Introduction to Machine Learning:

Lecture 4 – Unsupervised Learning

Michael Kagan

SLAC NATIONAL ACCELERATOR LABORATORY

TRISEP Summer School
July 8-12, 2024

The Plan

- Lecture 1 – Machine Learning Fundamentals
- Lecture 2 – Intro to Neural Networks
- Lecture 3 – Intro to Deep Learning
- Lecture 4 – Intro to Unsupervised Learning
- Lecture 5 – Intro to Deep Generative Models

Beyond Regression and Classification

Beyond Regression and Classification

- Not all tasks are predicting a label from features, as in classification and regression

- May want to model a high-dim. signal
  - Data synthesis / simulation
  - Density estimation
  - Anomaly detection
  - Denoising, super resolution
  - Data compression
  - …

- Often don't have labels → Unsupervised Learning

Unsupervised Learning

- Our goal is to study the data density $p(x)$

- Even w/o labels, aim to characterize the distribution

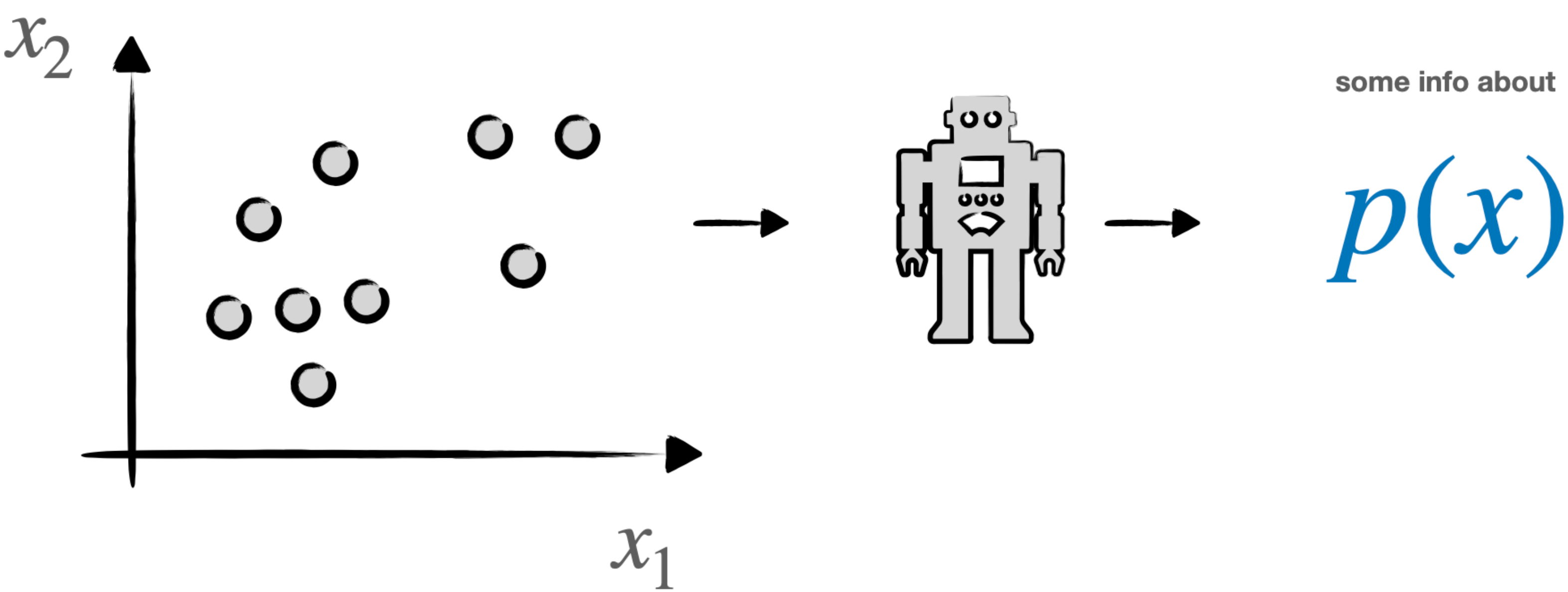


Image credit: L. Heinrich

Probability Models

A process

 $\rightarrow \mathbb{R}^2$

Generating new samples from randomness

A formula

$\mathbb{R}^2 \rightarrow \mathbb{R}$

$$p_{\mu,\Sigma}(x) = \frac{1}{\sqrt{|2\pi\Sigma|}} exp\left(-\frac{1}{2}(x-\mu)^T\Sigma(x-\mu)\right)$$

Evaluating the Probability for a given sample

"Understanding $p(x)$" – ability to do either or both of these

Image credit: L. Heinrich

Probability Models as Sampling a Process

- In many cases, we don't have a theory of the underlying process → *Can still learn to sample*

- Deep learning can be very good at this!

$$\text{face} \sim p(\text{face})$$

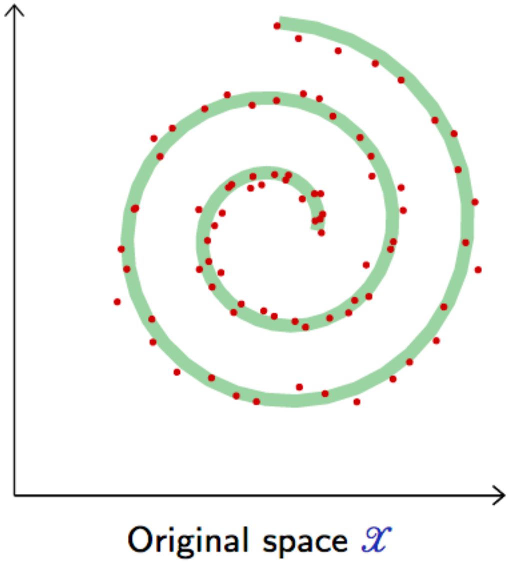


https://thispersondoesnotexist.com/

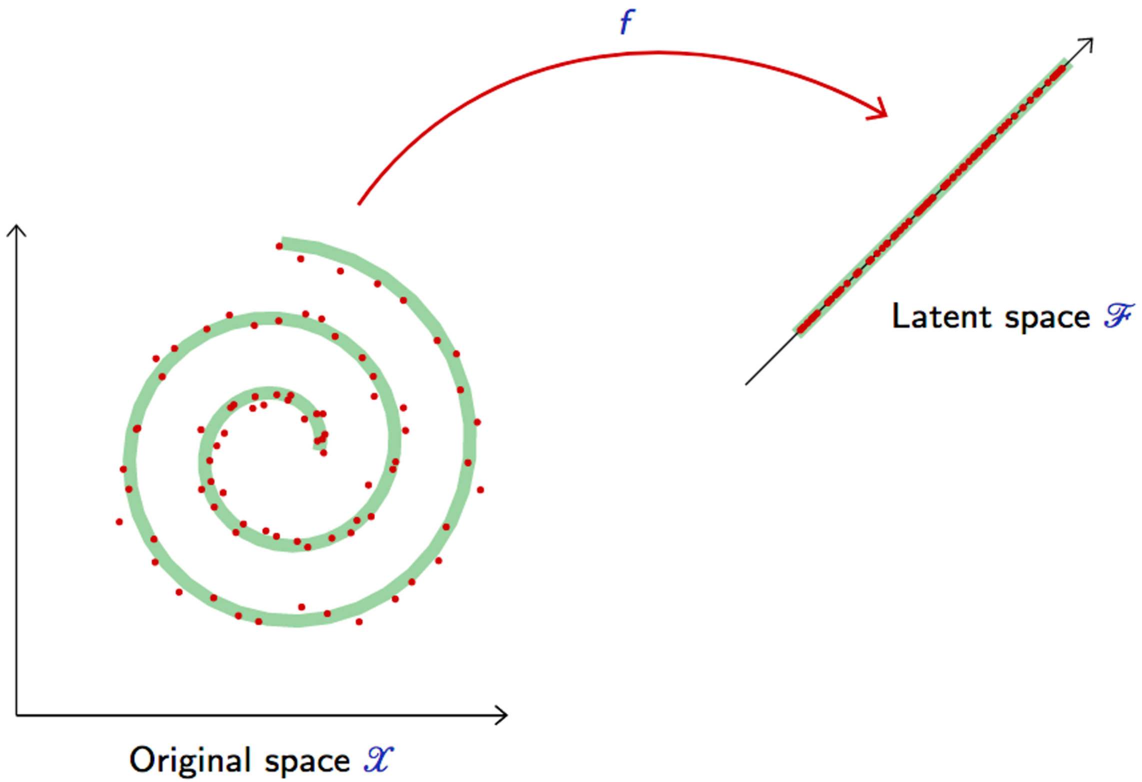
Learning Objective

- Unsupervised learning is more heterogeneous than supervised learning
- Many architectures, losses, learning strategies
- Often constructed so model converges to $p(x)$
  - Variational inference, Adversarial learning, Self-supervision, …
- Often framed as **modeling the lower dimensional "meaningful degrees of freedom"** that describe the data

Modeling Data and Meaningful Degrees of Freedom

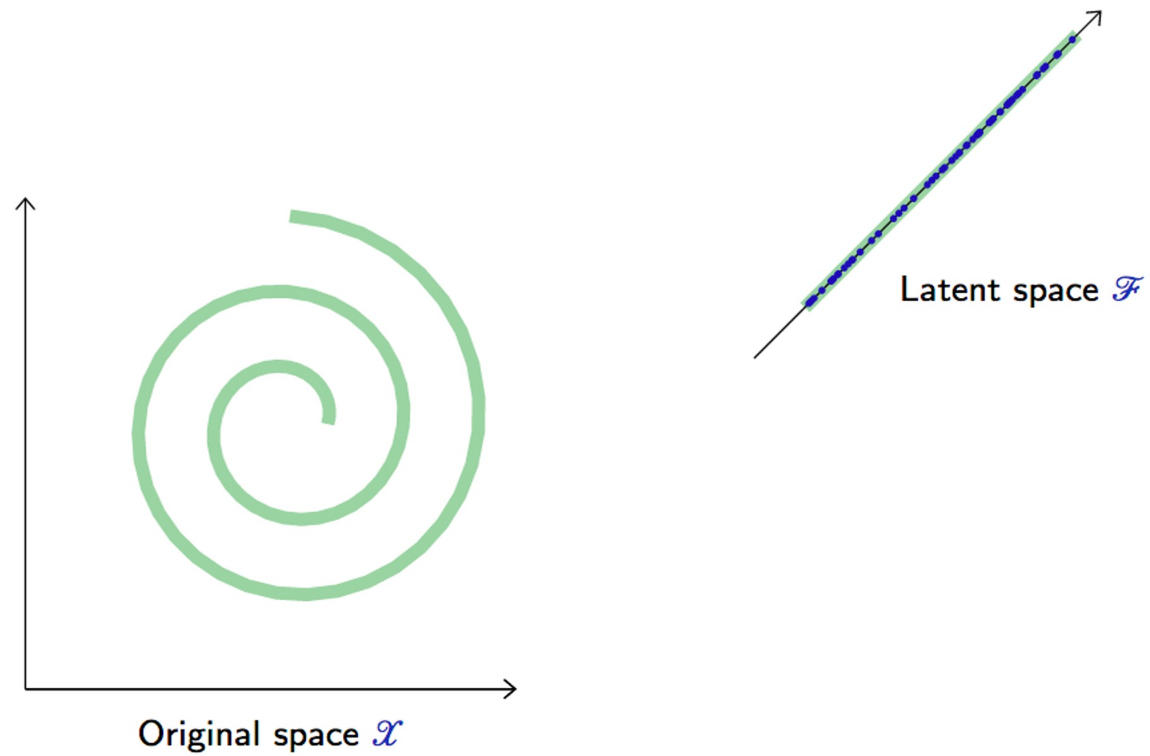


Modeling Data and Meaningful Degrees of Freedom



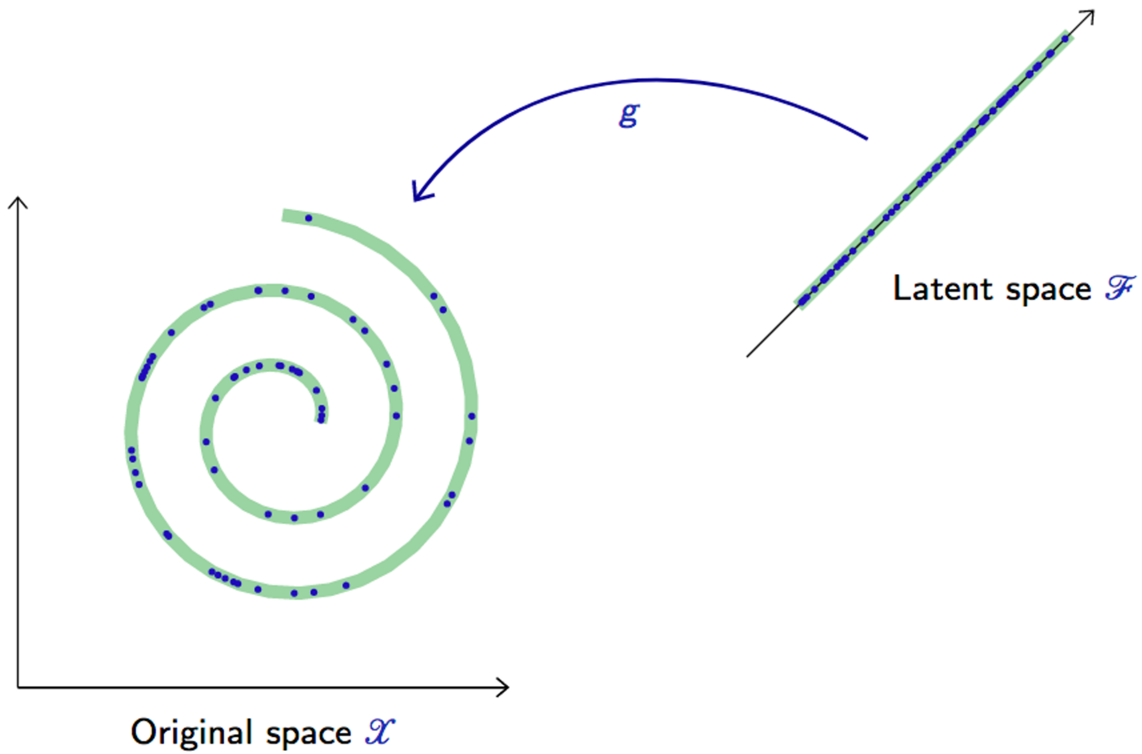
Fleuret, [Deep Learning Course]

Modeling Data and Meaningful Degrees of Freedom



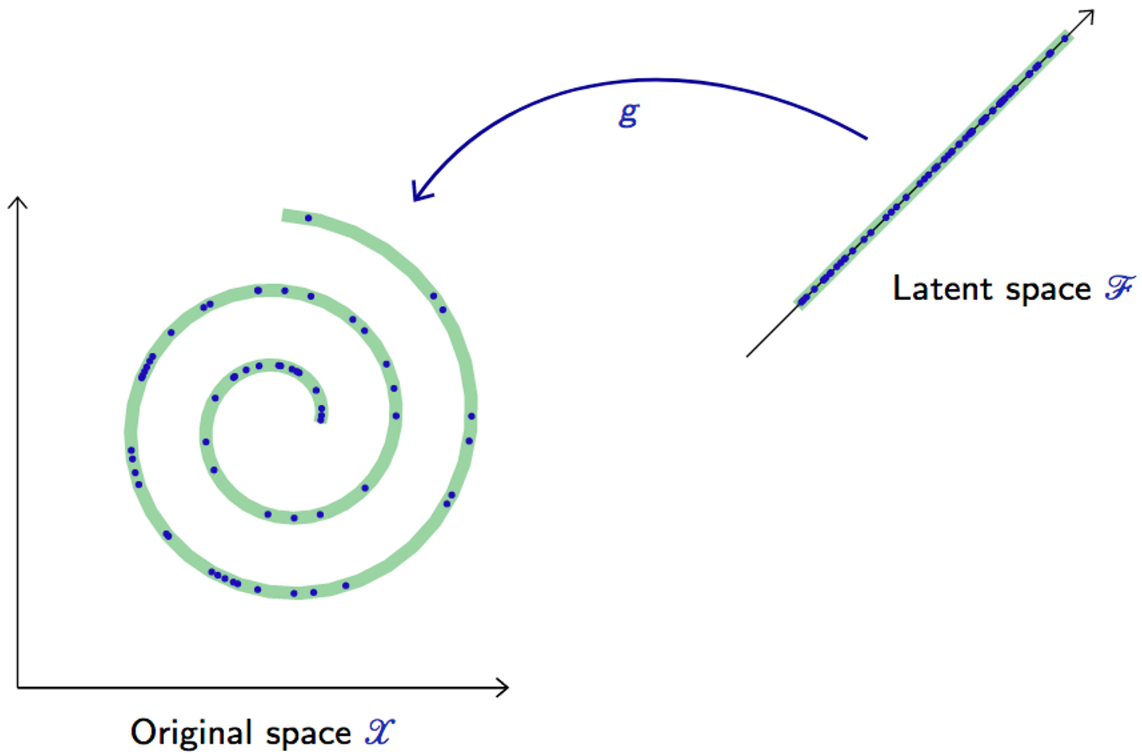
Fleuret, Deep Learning Course

Modeling Data and Meaningful Degrees of Freedom



Fleuret, [Deep Learning Course]

How can we find the "meaningful degrees of freedom" in the data?



Meaningful Representations

- Dimensionality Reduction / Compression

- Can we learn to:

  1. Compress the data to a *latent space* with smaller number of dimensions

  2. Recover the original data from this latent space?

- Latent space must encode and retain the important information about the data

Principle Components Analysis

Dimensionality Reduction

Find a low dimensional (less complex) representation of the data with a mapping $Z=h(X)$

Principle Components Analysis

- Given data $\{\mathbf{x}_i\}_{i=1\ldots N}$ can we find a directions in features space that explain most variation of data?

Principle Components Analysis

- Given data $\{\mathbf{x}_i\}_{i=1\ldots N}$ can we find a directions in features space that explain most variation of data?

- Data covariance: $$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})^2$$

Principle Components Analysis

- Given data $\{\mathbf{x}_i\}_{i=1\ldots N}$ can we find a directions in features space that explain most variation of data?
- Data covariance: $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})^2$
- Let $\mathbf{u}_1$ be the projected direction, we can solve:

$$\mathbf{u}_1^* = \arg\max_{\mathbf{u}_1} \; \underbrace{\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1}_{\text{Variance of projected data}} + \lambda\underbrace{(1 - \mathbf{u}_1^T \mathbf{u}_1)}_{\text{Unit length vector constraint}}$$

Principle Components Analysis

- Given data $\{\mathbf{x}_i\}_{i=1\ldots N}$ can we find a directions in features space that explain most variation of data?
- Data covariance: $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})^2$
- Let $\mathbf{u}_1$ be the projected direction, we can solve:

$$\mathbf{u}_1^* = \arg\max_{\mathbf{u}_1} \underbrace{\mathbf{u}_1^T\mathbf{S}\mathbf{u}_1}_{\text{Variance of projected data}} + \lambda\underbrace{(1 - \mathbf{u}_1^T\mathbf{u}_1)}_{\text{Unit length vector constraint}}$$

$$\nabla_{u_1}[\,\boldsymbol{u}_1^T\boldsymbol{S}\boldsymbol{u}_1 + \lambda(1 - \boldsymbol{u}_1^T\boldsymbol{u}_1)\,] = \boldsymbol{S}\boldsymbol{u}_1 - \lambda\boldsymbol{u}_1 = 0$$

Principle Components Analysis

- Given data $\{\mathbf{x}_i\}_{i=1\ldots N}$ can we find a directions in features space that explain most variation of data?
- Data covariance: $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})^2$
- Let $\mathbf{u}_1$ be the projected direction, we can solve:

$$\mathbf{u}_1^* = \arg\max_{\mathbf{u}_1} \underbrace{\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1}_{\text{Variance of projected data}} + \lambda\underbrace{(1 - \mathbf{u}_1^T \mathbf{u}_1)}_{\text{Unit length vector constraint}}$$

$$\rightarrow \mathbf{S}\mathbf{u}_1 = \lambda \mathbf{u}_1$$

Principle Components Analysis

- Given data $\{\mathbf{x}_i\}_{i=1\ldots N}$ can we find a directions in features space that explain most variation of data?

- Data covariance: $$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})^2$$

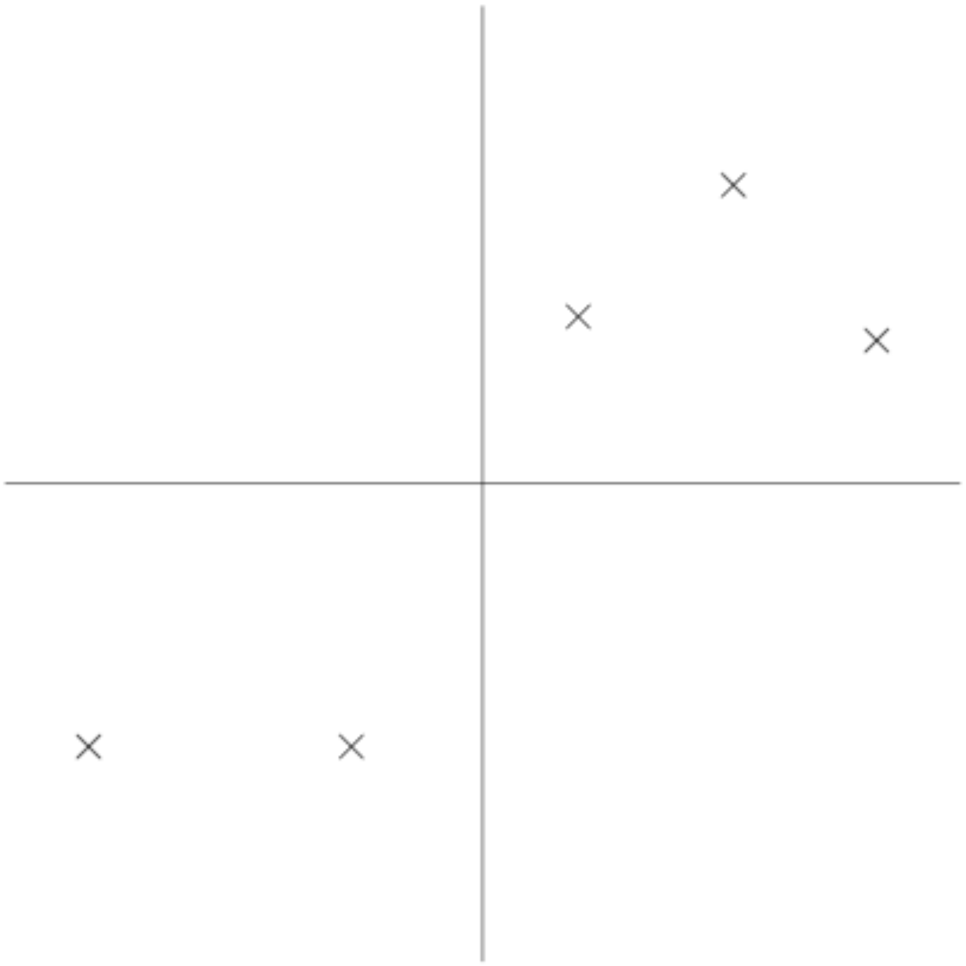
- Let $\mathbf{u}_1$ be the projected direction, we can solve:

$$\mathbf{u}_1^* = \arg\max_{\mathbf{u}_1} \underbrace{\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1}_{\text{Variance of projected data}} + \lambda\underbrace{(1 - \mathbf{u}_1^T \mathbf{u}_1)}_{\text{Unit length vector constraint}}$$

$$\rightarrow \mathbf{S}\mathbf{u}_1 = \lambda \mathbf{u}_1$$

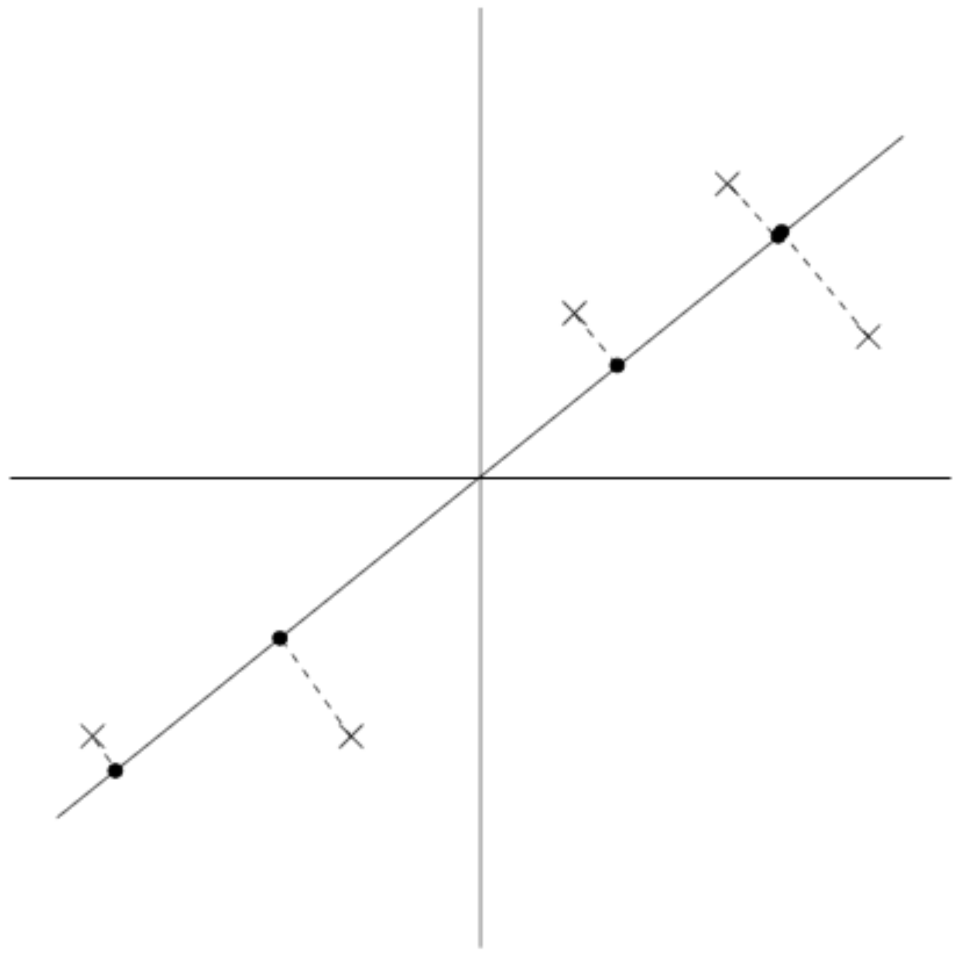
- *Principle components* are the **eigenvectors** of the data covariance matrix!
  - Eigenvalues are the variance explained by that component

PCA Example



[Ng]

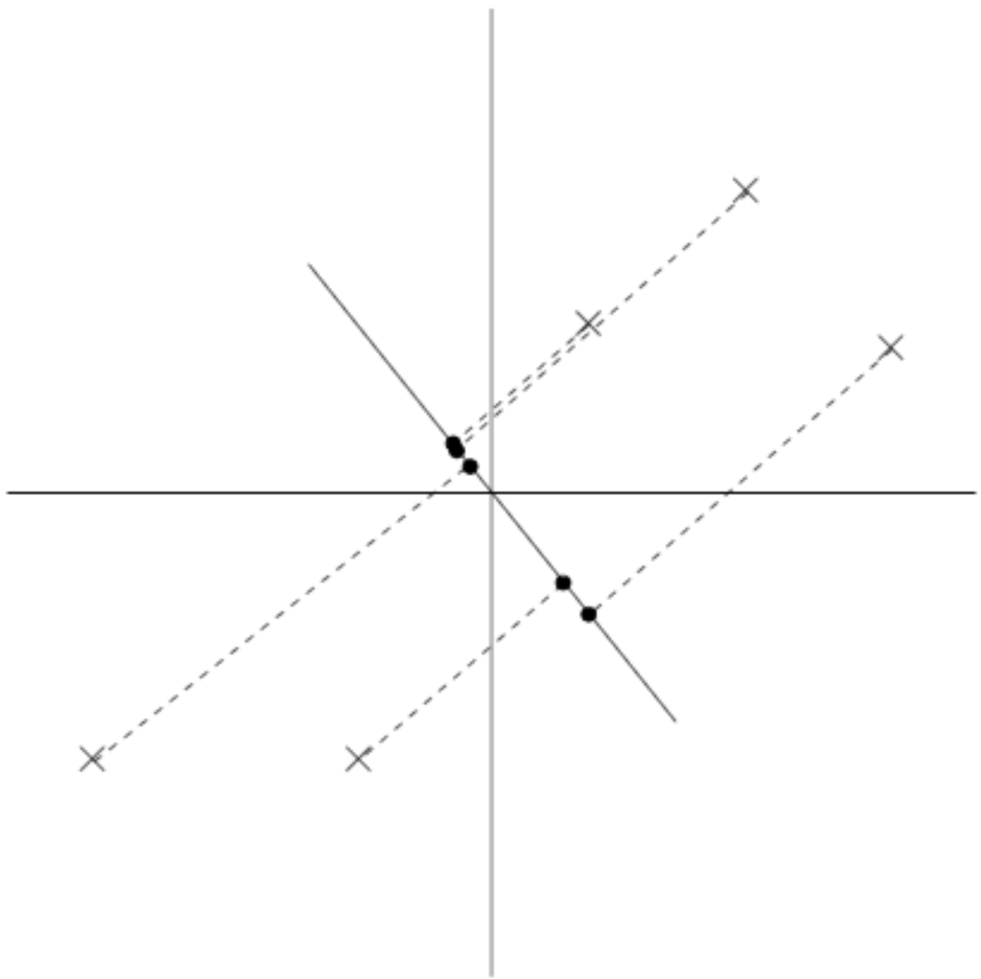
PCA Example



First principle component, projects on to this axis have large variance

[Ng]

PCA Example



Second principle component, projects have small variance

[Ng]

Autoencoders

Autoencoders

- Map a space to itself through a compression

$$x \rightarrow z \rightarrow \hat{x}$$

Data

Latent space

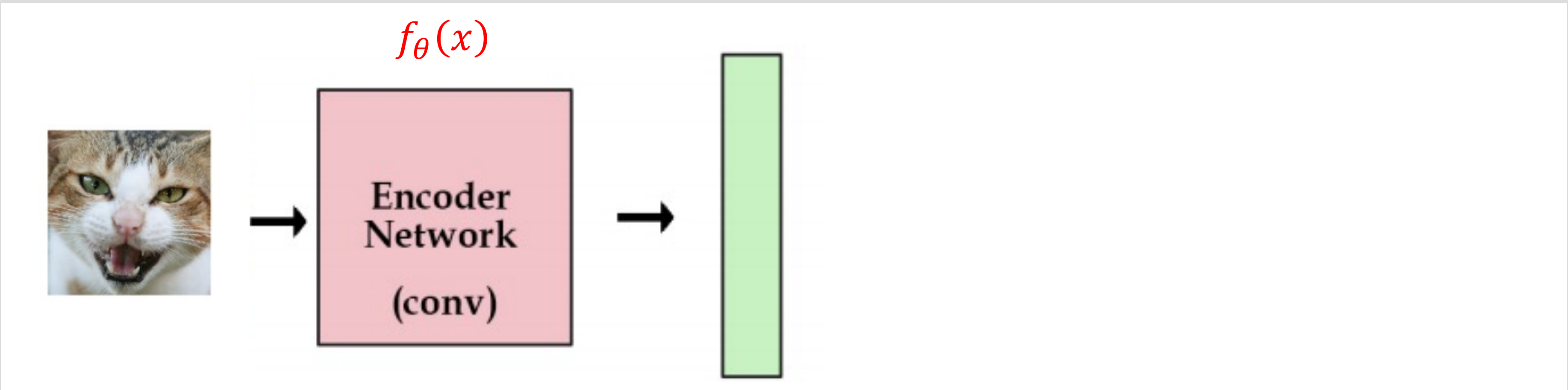
Reconstruction

Autoencoders

- Map a space to itself through a compression

$$x \to z \to \hat{x}$$

  - **Encoder**: Map from data to a lower dim. latent space
    - Neural network $f_\theta(x)$ with parameters θ

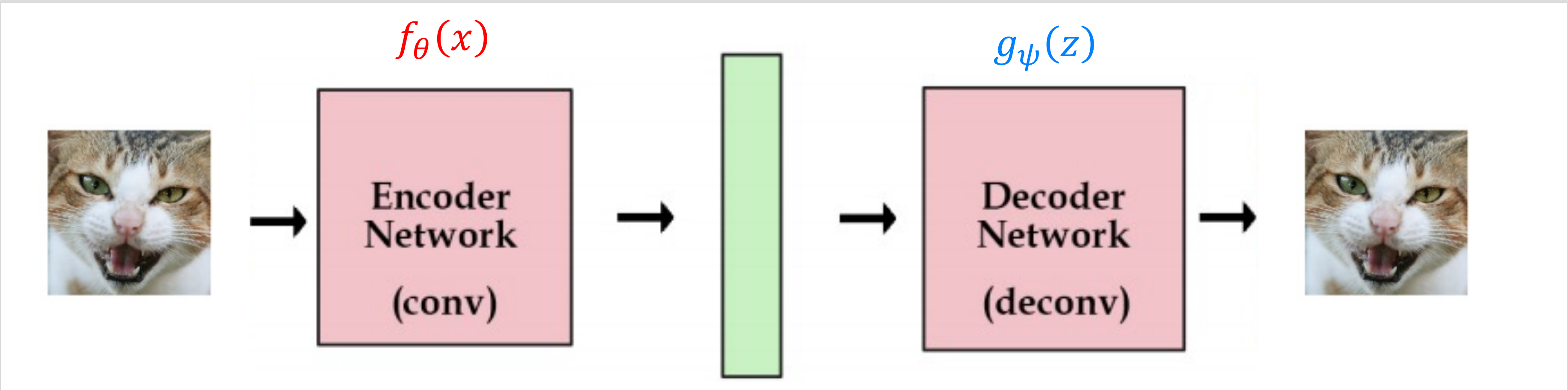


Autoencoders

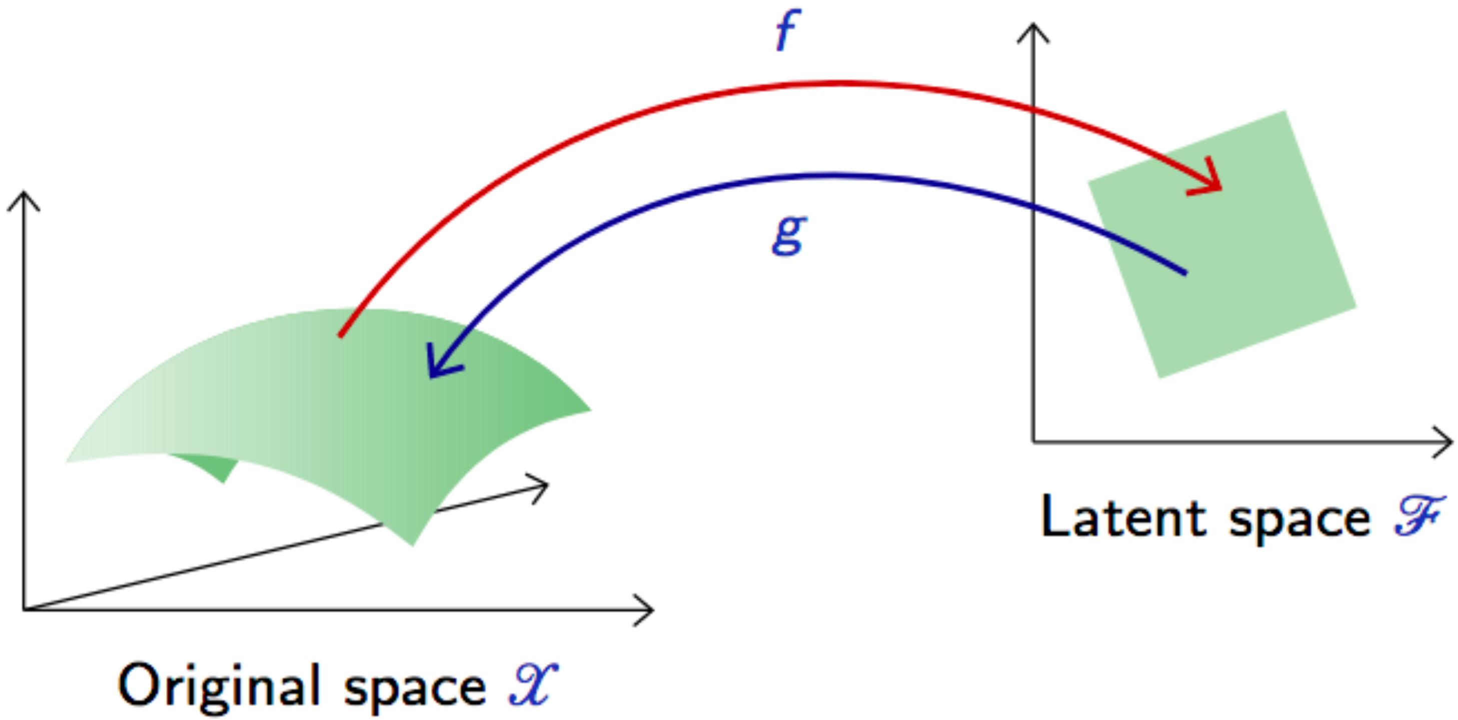
- Map a space to itself through a compression

$$x \to z \to \hat{x}$$

  - **Encoder**: Map from data to a lower dim. latent space
    - Neural network $f_\theta(x)$ with parameters θ
  - **Decoder**: Map from latent space back to data space
    - Neural network $g_\psi(z)$ with parameters ψ



Autoencoder Mappings



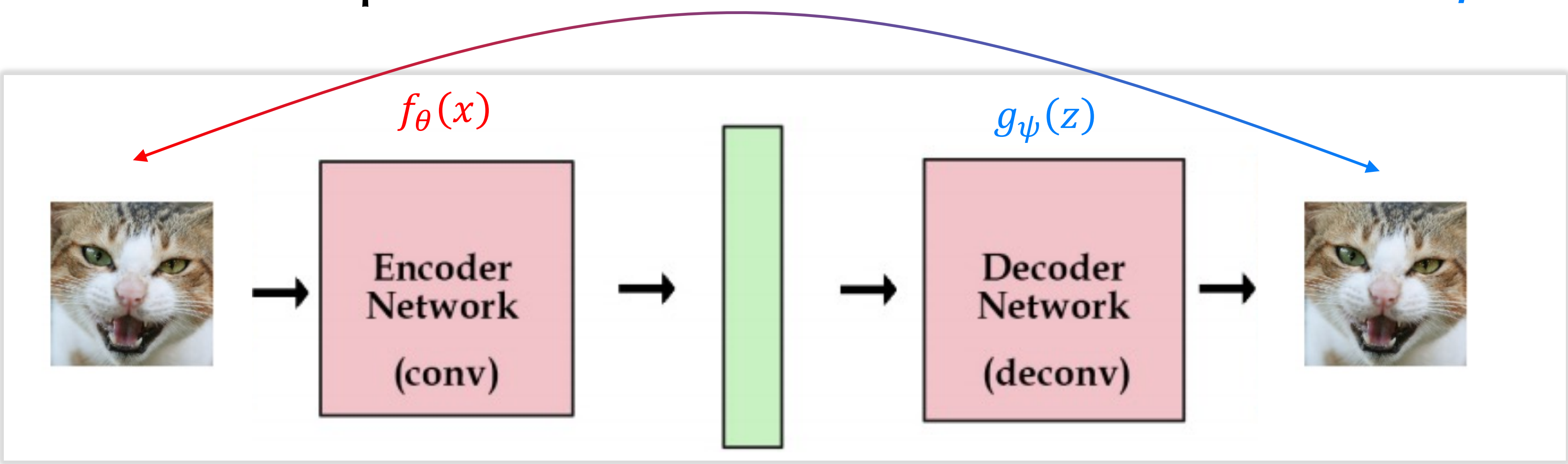
- Latent space is of lower dimension than data
- Model must learn a “good” parametrization and capture dependencies between components

Autoencoder Loss

$$L(\theta, \psi) = \frac{1}{N} \sum_n \| x_n - g_\psi(f_\theta(x_n)) \|^2$$

- **Loss**: mean *reconstruction loss* (MSE) between data and encoded-decoded data

- Min. over params. of encoder (θ) and decoder (ψ).



Autoencoder Loss

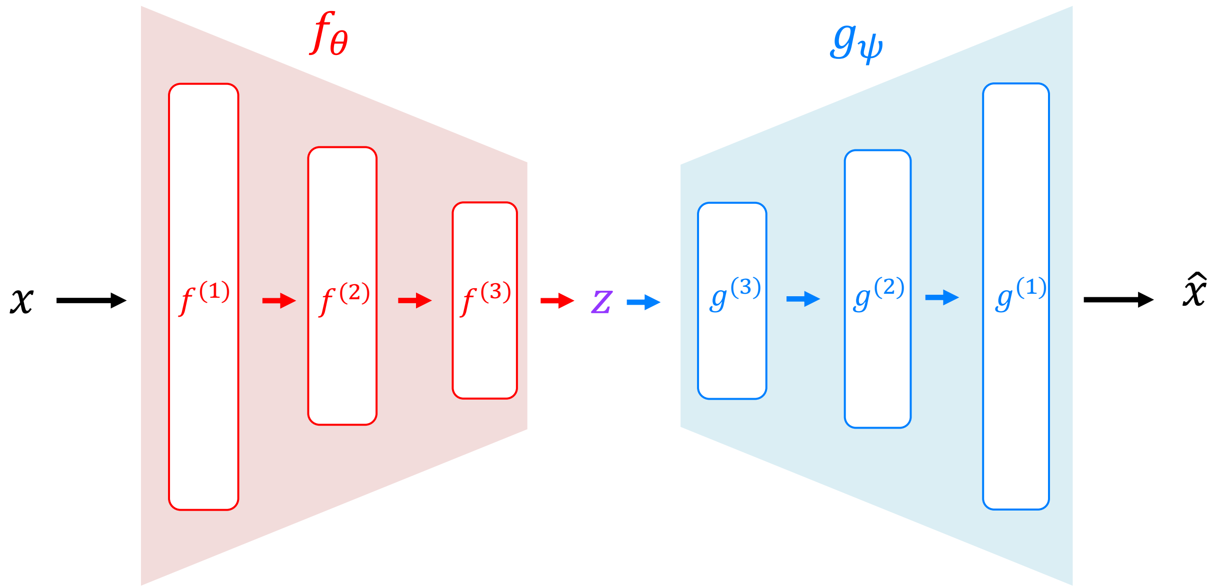
$$L(\theta, \psi) = \frac{1}{N} \sum_n \left\| x_n - g_\psi\left(f_\theta(x_n)\right) \right\|^2$$

- Loss: mean *reconstruction loss* (MSE) between data and encoded-decoded data

- Min. over params. of encoder (θ) and decoder (ψ).

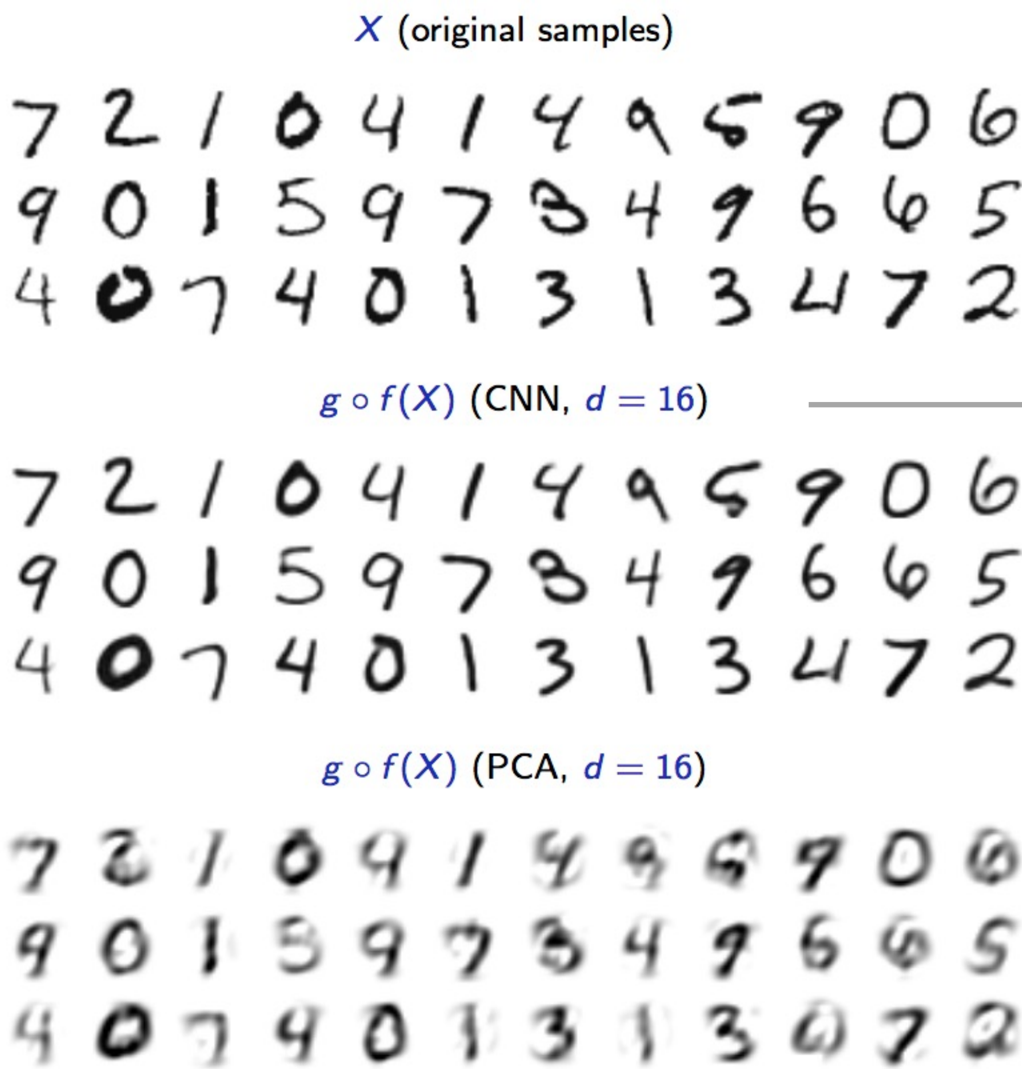
- NOTE: if $f_\theta(x)$ and $g_\psi(z)$ are linear, optimal solution given by Principle Components Analysis

Deep Autoencoder



- When f_θ and g_ψ are multiple neural network layers, can learn complex mappings
  - f_θ and g_ψ can be Fully Connected, CNNs, RNNs, etc.
  - Choice of network structure will depend on data

Deep Convolutional Autoencoder

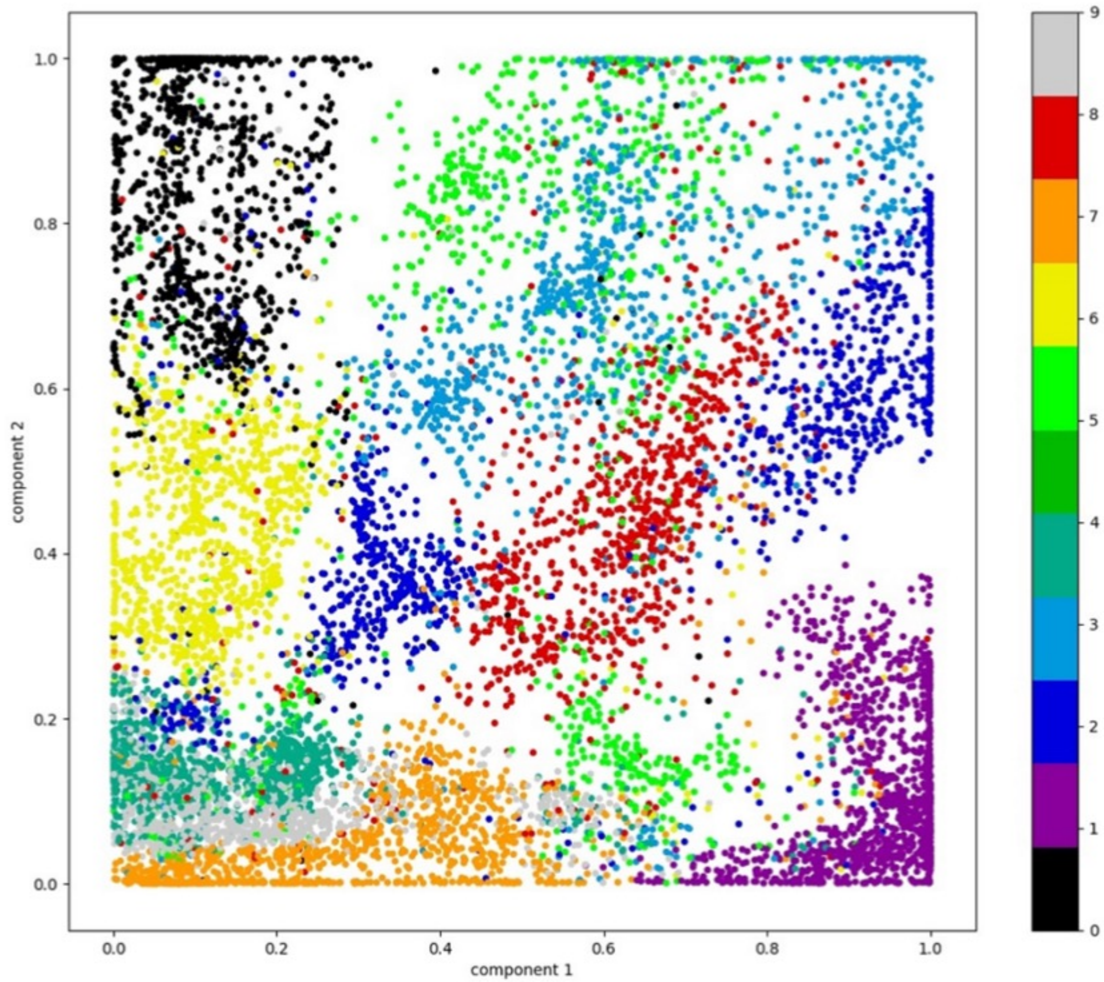


f_θ and g_ψ are five convolutional layers

Fleuret, Deep Learning Course

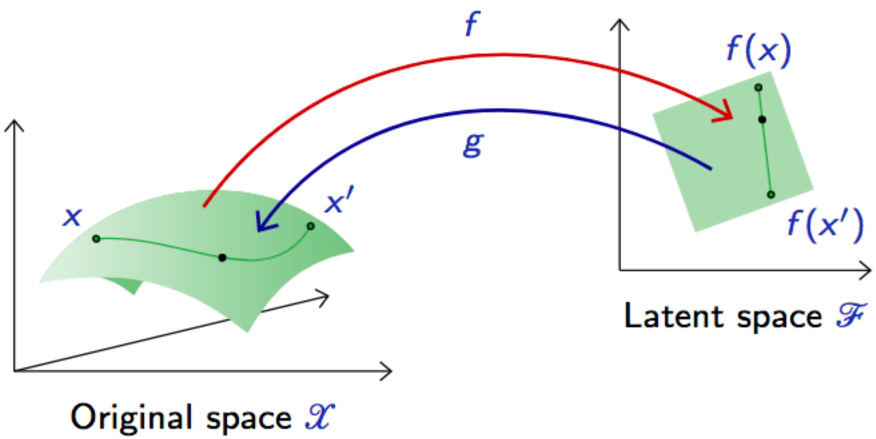
The Latent Space

- Can look at latent space to see how the model arranges the data



Interpolating in Latent Space

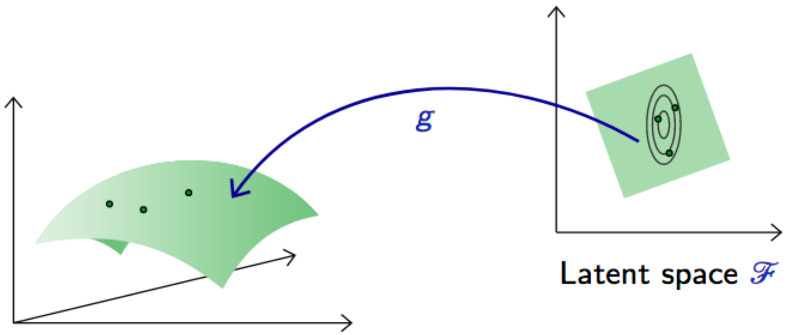
$$\alpha \in [0,1], \quad \xi(x, x', \alpha) = g((1-\alpha)f(x) + \alpha f(x')).$$



Autoencoder interpolation ($d = 8$)

Can We Generate Data with Decoder?

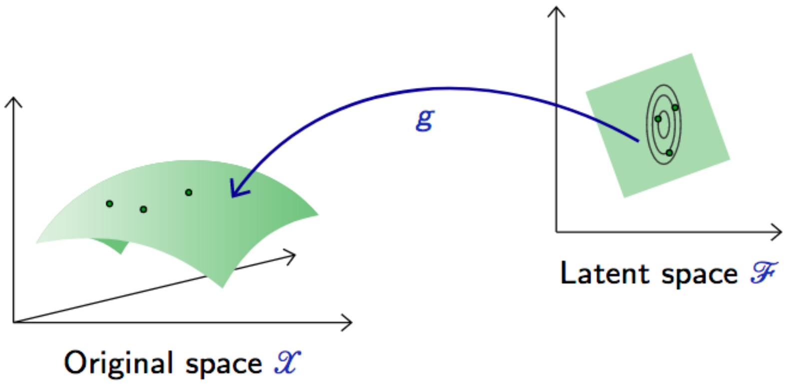
- Can we sample in latent space and decode to generate data?



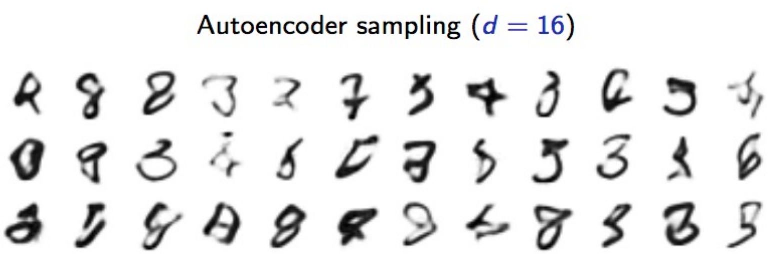
Fleuret, [Deep Learning Course]

Can We Generate Data with Decoder?

- Can we sample in latent space and decode to generate data?

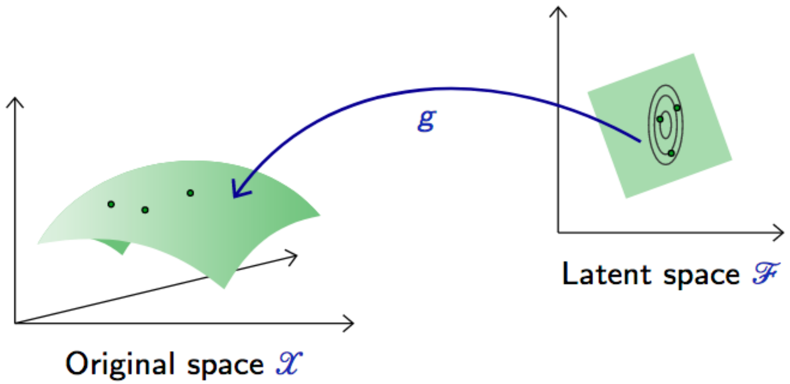


- What distribution to sample from in latent space?
  - Try Gaussian with mean and variance from data

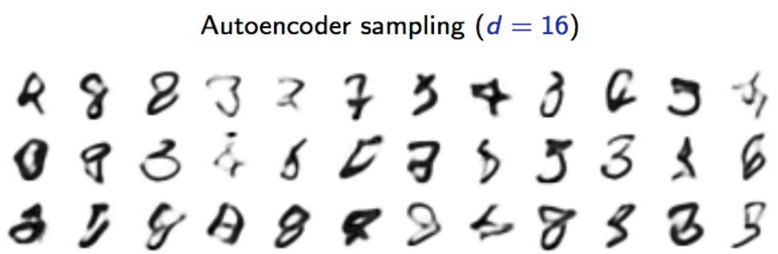


Can We Generate Data with Decoder?

- Can we sample in latent space and decode to generate data?



- What distribution to sample from in latent space?
  - Try Gaussian with mean and variance from data



- Don't know the right latent space density

Summary

- Unsupervised learning aims to characterize data, even without distributions

- Often framed as learning the meaningful degrees of freedom of a system

- We saw examples of powerful ways to learn these meaningful degrees of freedom, e.g. linearly with PCA and non-linearly with autoencoders

Backup