# 기계학습 (2022년도 2학기)

# Principal Component Analysis

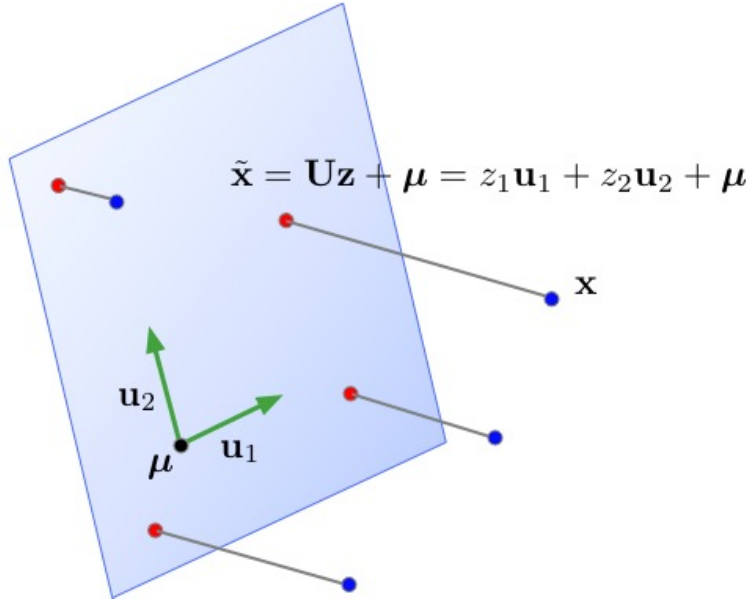
전북대학교 컴퓨터공학부

# Overview

- Today we'll cover the first unsupervised learning algorithm for this course: principal component analysis (PCA)
- Dimensionality reduction: map the data to a lower dimensional space
  - Save computation/memory
  - Reduce over fitting
  - Visualize in 2 dimensions
- PCA is a linear model, with a closed-form solution. It's useful for understanding lots of other algorithms.
  - Autoencoders
  - Matrix factorizations (next lecture)
- Today's lecture is very linear-algebra-heavy.
  - Especially orthogonal matrices and eigendecompositions.
  - Don't worry if you don't get it immediately | next few lectures won’t build on it

# Projection onto a subspace

- Set-up: given a dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\} \subset \mathbb{R}^D$
- Set $\mu$ to the mean of the data, $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}^{(i)}$
- Goal: find a K-dimensional subspace $\mathcal{S} \subset \mathbb{R}^D$ such that $\mathbf{x}^{(n)} - \mu$ is "well-represented" by its projection onto $\mathcal{S}$
- Recall: The projection of a point $\boldsymbol{x}$ onto $\mathcal{S}$ is the point in $\mathcal{S}$ closest to $\boldsymbol{x}$.



# Projection onto a subspace

기저: 어떤 벡터 공간의 기저(basis)는 그 벡터 공간을 선형생성하는 선형 독립인 벡터들. 즉, 벡터 공간의 임의의 벡터에게 선형결합으로서 유일한 표현을 부여하는 벡터들.

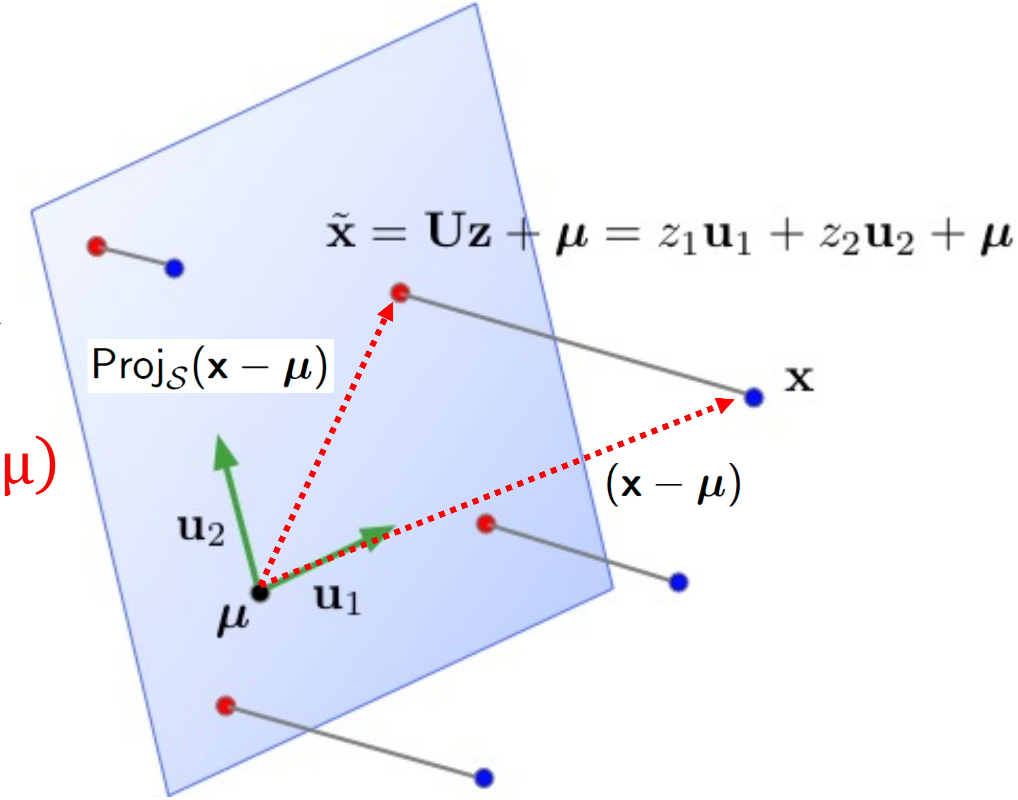
- Let $\{\mathbf{u}_k\}_{k=1}^{K}$ be an orthonormal basis of the subspace $\mathcal{S}$
- Approximate each data point x as:

$$\tilde{\mathbf{x}} = \boldsymbol{\mu} + \mathrm{Proj}_{\mathcal{S}}(\mathbf{x} - \boldsymbol{\mu})$$

$$= \boldsymbol{\mu} + \sum_{k=1}^{K} z_k \mathbf{u}_k$$

$$\mathrm{x} - \mu$$
$$= \tilde{\mathrm{x}} + \alpha - \mu$$
$$= \mathrm{z}_1\mathrm{u}_1 + \mathrm{z}_2\mathrm{u}_2 + \alpha - \mu$$

$$\mathrm{u}_1^{\mathrm{T}}(\mathrm{z}_1\mathrm{u}_1 + \mathrm{z}_2\mathrm{u}_2 + \alpha - \mu)$$
$$= \mathrm{u}_1^{\mathrm{T}}\mathrm{z}_1\mathrm{u}_1 = \mathrm{z}_1$$

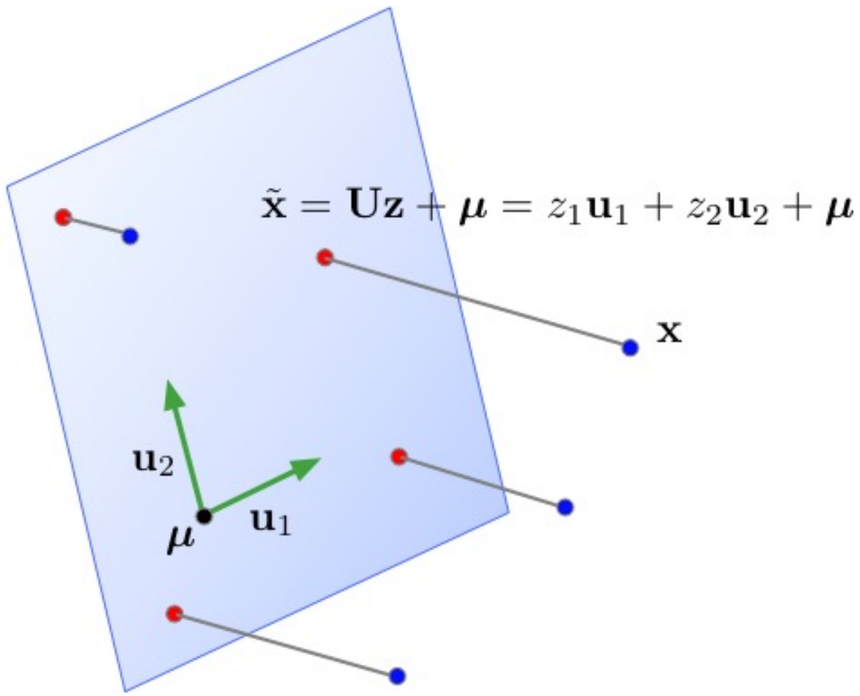



- From linear algebra: $z_k = \mathbf{u}_k^T(\mathbf{x} - \boldsymbol{\mu})$
- Let $\mathbf{U}$ be a matrix with columns $\{\mathbf{u}_k\}_{k=1}^{K}$
  then $\mathbf{z} = \mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu})$

예: 3차원 공간의 **x**를 2차원 공간($\mathcal{S}$)으로 projection 시키는 3x2 행렬

- Also: $\tilde{\mathbf{x}} = \boldsymbol{\mu} + \mathbf{U}\mathbf{z}$

# Projection onto a Subspace

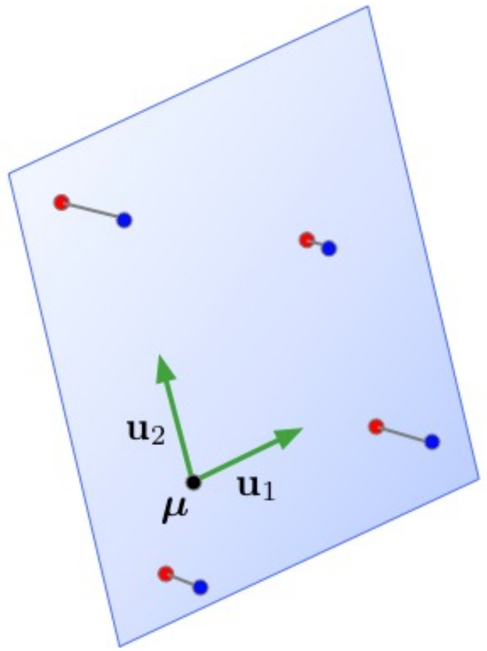



$$\mathbf{z} = \mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu})$$

- In machine learning, $\tilde{\mathbf{x}}$ is also called the reconstruction of $\mathbf{x}$.
- $\mathbf{z}$ is its representation, or code.

# Projection onto a Subspace

- If we have a *K*-dimensional subspace in a *D*-dimensional input space, then $\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{z} \in \mathbb{R}^K$.
- If the data points $\mathbf{x}$ all lie close to their reconstructions, then we can approximate distances, etc. in terms of these same operations on the code vectors $\mathbf{z}$.
  (좋은 approximation: 원래 data space에서의 data 분포를 잘 반영해야 함.
  즉 원래 space에서의 거리 추세가 subspace에서도 반영되어야 함.)
- If $K \ll D$, then it's much cheaper to work with $\mathbf{z}$ than $\mathbf{x}$.
- A mapping to a space that's easier to manipulate or visualize is called a representation (표현, 표상), and learning such a mapping is representation learning.
- Mapping data to a low-dimensional space is called dimensionality reduction.



# Learning a Subspace

- **How to choose a good subspace $\mathcal{S}$?**
  - **Need to choose $D \times K$ matrix U with orthonormal columns.**
    (즉, 행렬 **U**의 각 column이 subspace S의 basis에 해당)
- Two criteria:
  - Minimize the reconstruction error

$$\min \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}^{(i)} - \tilde{\mathbf{x}}^{(i)}\|^2$$

  - Maximize the variance of the code vectors
    → code vector가 data 분포를 최대한 잘 반영해야 함, 즉 중요한 data 분포 특성을 나타낼 수 있어야 함

$$\begin{aligned}\max \sum_j \mathrm{Var}(z_j) &= \frac{1}{N} \sum_j \sum_i (z_j^{(i)} - \bar{z}_j)^2 \\ &= \frac{1}{N} \sum_i \|\mathbf{z}^{(i)} - \bar{\mathbf{z}}\|^2 \\ &= \frac{1}{N} \sum_i \|\mathbf{z}^{(i)}\|^2\end{aligned}$$

Exercise: show $\bar{\mathbf{z}} = 0$

(직접 해볼 것!)

  - Note: here, $\bar{\mathbf{z}}$ denotes the mean, not a derivative.

# Learning a Subspace

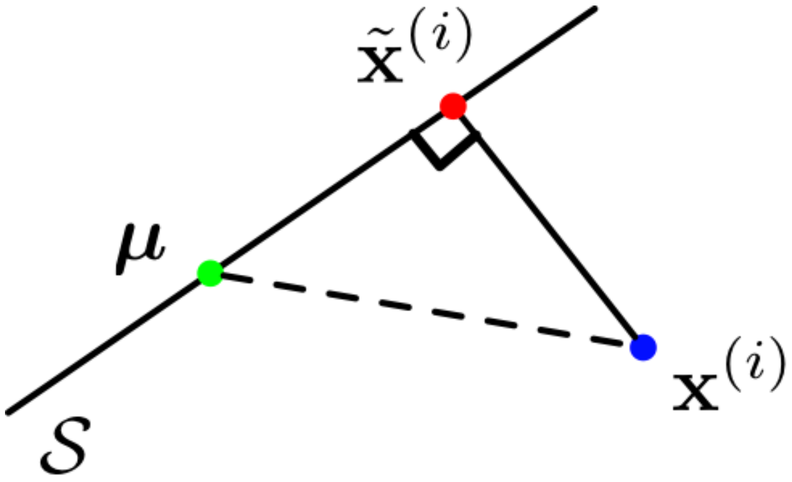
- These two criteria are equivalent! I.e., we'll show

$$\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)} - \tilde{\mathbf{x}}^{(i)}\|^2 = \text{const} - \frac{1}{N}\sum_{i}\|\mathbf{z}^{(i)}\|^2$$

- Observation: by unitarity,

$$\tilde{\mathbf{x}} = \boldsymbol{\mu} + \mathbf{U}\mathbf{z} \qquad \|\tilde{\mathbf{x}}^{(i)} - \boldsymbol{\mu}\| = \|\mathbf{U}\mathbf{z}^{(i)}\| = \|\mathbf{z}^{(i)}\|$$

- By the Pythagorean Theorem,




$$\underbrace{\frac{1}{N}\sum_{i=1}^{N}\|\tilde{\mathbf{x}}^{(i)} - \boldsymbol{\mu}\|^2}_{\text{projected variance}} + \underbrace{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)} - \tilde{\mathbf{x}}^{(i)}\|^2}_{\text{reconstruction error}}$$

$$= \underbrace{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)} - \boldsymbol{\mu}\|^2}_{\text{constant}}$$

# Principal Component Analysis

Choosing a subspace to maximize the projected variance, or minimize the reconstruction error, is called principal component analysis (PCA).

Recall:

- Spectral Decomposition: a symmetric matrix **A** has a full set of eigenvectors, which can be chosen to be orthogonal. This gives a decomposition

$$\mathbf{A} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top,$$

Eigenvector (고유벡터): 선형변환 A에 의한 변환 결과가 자기 자신의 상수배가 되는 0이 아닌 벡터
Eigenvalue (고유값): 이 상수배 값, $\mathbf{AQ} = \mathbf{Q\Lambda} \rightarrow \mathbf{A} = \mathbf{Q\Lambda Q^{-1}} = \mathbf{Q\Lambda Q^T}$

where **Q** is orthogonal and $\boldsymbol{\Lambda}$ is diagonal. The columns of **Q** are eigenvectors, and the diagonal entries $\lambda_j$ of $\boldsymbol{\Lambda}$ are the corresponding eigenvalues.

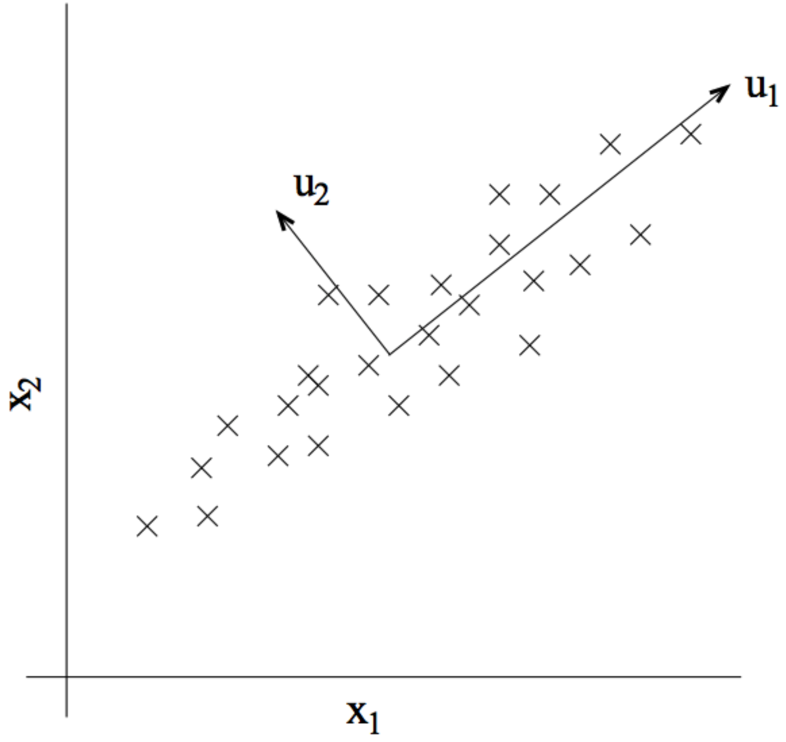
- I.e., symmetric matrices are diagonal in some basis (**Q**).
- A symmetric matrix **A** is positive semidefinite iff each $\lambda_j \geq 0$.
  (선형변환에서 비음수 eigenvalue들은 새로운 coordinates들을 표준 coordinates들 각성분의 scale로 생각할 수 있어, 변형을 이들에 대한 조합으로 쉽게 이해하고 해석할 수 있음)

# Principal Component Analysis

- Consider the empirical covariance matrix (공분산 행렬):

$$\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top$$

- Recall: Covariance matrices are symmetric and positive semidefinite.
(대칭 행렬은 항상 직교행렬 고윳값 대각화 가능!!)
- The optimal PCA subspace is spanned by the top $K$ eigenvectors of $\boldsymbol{\Sigma}$.
  - More precisely, choose the first $K$ of any orthonormal eigenbasis for $\boldsymbol{\Sigma}$.
  - The general case is tricky, but we'll show this for $K = 1$.
- These eigenvectors are called principal components, analogous to the principal axes of an ellipse.



# Deriving PCA

- For $K = 1$, we are fitting a unit vector $\mathbf{u}$, and the code is a scalar $z = \mathbf{u}^\top(\mathbf{x} - \boldsymbol{\mu})$

$$
\begin{aligned}
\frac{1}{N}\sum_i [z^{(i)}]^2 &= \frac{1}{N}\sum_i (\mathbf{u}^\top(\mathbf{x}^{(i)} - \boldsymbol{\mu}))^2 \\
&= \frac{1}{N}\sum_{i=1}^{N} \mathbf{u}^\top(\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top \mathbf{u} \\
&= \mathbf{u}^\top \left[\frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}^{(i)} - \boldsymbol{\mu})(\mathbf{x}^{(i)} - \boldsymbol{\mu})^\top\right] \mathbf{u} \\
&= \mathbf{u}^\top \boldsymbol{\Sigma} \mathbf{u} \\
&= \mathbf{u}^\top \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top \mathbf{u} \qquad \text{Spectral Decomposition} \\
&= \mathbf{a}^\top \boldsymbol{\Lambda} \mathbf{a} \qquad \text{for } \mathbf{a} = \mathbf{Q}^\top \mathbf{u} \\
&= \sum_{j=1}^{D} \lambda_j a_j^2
\end{aligned}
$$

# Deriving PCA

(7페이지 참고)

$$\max \sum_j \mathrm{Var}(z_j) = \frac{1}{N}\sum_j \sum_i (z_j^{(i)} - \bar{z}_j)^2$$

- Maximize $\mathbf{a}^\top \mathbf{\Lambda} \mathbf{a} = \sum_{j=1}^{D} \lambda_j a_j^2$ for $\mathbf{a} = \mathbf{Q}^\top \mathbf{u}$.
  - This is a change-of-basis to the eigenbasis of $\mathbf{\Sigma}$.
- Assume the $\lambda_i$ are in sorted order. For simplicity, assume they are all distinct.
- Observation: since **u** is a unit vector, then by unitarity, **a** is also a unit vector. I.e., $\sum_j a_j^2 = 1$ (직교행렬들의 곱도 직교행렬)
- By inspection, set $a_1 = \pm 1$ and $a_j = 0$ for $j \neq 1$
- Hence, $\mathbf{u} \models \mathbf{Q}\mathbf{a} = \mathbf{q}_1$ (the top eigenvector)

- A similar argument shows that the k-th principal component is the k-th eigenvector of $\mathbf{\Sigma}$. If you're interested, look up the Courant-Fischer Theorem.

# Decorrelation

- Interesting fact: the dimensions of **z** are decorrelated. For now, let Cov denote the empirical covariance.

$$
\begin{aligned}
\mathrm{Cov}(\mathbf{z}) &= \mathrm{Cov}(\mathbf{U}^\top(\mathbf{x}-\boldsymbol{\mu})) \\
&= \mathbf{U}^\top \mathrm{Cov}(\mathbf{x})\mathbf{U} \\
&= \mathbf{U}^\top \boldsymbol{\Sigma}\mathbf{U} \\
&= \mathbf{U}^\top \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top\mathbf{U} \\
&= \begin{pmatrix}\mathbf{I} & \mathbf{0}\end{pmatrix} \boldsymbol{\Lambda} \begin{pmatrix}\mathbf{I} \\ \mathbf{0}\end{pmatrix} && \text{by orthogonality} \\
&= \text{top left } K \times K \text{ block of } \boldsymbol{\Lambda}
\end{aligned}
$$

$[\underbrace{1,\ldots 1}_{K},0,0,\ldots 0]$

- If the covariance matrix is diagonal, this means the features are uncorrelated.
- This is why PCA was originally invented (in 1901!).

# Recap

- Dimensionality reduction aims to find a low-dimensional representation of the data.
- PCA projects the data onto a subspace which maximizes the projected variance, or equivalently, minimizes the reconstruction error.
- The optimal subspace is given by the top eigenvectors of the empirical covariance matrix.
- PCA gives a set of decorrelated features.

# Applying PCA to faces

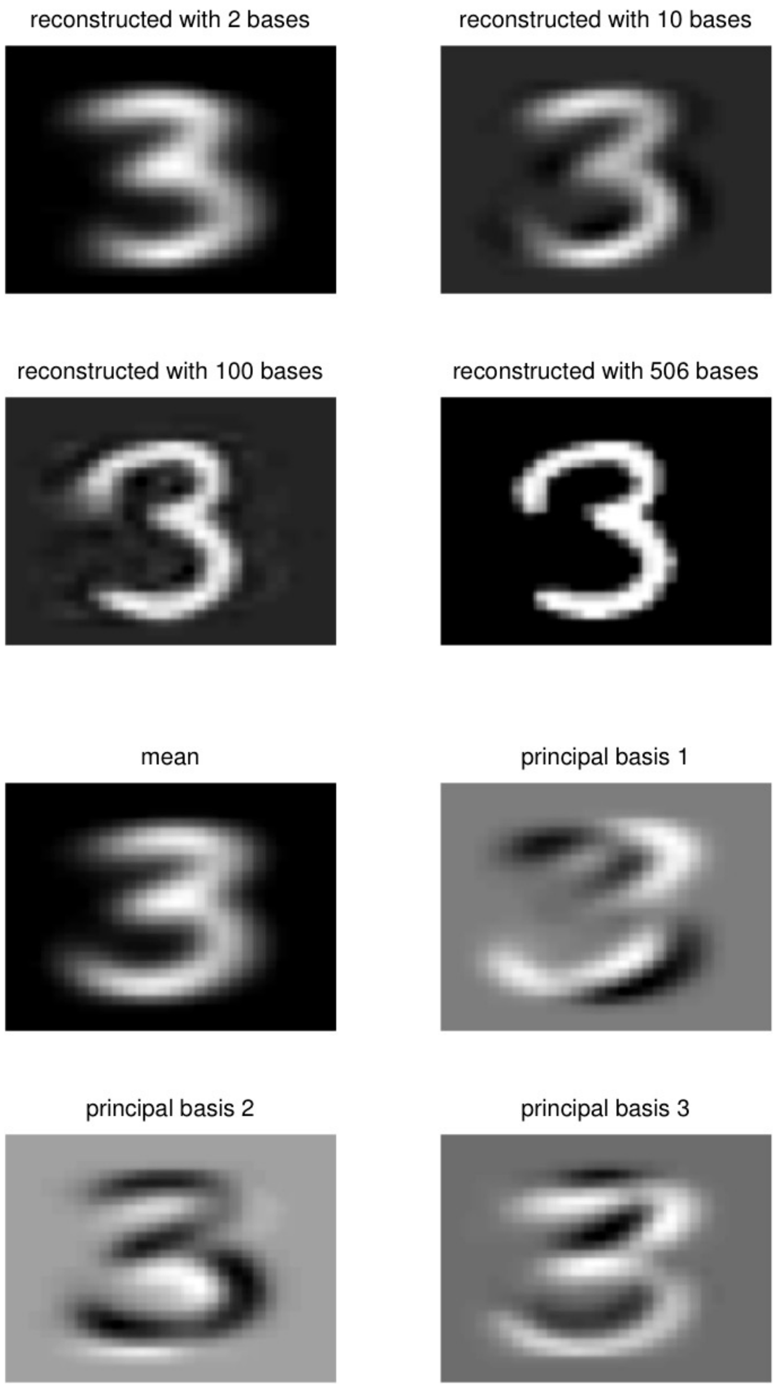
- Consider running PCA on 2429 19x19 grayscale images (CBCL data)
- Can get good reconstructions with only 3 components

- PCA for pre-processing: can apply classifier to latent representation
  - For face recognition PCA with 3 components obtains 79% accuracy on face/non-face discrimination on test data vs. 76.8% for a Gaussian mixture model (GMM) with 84 states. (We'll cover GMMs later in the course.)
- Can also be good for visualization

# Applying PCA to faces: Learned basis

- Principal components of face images (“eigenfaces")

# Applying PCA to digits

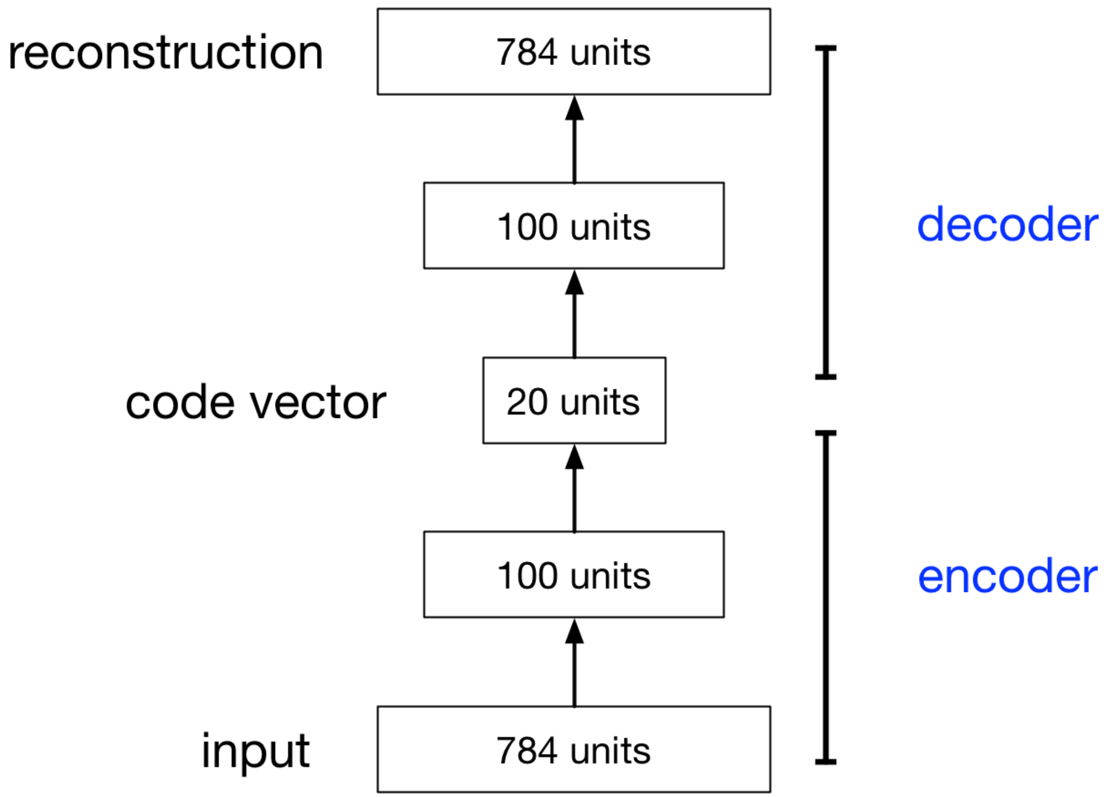


# Next

- Two more interpretations of PCA, which have interesting generalizations.
  - Autoencoders
  - **Matrix factorization (later lecture)**

# Autoencoders

- An autoencoder is a feed-forward neural net whose job it is to take an input **x** and predict **x**.
- To make this non-trivial, we need to add a bottleneck layer whose dimension is much smaller than the input.



# Linear Autoencoders

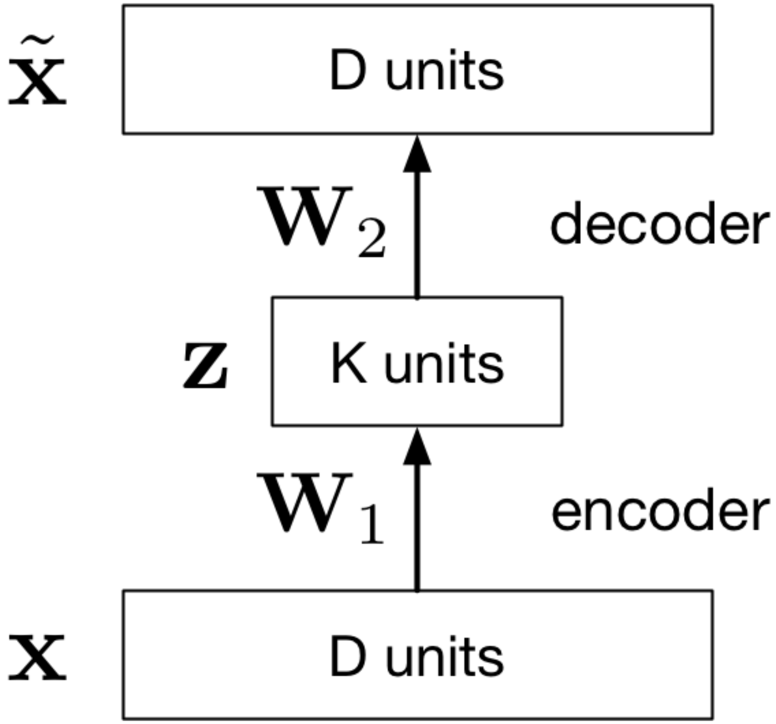
Why autoencoders?

- Map high-dimensional data to two dimensions for visualization
- Learn abstract features in an unsupervised way so you can apply them to a supervised task
  - Unlabled data can be much more plentiful than labeled data

# Linear Autoencoders

- The simplest kind of autoencoder has one hidden layer, linear activations, and squared error loss.

$$\mathcal{L}(\mathbf{x}, \tilde{\mathbf{x}}) = \|\mathbf{x} - \tilde{\mathbf{x}}\|^2$$



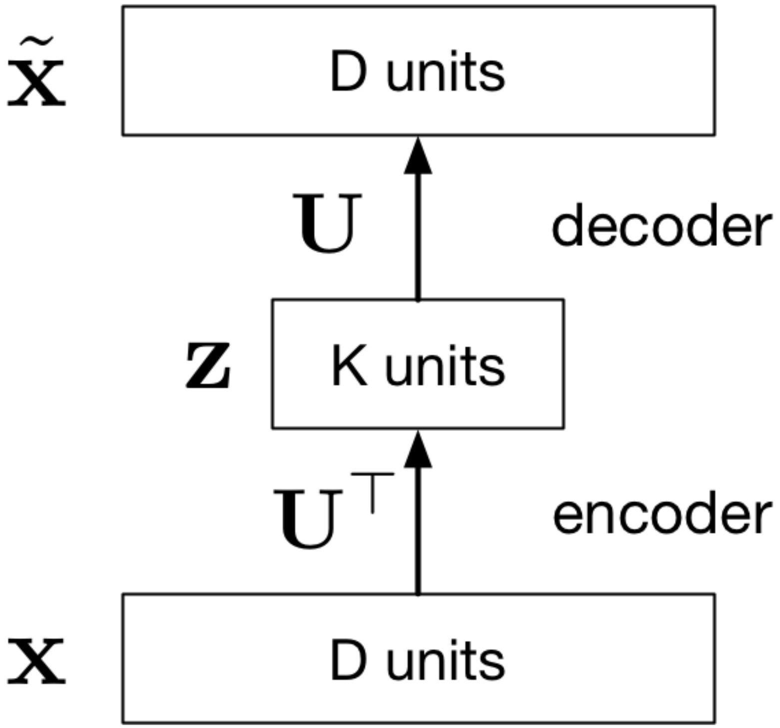

- This network computes $\tilde{\mathbf{x}} = \mathbf{W}_2\mathbf{W}_1\mathbf{x}$, which is a linear function.

- If $K \geq D$, we can choose $\mathbf{W}_2$ and $\mathbf{W}_1$ such that $\mathbf{W}_2\mathbf{W}_1$ is the identity matrix. This isn't very interesting.

- But suppose $K < D$ :
  - $\mathbf{W}_1$ maps $\mathbf{x}$ to a $K$-dimensional space, so it's doing dimensionality reduction.

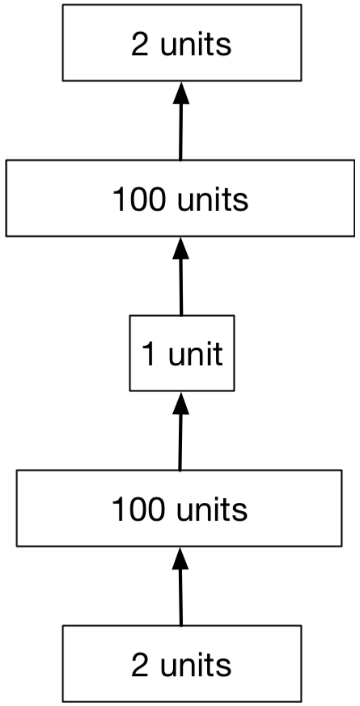
## Linear Autoencoders

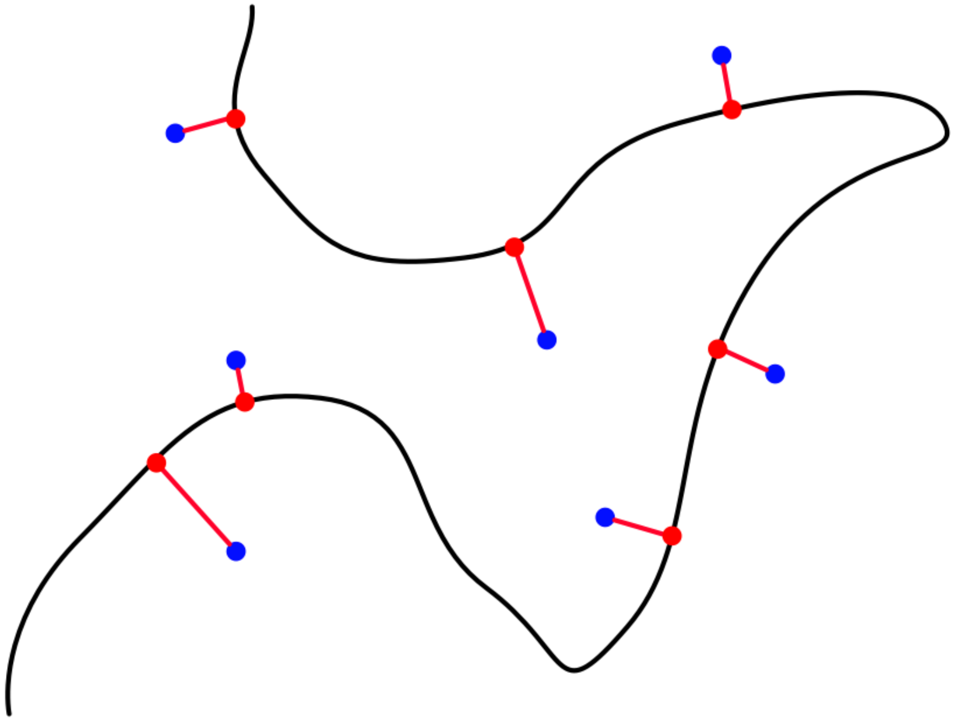
- Observe that the output of the autoencoder must lie in a $K$-dimensional subspace spanned by the columns of $\mathbf{W}_2$.
- We saw that the best possible $K$-dimensional subspace in terms of reconstruction error is the PCA subspace.
- The autoencoder can achieve this by setting $\mathbf{W}_1 = \mathbf{U}^T$ and $\mathbf{W}_2 = \mathbf{U}$.
- Therefore, the optimal weights for a linear autoencoder are just the principal components!



# Nonlinear Autoencoders

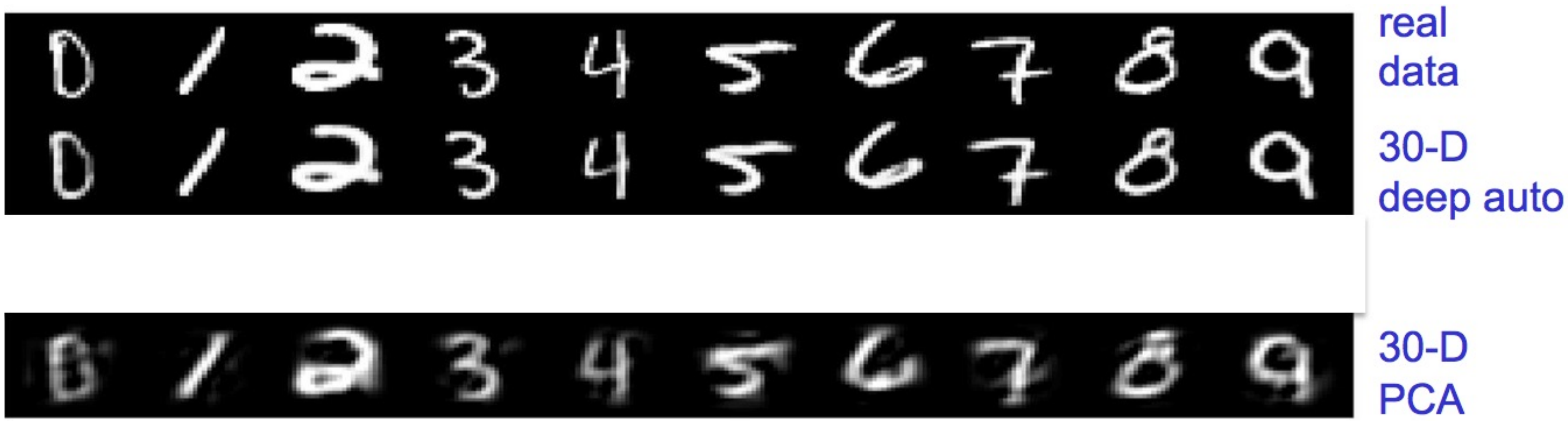
- Deep nonlinear autoencoders learn to project the data, not onto a subspace, but onto a nonlinear manifold
- This manifold is the image of the decoder
- This is a kind of nonlinear dimensionality reduction



# Nonlinear Autoencoders

- Nonlinear autoencoders can learn more powerful codes for a given dimensionality, compared with linear autoencoders (PCA)



# Nonlinear Autoencoders

- Here's a 2-dimensional autoencoder representation of newsgroup articles.
- They're color-coded by topic, but the algorithm wasn't given the labels.