAE Objective (Known As the Evidence Lower Bound or ELBO)

$$\begin{aligned}\log p_g(x) &= \mathbb{E}_{q_f}[\log p_g(x)] \\ &= \mathbb{E}_{q_f} \left[ \log \frac{p_g(x)p_g(z|x)}{p_g(z|x)} \right] \\ &= \mathbb{E}_{q_f} \left[ \log \frac{p_g(x, z)}{q_f(z|x)} \frac{q_f(z|x)}{p_g(z|x)} \right] \\ &= \mathbb{E}_{q_f} \left[ \log \frac{p_g(x, z)}{q_f(z|x)} \right] + \mathbb{E}_{q_f} \left[ \log \frac{q_f(z|x)}{p_g(z|x)} \right] \\ &= \text{ELBO}(x; p_g, q_f) + KL(q_f(z|x), p_g(z|x)) \\ &\Rightarrow \text{ELBO}(x; p_g, q_f) = \log p_g(x) - KL(q_f(z|x), p_g(z|x)) \\ &\leq \log p_g(x)\end{aligned}$$

Lower bound! Tight if  $q_f(z|x) = p_g(z|x)$

# The ELBO Can Be Interpreted As a Reconstruction Error Term and a Regularization Term

- Minimizing the negative yields error + regularization:

$$\begin{aligned}\text{ELBO}(x; p_g, q_f) &= \mathbb{E}_{q_f} \left[ \log \frac{p_g(x, z)}{q_f(z|x)} \right] \\ &= \mathbb{E}_{q_f} \left[ \log \frac{p_g(z)p_g(x|z)}{q_f(z|x)} \right] \\ &= \mathbb{E}_{q_f} [\log p_g(x|z)] + \mathbb{E}_{q_f} \left[ \log \frac{p_g(z)}{q_f(z|x)} \right] \\ &= \mathbb{E}_{q_f} [\log p_g(x|z)] - \mathbb{E}_{q_f} \left[ \log \frac{q_f(z|x)}{p_g(z)} \right] \\ &= \underbrace{\mathbb{E}_{q_f} [\log p_g(x|z)]}_{\text{Reconstruction}} - \underbrace{KL(q_f(z|x), p_g(z))}_{\text{Regularization}}\end{aligned}$$

# Optimizing the ELBO Objective Does Two Things Simultaneously

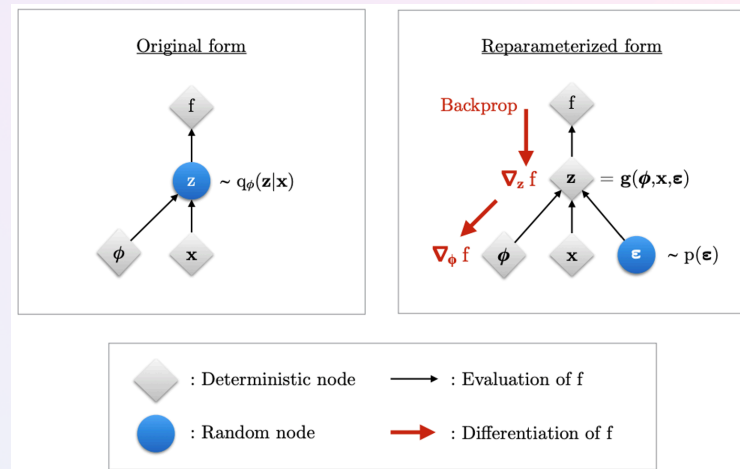
$$\min_{f,g} -\frac{1}{n} \sum_i (\mathbb{E}_{q_f}[\log p_g(x_i|z_i)] - KL(q_f(z_i|x_i), p_g(z_i)))$$

- $p_g$  - Optimize the bound on the likelihood.
- $q_f$  - Improve the bound (i.e., make it tighter).

# Reparameterization Trick Allows Us to Compute Gradients for $q_f$

$$\min_{f,g} -\frac{1}{n} \sum_{i=1}^n \mathbb{E}_{q_f(z_i|x_i)} [\log p_g(x_i | z_i)] + \text{KL}(q_f(z_i | x_i) \| p_g(z_i))$$

$$\min_{f,g} -\frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\epsilon} [\log p_g(x_i | z_i = f(x_i) + \epsilon)] + \mathbb{E}_{\epsilon} \left[ \log \frac{q_f(f(x_i) + \epsilon | x_i)}{p_g(f(x_i) + \epsilon)} \right]$$



In this figure,  $f$  is the loss function,  $g$  is the reparameterization function, and  $\phi$  are the parameters of the encoder.

Kingma, D. P., & Welling, M. (2019). An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4), 307-392. In this figure,  $f$  is the loss function,  $g$  is the reparameterization function, and  $\phi$  are the parameters of the encoder.

# Putting It All Together: the VAE Algorithm Using Sgd

1. Get minibatch of data  $x$ .
2. Pass through encoder to get  $\mu, \sigma^2 = f(x)$ .
3. Sample from  $z \sim q_f(z|x, (\mu, \sigma^2) = f(x))$  using reparametrization trick.
4. Pass through decoder to get output parameters  $\theta = g(z)$ .
5. Compute conditional log likelihood of  $p_g(x|z, \theta = g(z))$ .
6. Loss is negative conditional log likelihood + KL term.
7. Backpropagate to gradients for both  $g$  and  $f$  and update model.