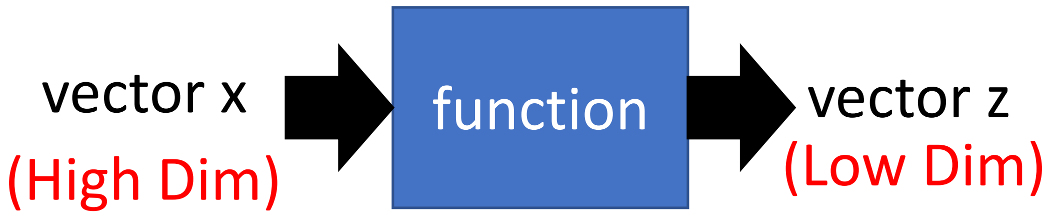
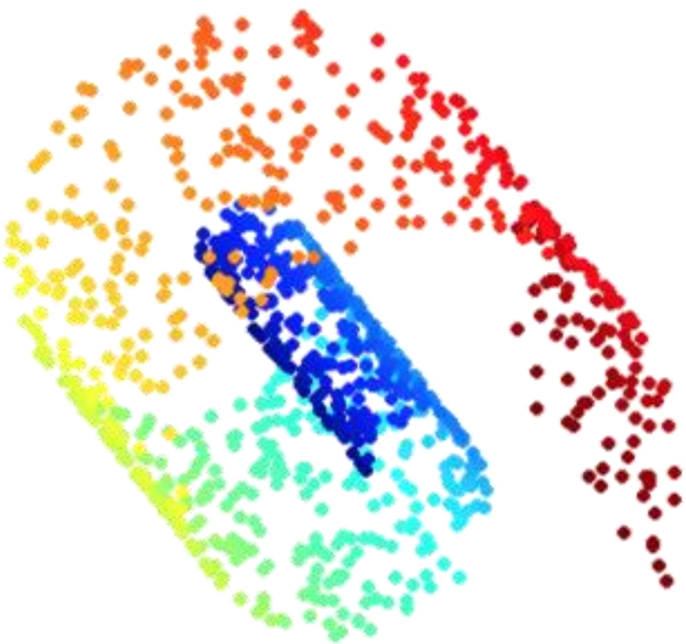
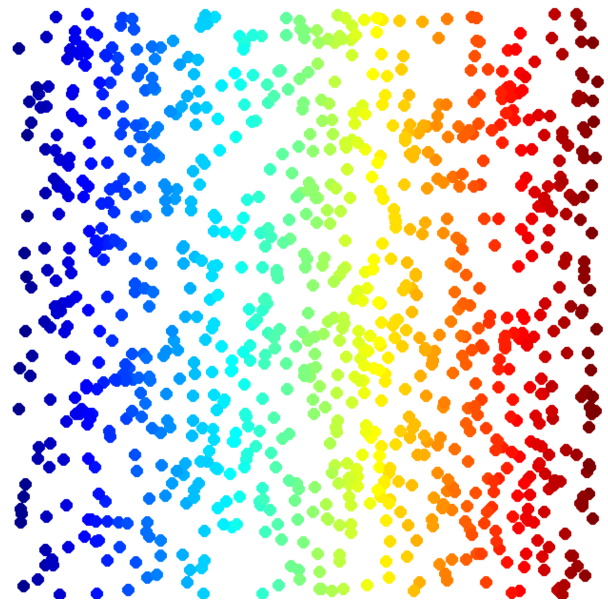
Unsupervised Learning:
Principle Component Analysis

Dimension Reduction





Looks like 3-D



Actually, 2-D

Clustering

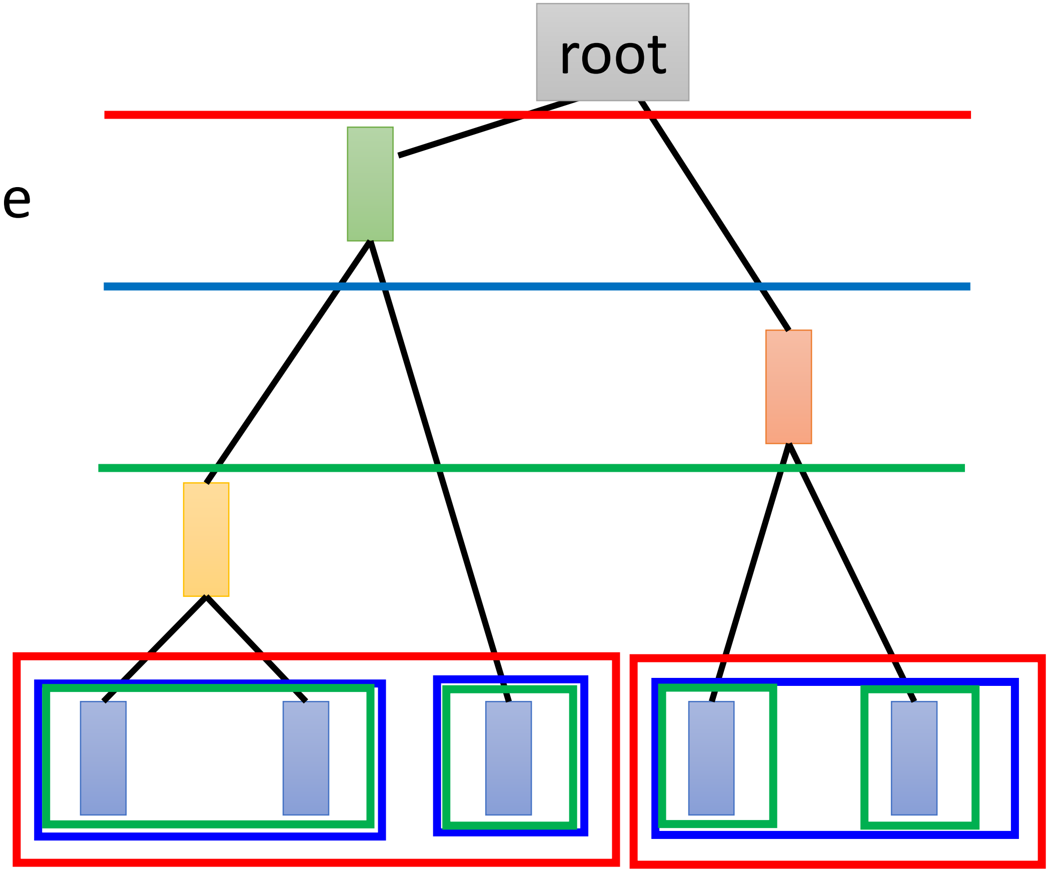
- K-means
  - Clustering $X = \{x^1, \cdots, x^n, \cdots, x^N\}$ into K clusters
  - Initialize cluster center c^i, i=1,2, … K (K random x^n from X)
  - Repeat
    - For all x^n in X: $b_i^n \begin{cases} 1 & x^n \text{ is most “\textbf{\textit{close}}” to } c^i \\ 0 & \text{Otherwise} \end{cases}$
    - Updating all c^i: $c^i = \sum_{x^n} b_i^n x^n \Big/ \sum_{x^n} b_i^n$

Clustering

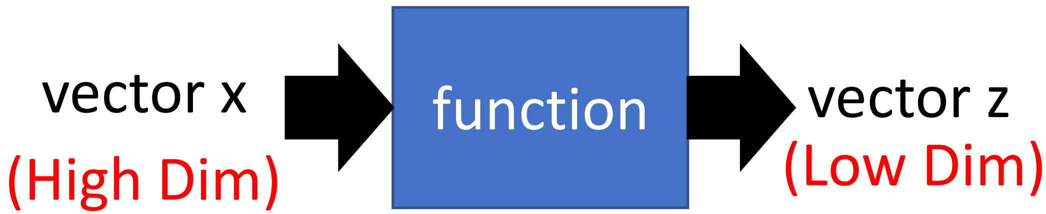
- Hierarchical Agglomerative Clustering (HAC)

Step 1: build a tree

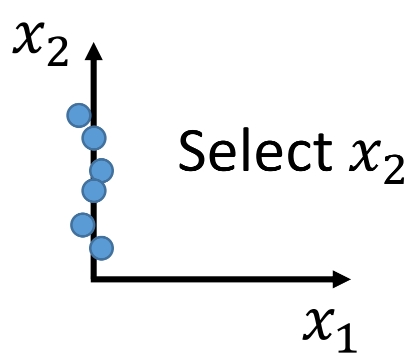
Step 2: pick a threshold

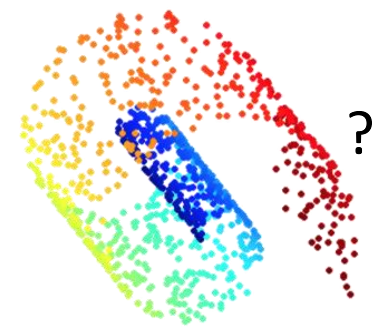


Distributed Representation



- Feature selection





- Principle component analysis (PCA)

$$z = Wx$$

PCA

$$z = Wx$$

Reduce to 1-D:

$$z_1 = w^1 \cdot x$$

$$z_2 = w^2 \cdot x$$

$$W = \begin{bmatrix} (w^1)^T \\ (w^2)^T \\ \vdots \end{bmatrix}$$

Orthogonal matrix

Project all the data points x onto w^1, and obtain a set of z_1

We want the variance of z_1 as large as possible

$$Var(z_1) = \frac{1}{N}\sum_{z_1}(z_1 - \bar{z_1})^2 \quad \|w^1\|_2 = 1$$

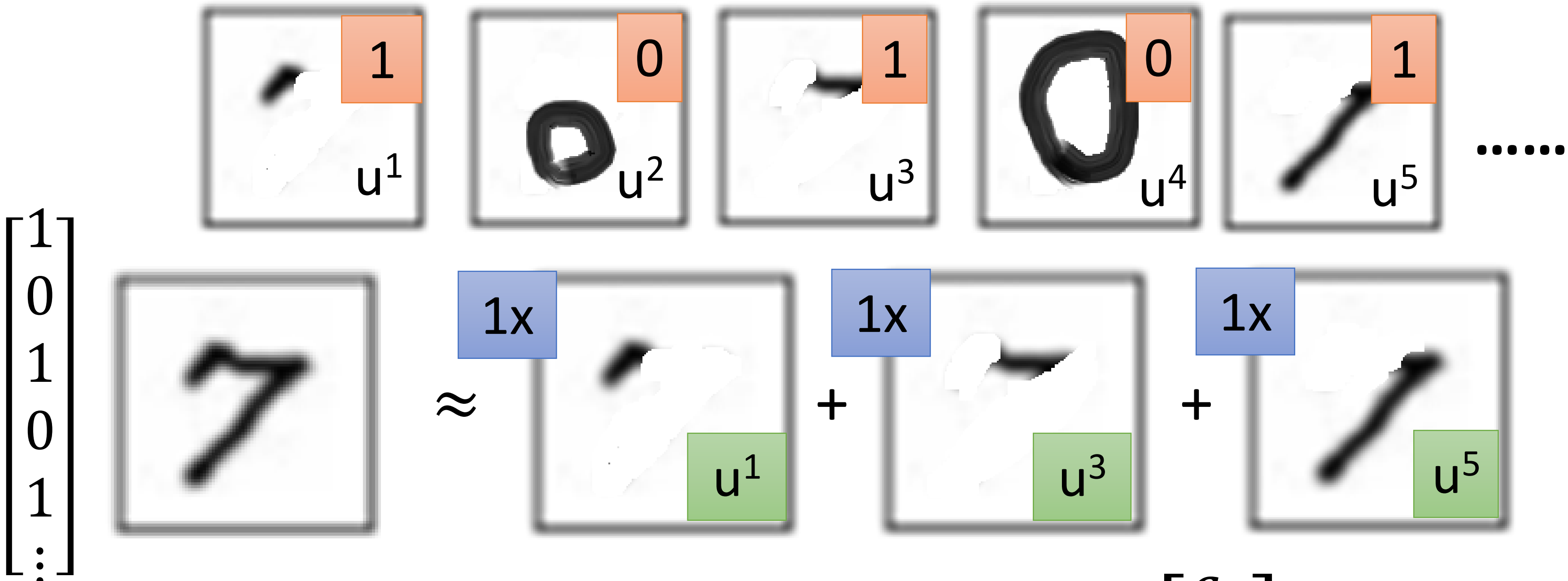
We want the variance of z_2 as large as possible

$$Var(z_2) = \frac{1}{N}\sum_{z_2}(z_2 - \bar{z_2})^2 \quad \|w^2\|_2 = 1$$

$$w^1 \cdot w^2 = 0$$

PCA – Another Point of View

Basic Component:



$$x \approx c_1 u^1 + c_2 u^2 + \cdots + c_K u^K + \bar{x}$$

Pixels in a digit image

component

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_K \end{bmatrix}$$

Represent a digit image

PCA – Another Point of View

$$x - \bar{x} \approx c_1 u^1 + c_2 u^2 + \cdots + c_K u^K \;\; = \hat{x}$$

Reconstruction error:
$\| (x - \bar{x}) - \hat{x} \|_2$

Find $\{u^1, \ldots, u^K\}$ minimizing the error

$$L = \min_{\{u^1,\ldots,u^K\}} \sum \left\| (x - \bar{x}) - \underbrace{\left(\sum_{k=1}^{K} c_k u^k \right)}_{\hat{x}} \right\|_2$$

PCA: $z = Wx$

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_K \end{bmatrix} = \begin{bmatrix} (w_1)^{\mathrm{T}} \\ (w_2)^{\mathrm{T}} \\ \vdots \\ (w_K)^{\mathrm{T}} \end{bmatrix} x$$

$\{w^1, w^2, \ldots w^K\}$ (from PCA) is the component $\{u^1, u^2, \ldots u^K\}$ minimizing L

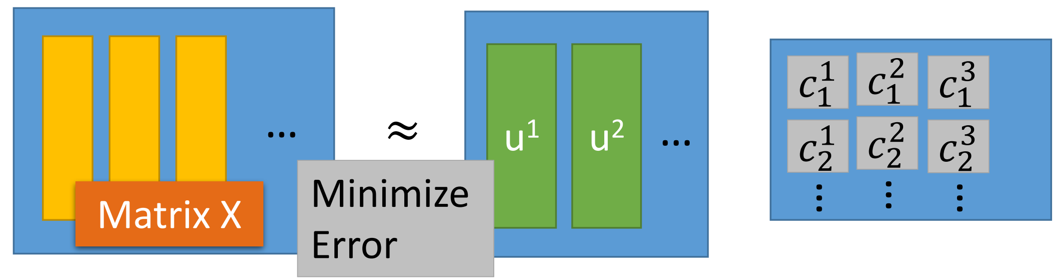
Proof in [Bishop, Chapter 12.1.2]

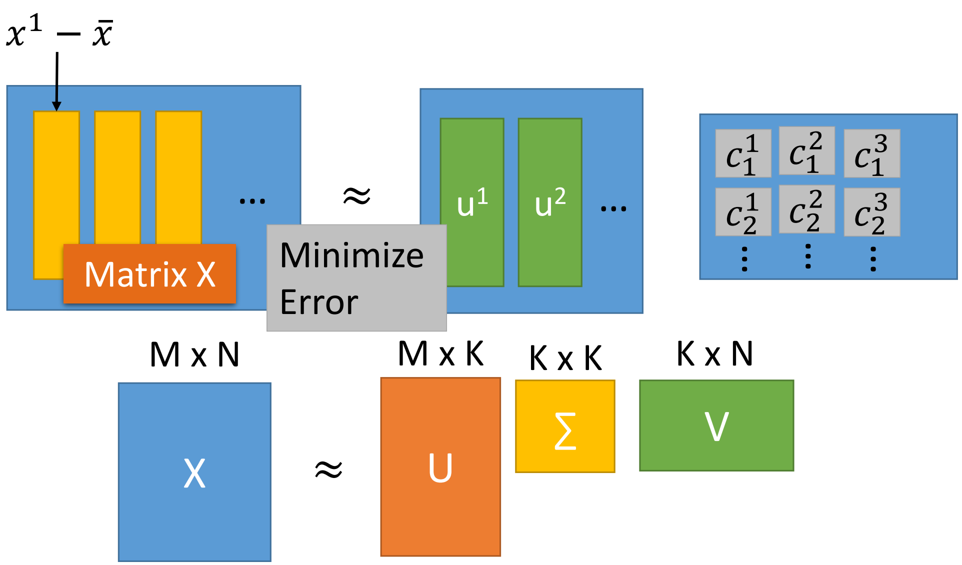
$$x - \bar{x} \approx c_1 u^1 + c_2 u^2 + \cdots + c_K u^K \;\; = \hat{x}$$

Reconstruction error:
$\| (x - \bar{x}) - \hat{x} \|_2$

Find $\{u^1, \ldots, u^K\}$ minimizing the error

$$x^1 - \bar{x} \approx c_1^1 u^1 + c_2^1 u^2 + \cdots$$
$$x^2 - \bar{x} \approx c_1^2 u^1 + c_2^2 u^2 + \cdots$$
$$x^3 - \bar{x} \approx c_1^3 u^1 + c_2^3 u^2 + \cdots$$
$$\vdots$$





K columns of U: a set of orthonormal eigen vectors corresponding to the K largest eigenvalues of XX^T

This is the solution of PCA

PCA looks like a neural network with one hidden layer (linear activation function)

Autoencoder

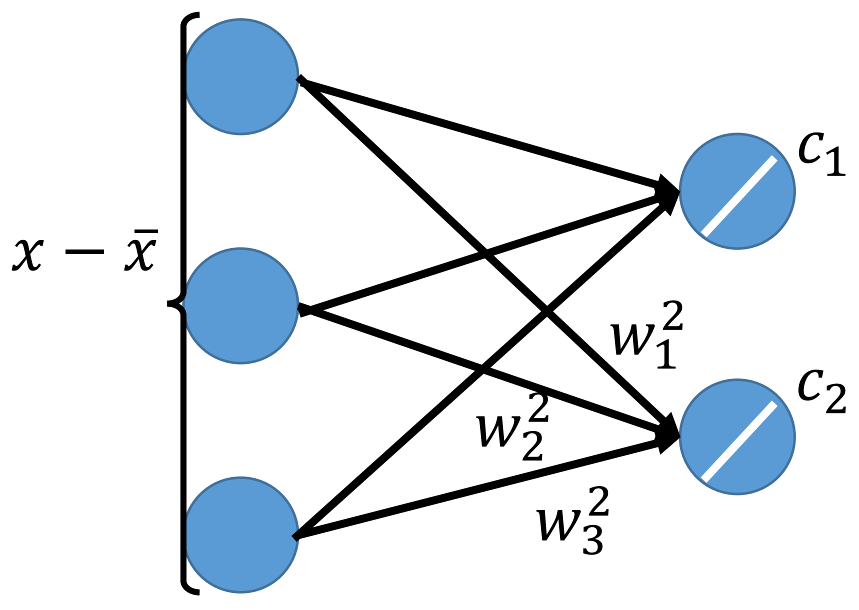
If $\{w^1, w^2, \dots w^K\}$ is the component $\{u^1, u^2, \dots u^K\}$

$$\hat{x} = \sum_{k=1}^{K} c_k w^k \Leftrightarrow x - \bar{x}$$

To minimize reconstruction error:

$$c_k = (x - \bar{x}) \cdot w^k$$

$K = 2$:



PCA looks like a neural network with one hidden layer (linear activation function)

Autoencoder

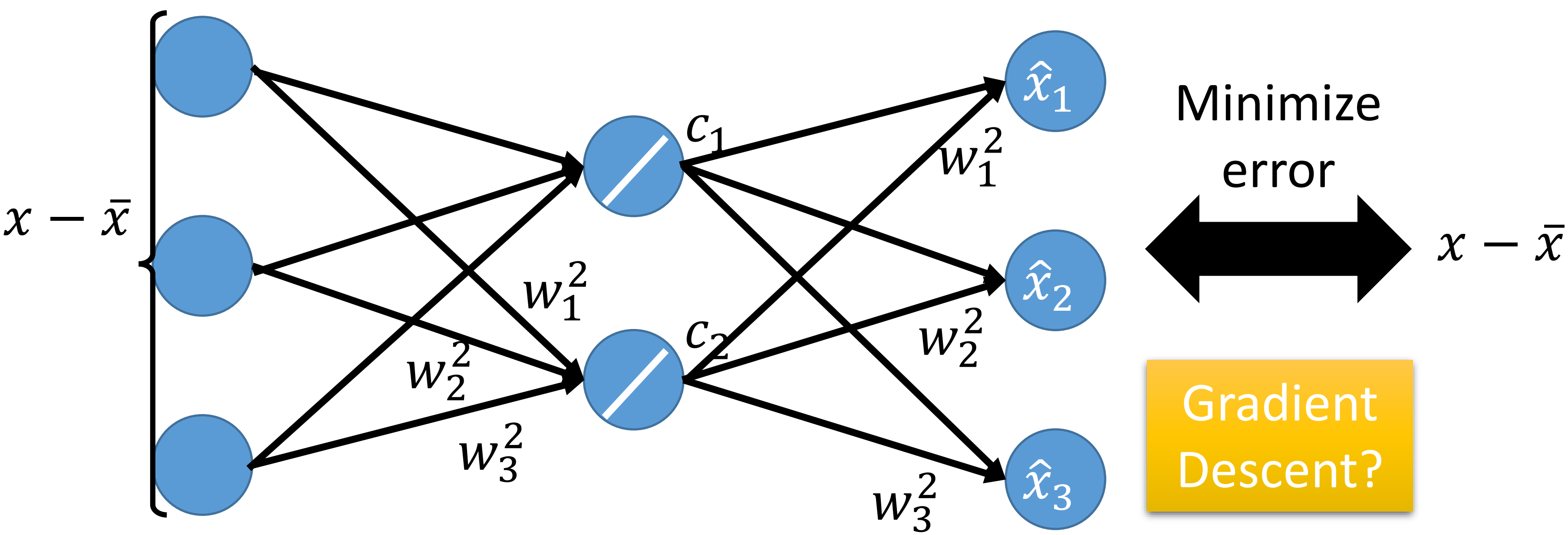
If $\{w^1, w^2, \dots w^K\}$ is the component $\{u^1, u^2, \dots u^K\}$

$$\hat{x} = \sum_{k=1}^{K} c_k w^k \Leftrightarrow x - \bar{x}$$

To minimize reconstruction error:

$$c_k = (x - \bar{x}) \cdot w^k$$

$K = 2$:



Gradient Descent?

PCA - Pokémon

- Inspired from: https://www.kaggle.com/strakul5/d/abcsds/pokemon/principal-component-analysis-of-pokemon-data
- 800 Pokemons, 6 features for each (HP, Atk, Def, Sp Atk, Sp Def, Speed)
- How many principle components? $\frac{\lambda_i}{\lambda_1+\lambda_2+\lambda_3+\lambda_4+\lambda_5+\lambda_6}$

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
ratio	0.45	0.18	0.13	0.12	0.07	0.04

Using 4 components is good enough