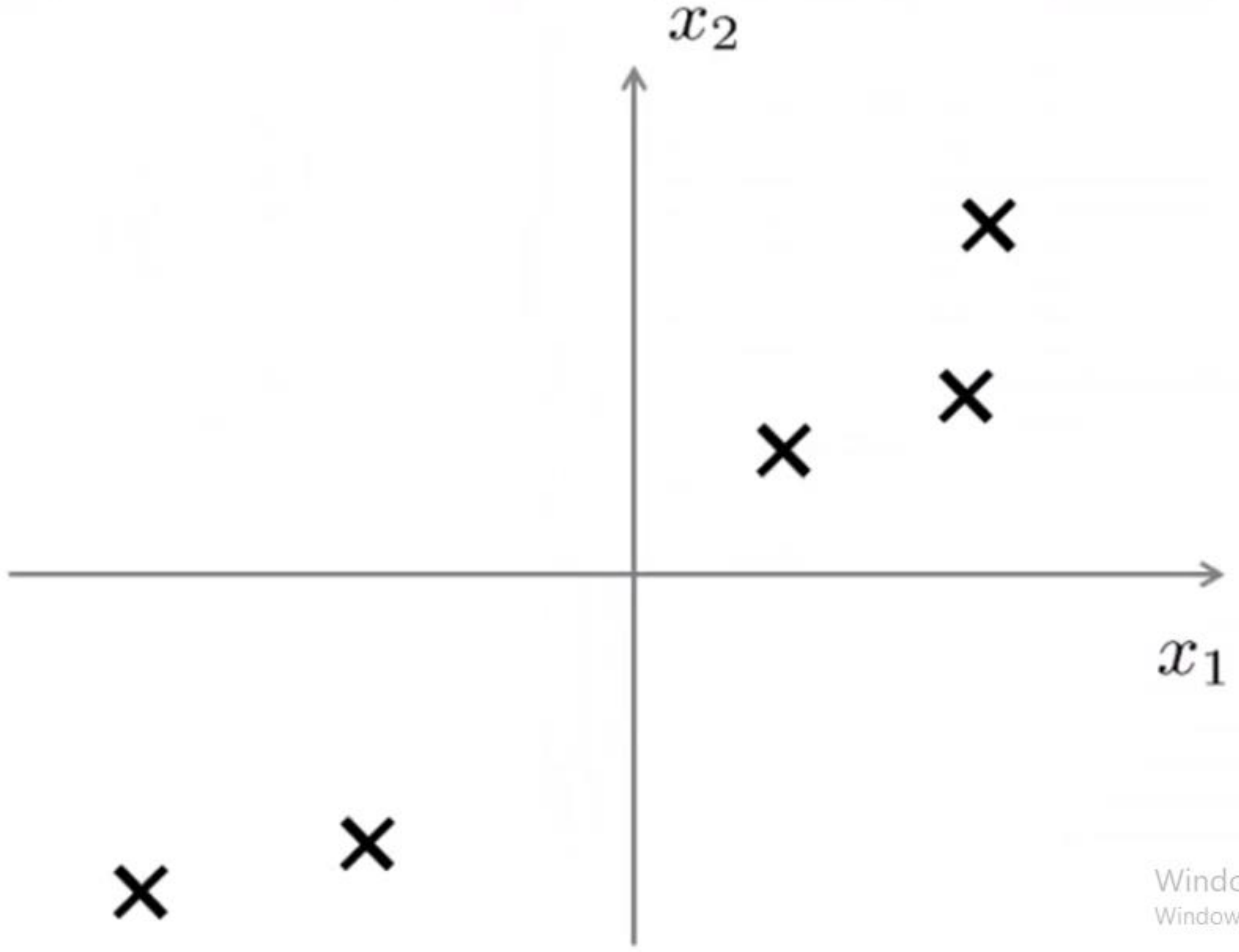
Problem Formulation

Principal Component Analysis

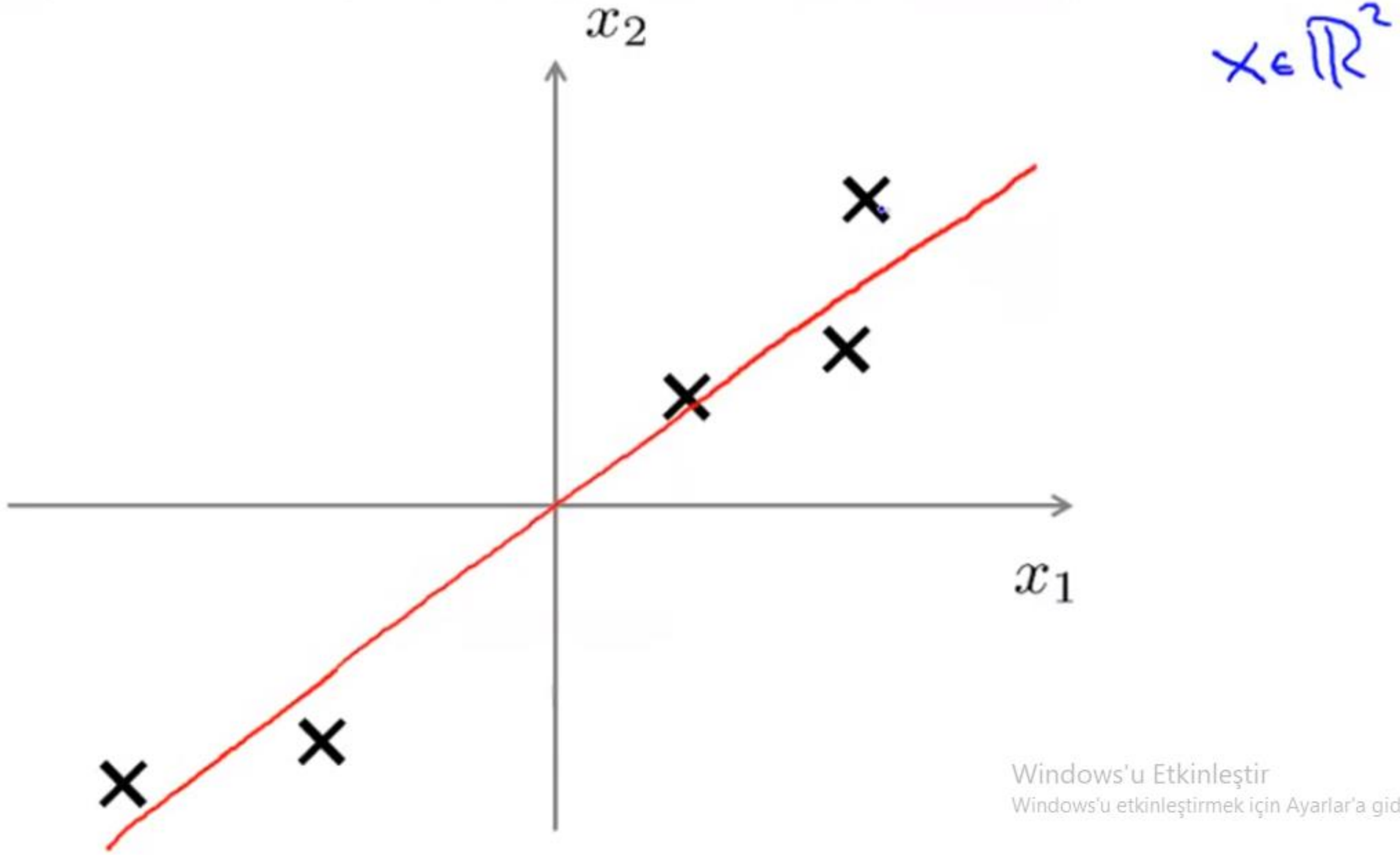
Unsupervised Learning

Principal Component Analysis (PCA) problem formulation

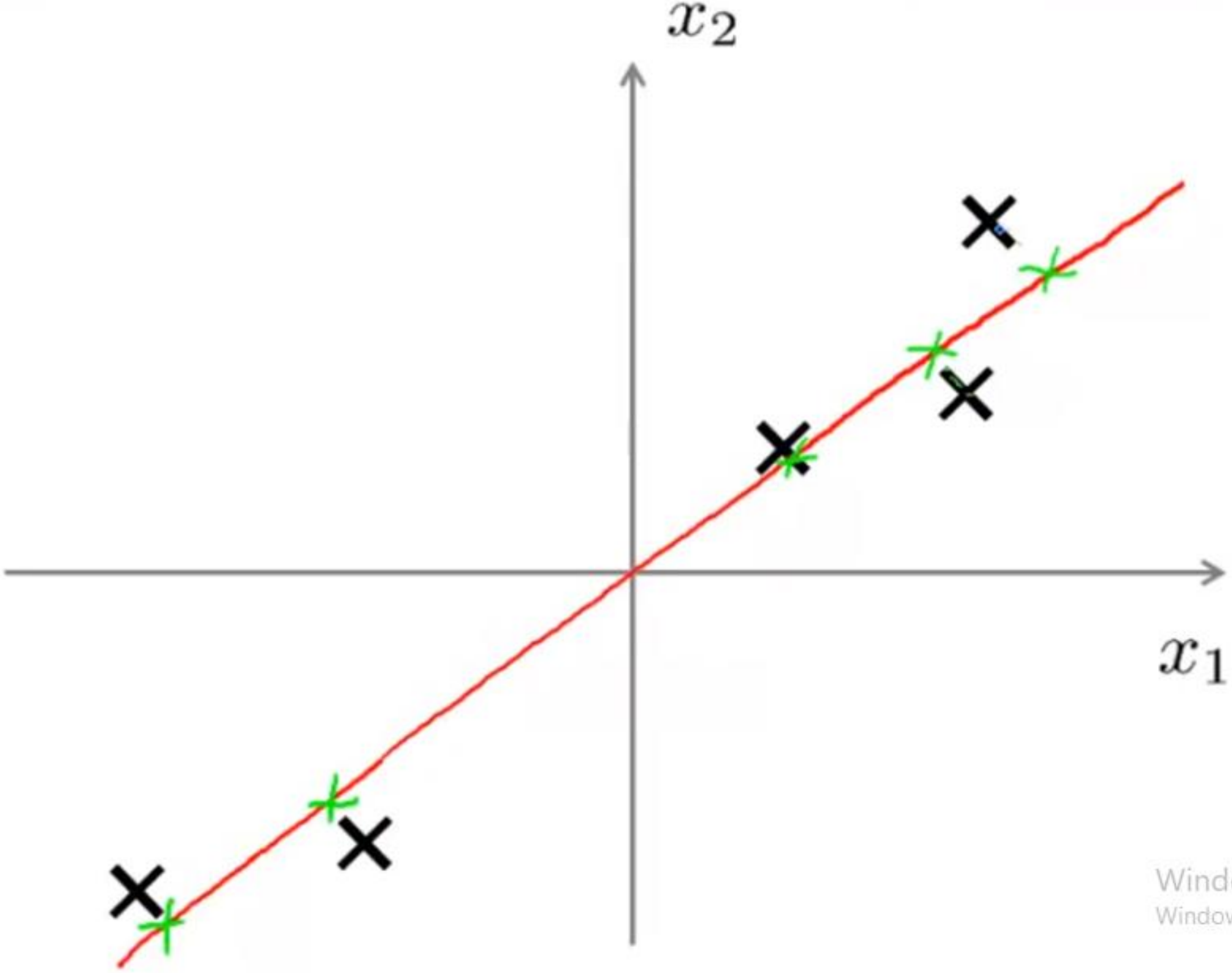
$x \in \mathbb{R}^2$



Principal Component Analysis (PCA) problem formulation

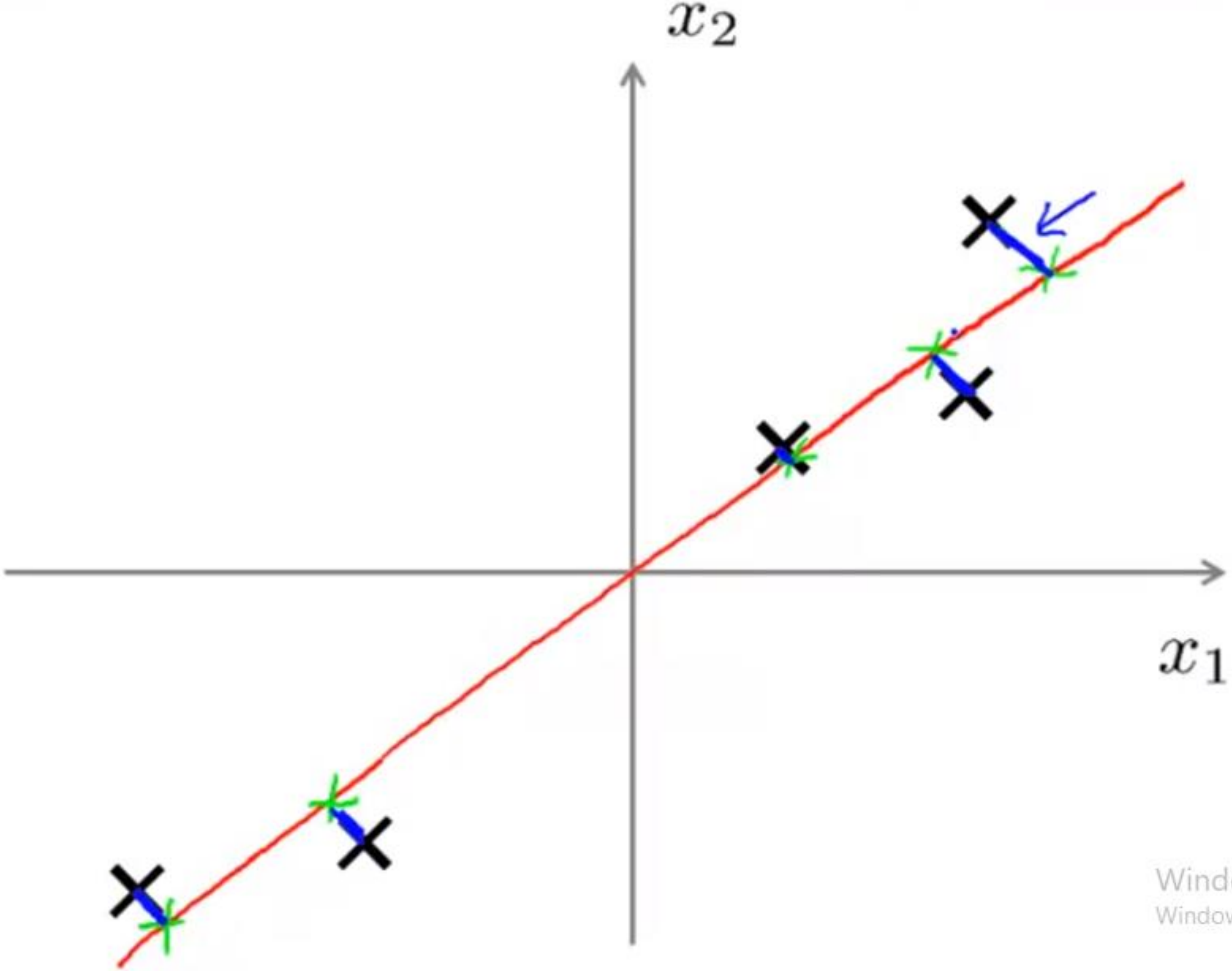


Principal Component Analysis (PCA) problem formulation



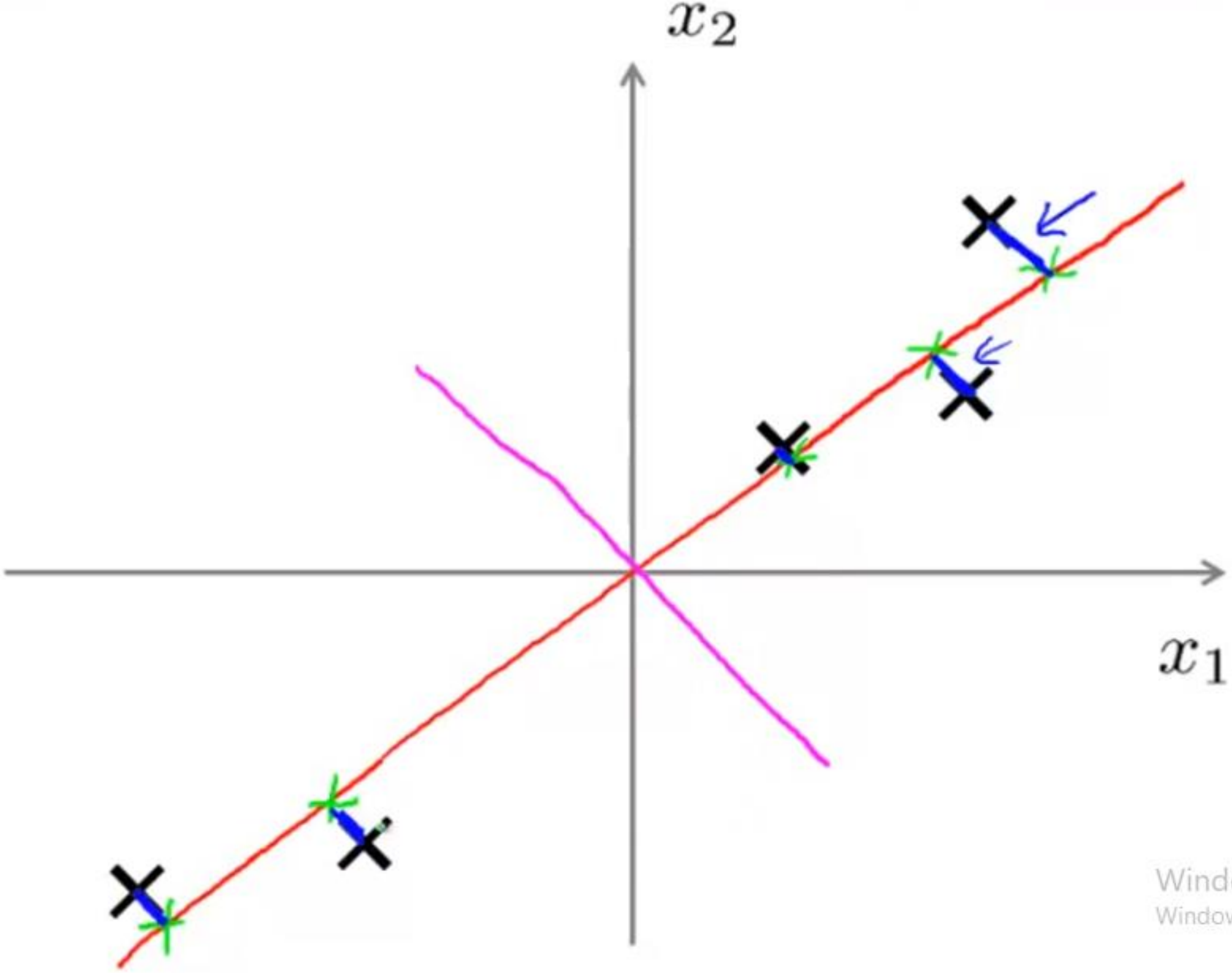
$x \in \mathbb{R}^2$

Principal Component Analysis (PCA) problem formulation



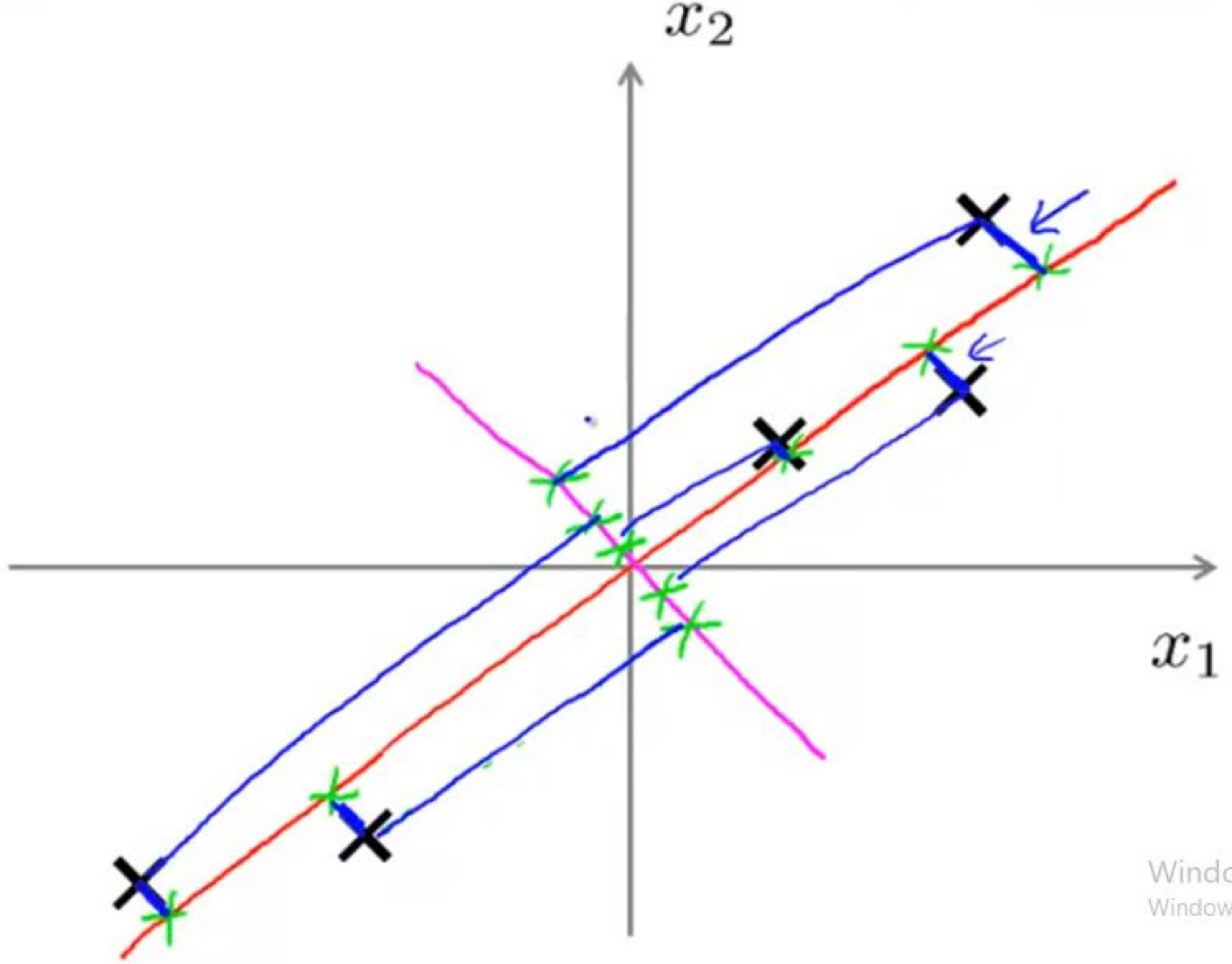
$x \in \mathbb{R}^2$

Principal Component Analysis (PCA) problem formulation



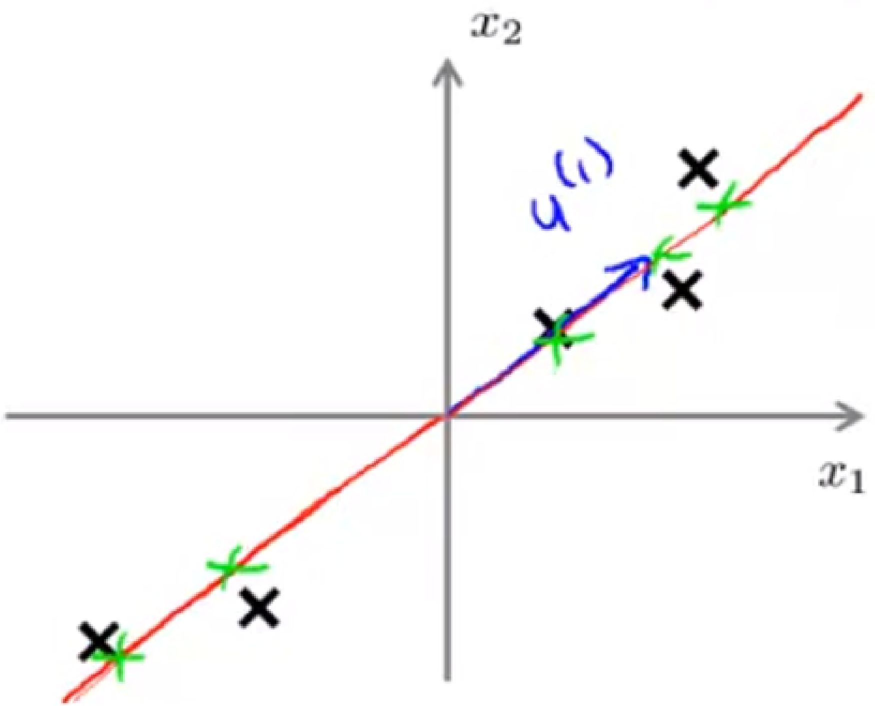
$x \in \mathbb{R}^2$

Principal Component Analysis (PCA) problem formulation



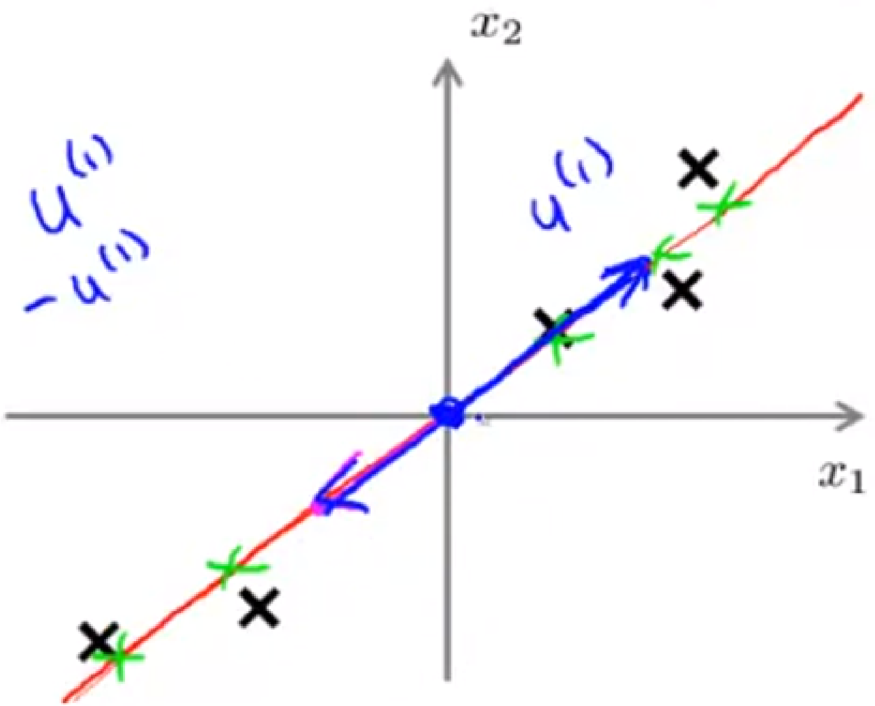
$x \in \mathbb{R}^2$

Principal Component Analysis (PCA) problem formulation



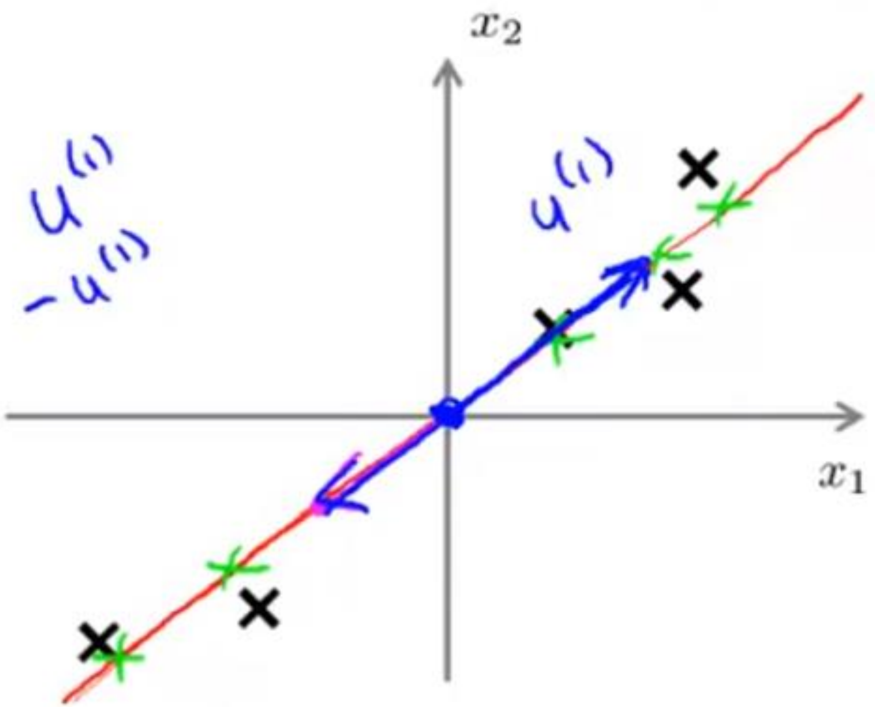
Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Principal Component Analysis (PCA) problem formulation



Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Principal Component Analysis (PCA) problem formulation

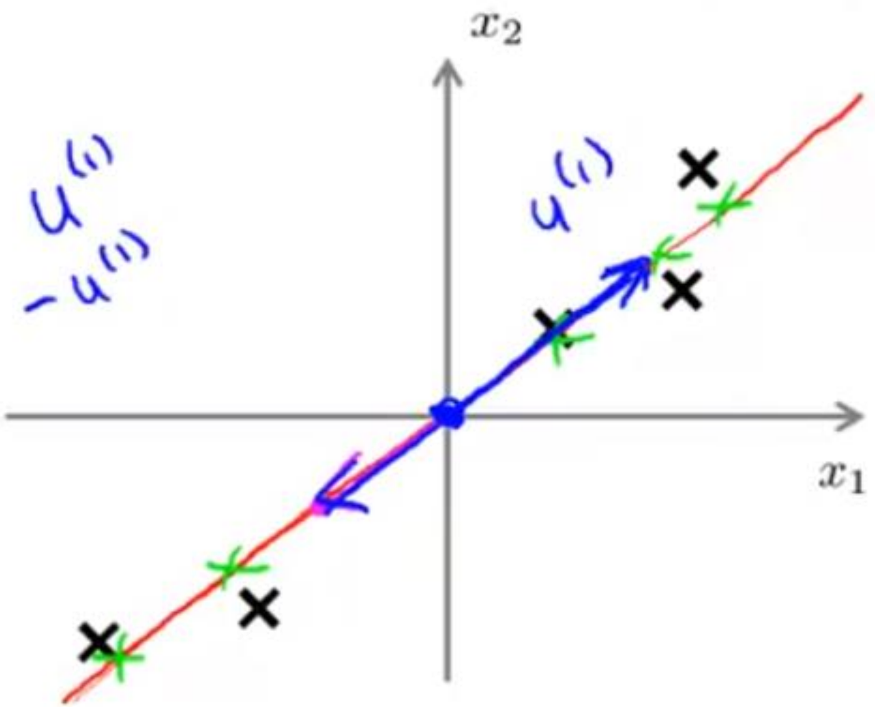


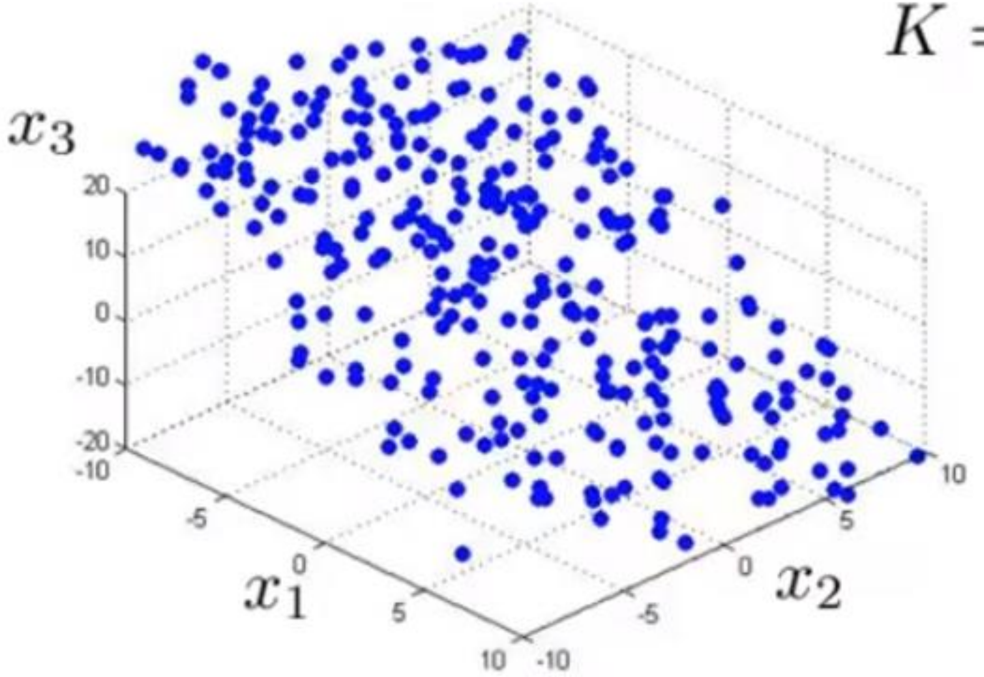
Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Reduce from n-dimension to k-dimension: Find k vectors $u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

Principal Component Analysis (PCA) problem formulation

$3D \rightarrow 2D$
$K = 2$



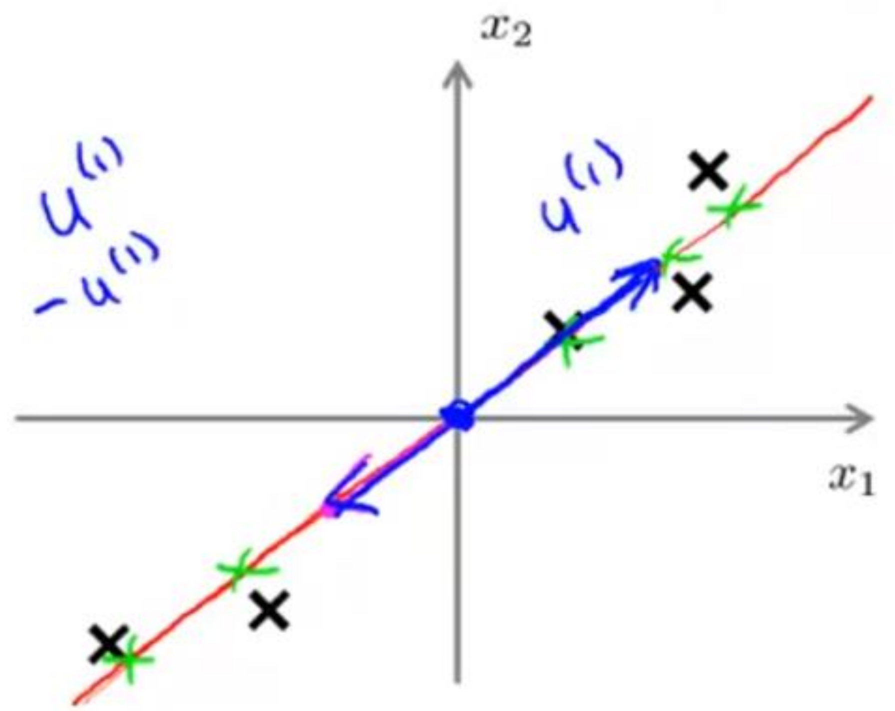


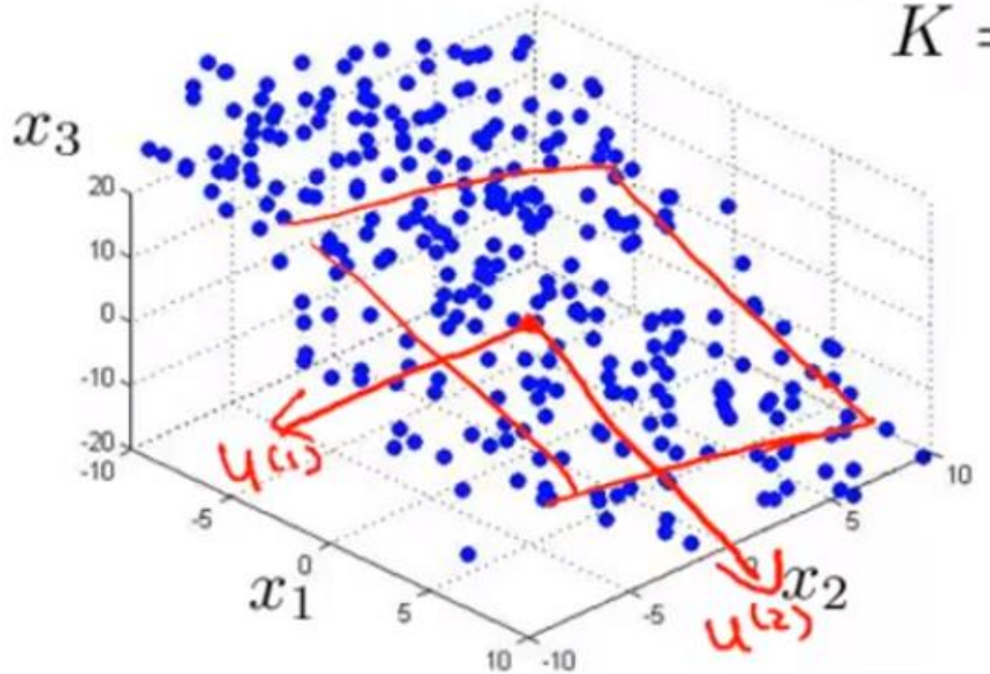
Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Reduce from n-dimension to k-dimension: Find k vectors $u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

Principal Component Analysis (PCA) problem formulation

$3D \to 2D$
$K = 2$

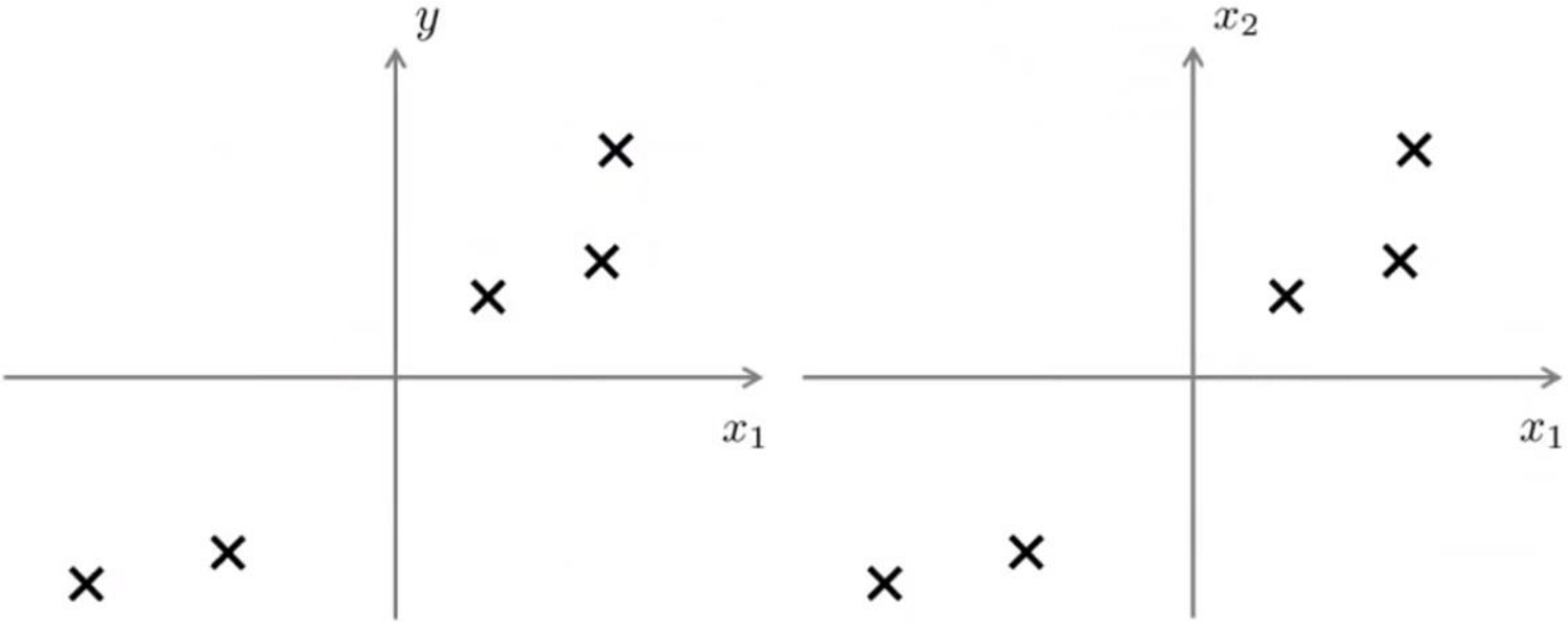




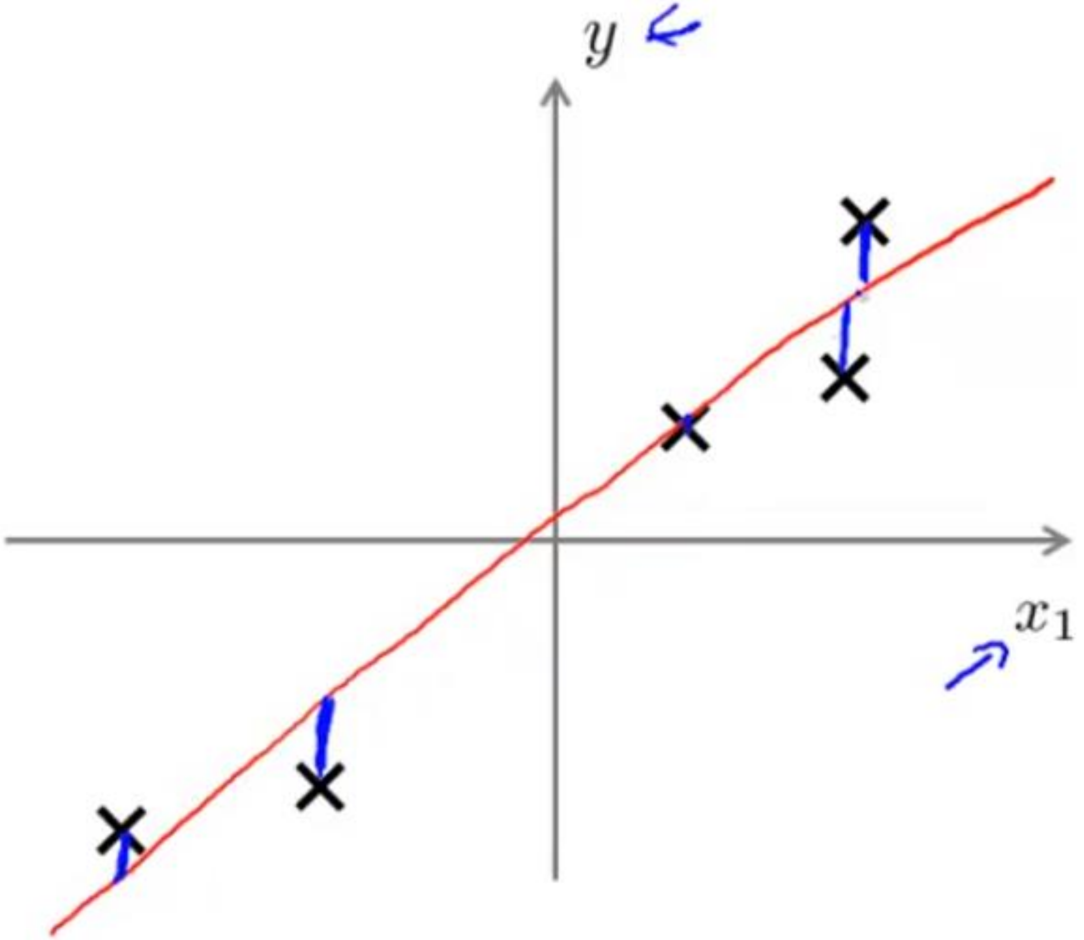
Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

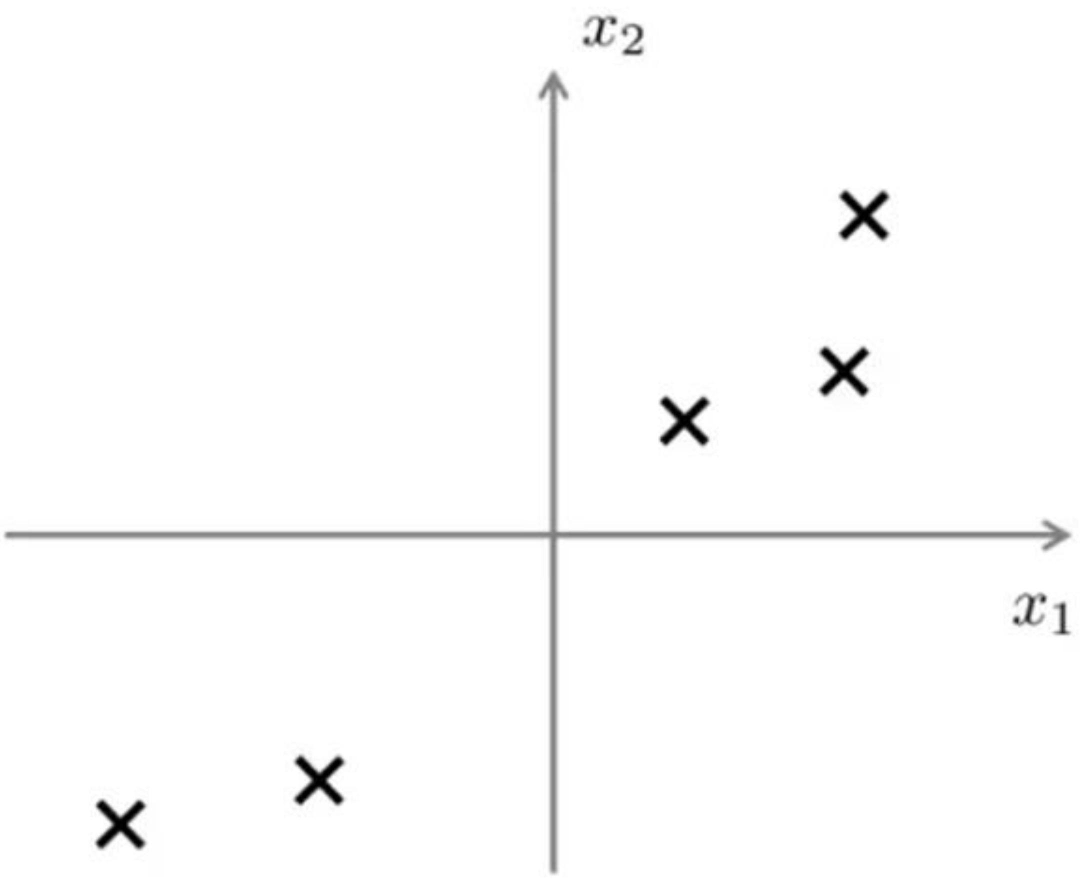
Reduce from n-dimension to k-dimension: Find k vectors $u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

PCA is not linear regression

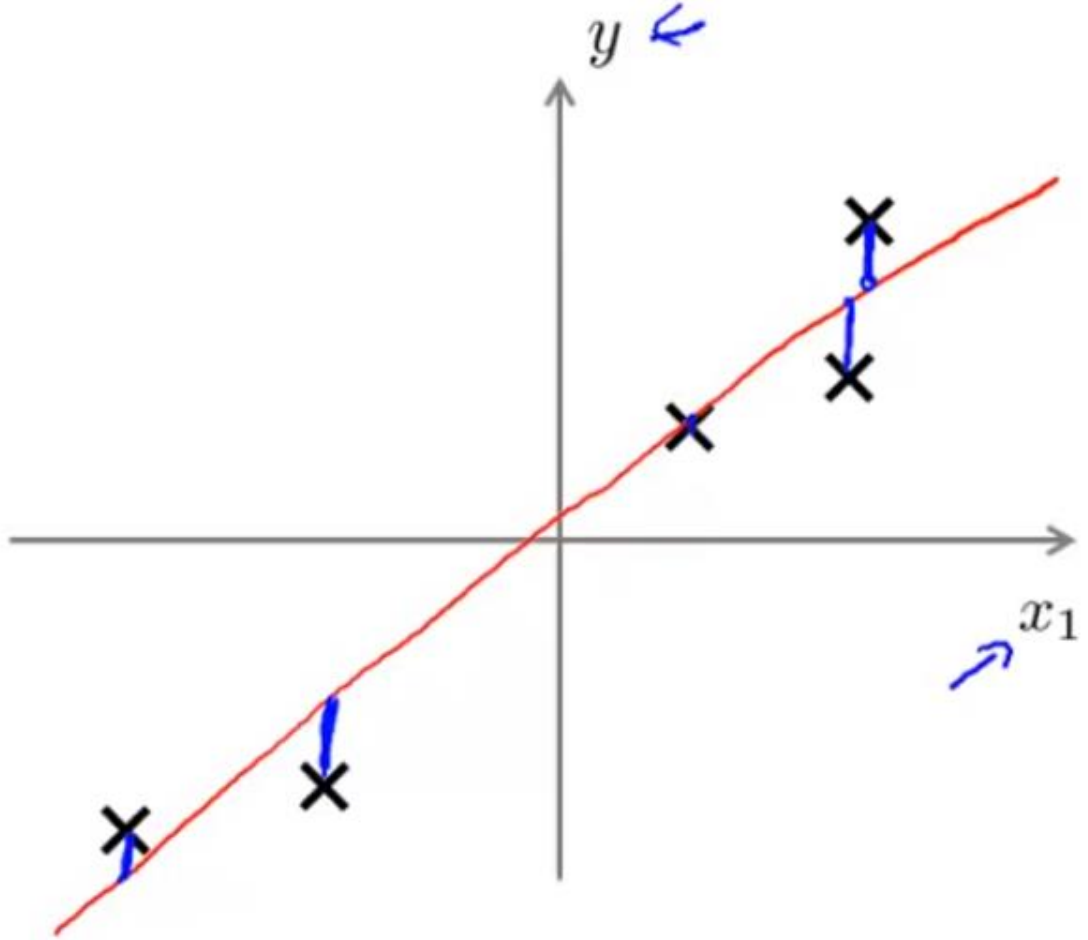


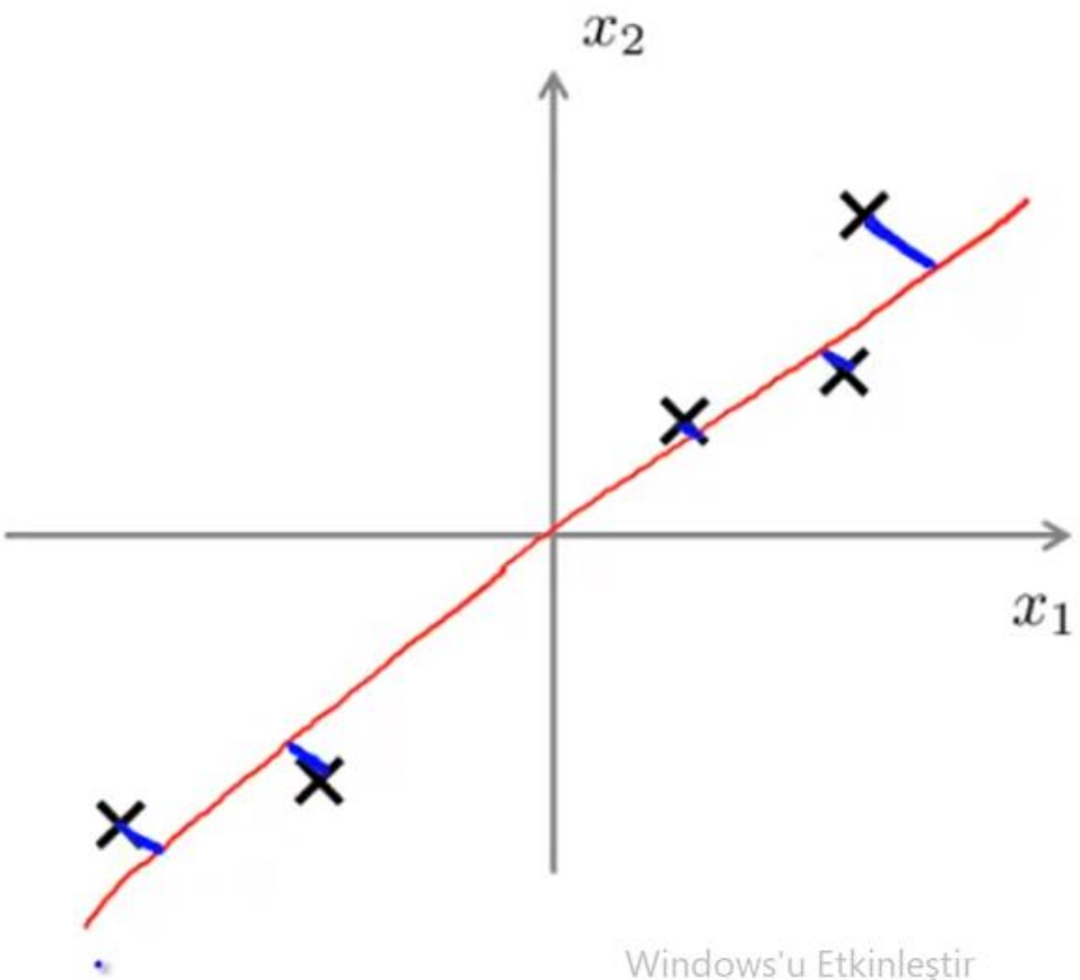
PCA is not linear regression





PCA is not linear regression





Algorithm

Principal Component Analysis

Unsupervised Learning

Data preprocessing

Training set: $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ $\leftarrow$

Preprocessing (feature scaling/mean normalization):

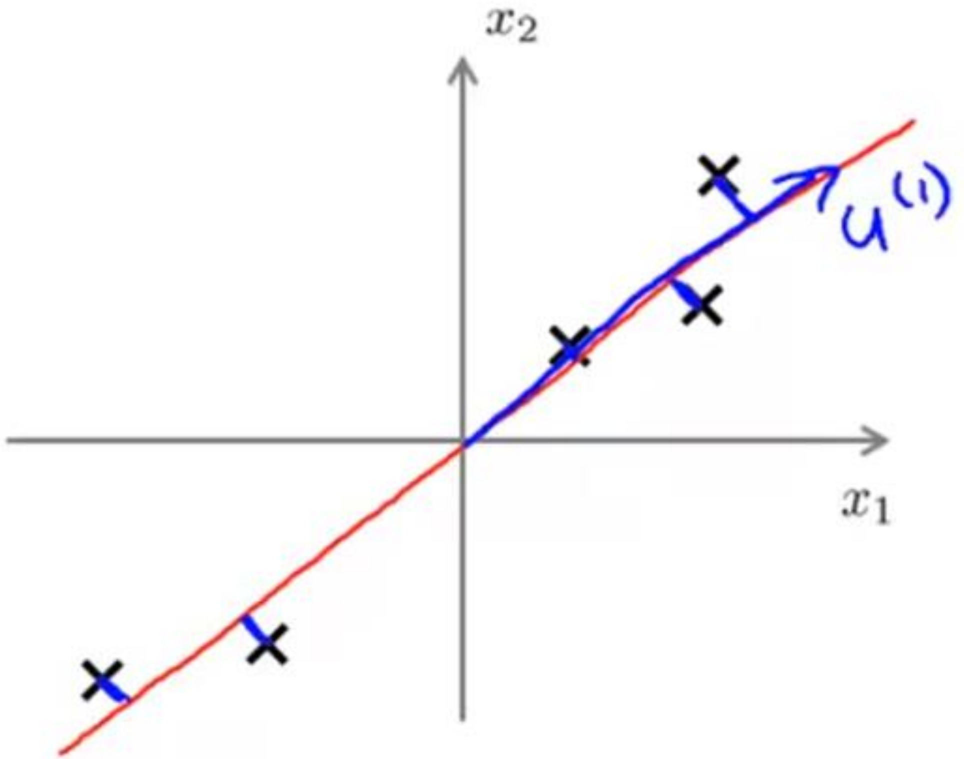
$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$$

Replace each $x_j^{(i)}$ with $x_j - \mu_j$.

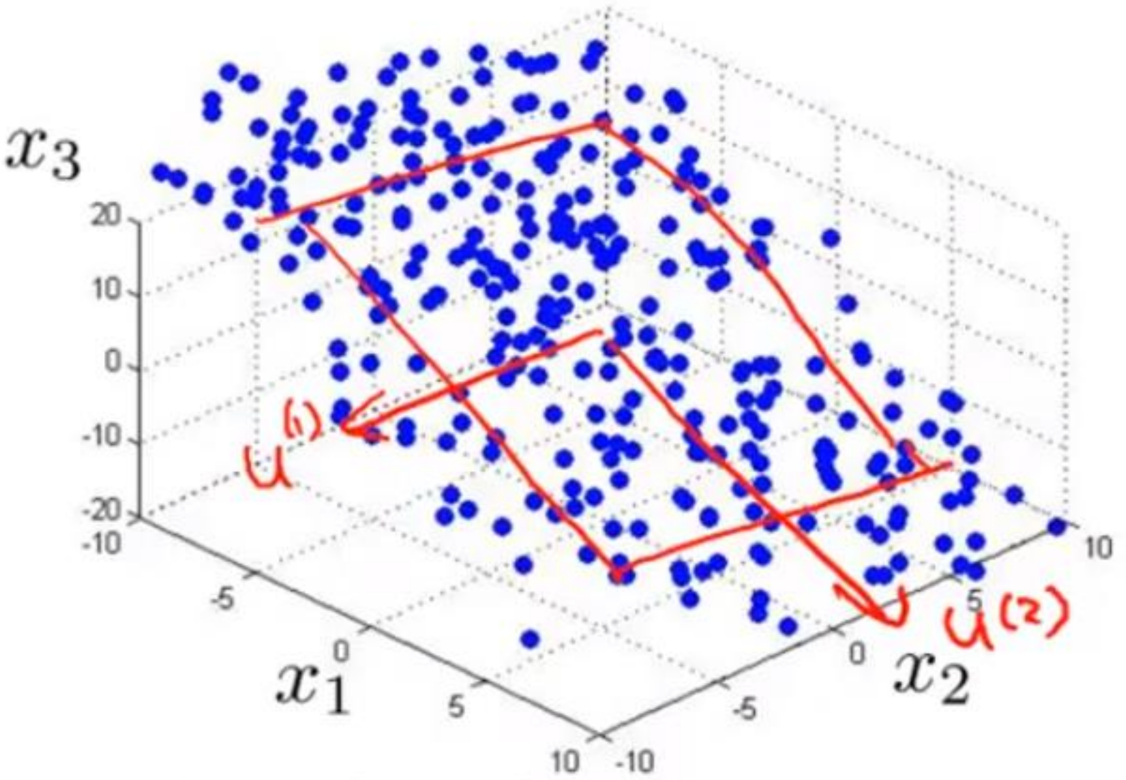
If different features on different scales (e.g., $x_1 =$ size of house, $x_2 =$ number of bedrooms), scale features to have comparable range of values.

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{s_j}$$

Principal Component Analysis (PCA) algorithm

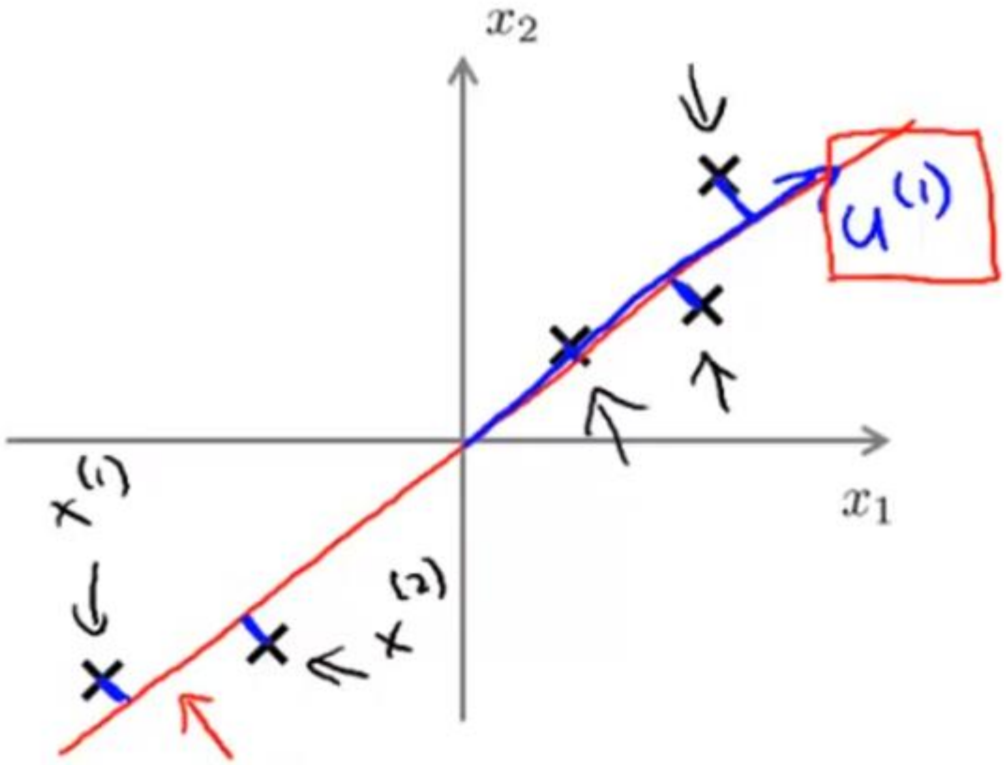


Reduce data from 2D to 1D



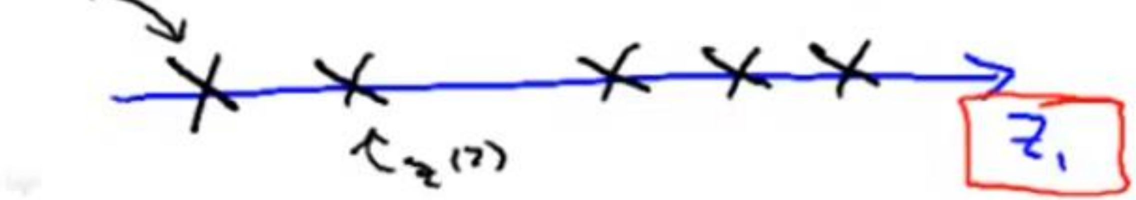
Reduce data from 3D to 2D

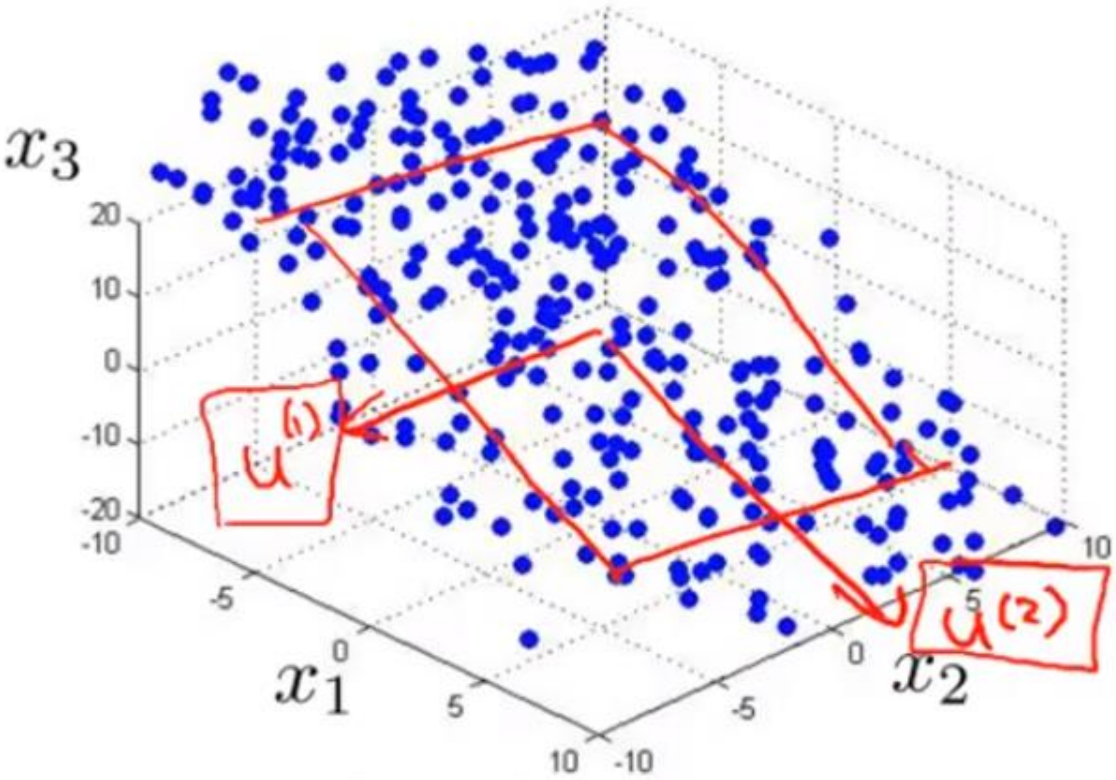
Principal Component Analysis (PCA) algorithm



Reduce data from 2D to 1D

$x^{(i)} \in \mathbb{R}^2 \rightarrow z^{(i)} \in \mathbb{R}$





Reduce data from 3D to 2D

$x^{(i)} \in \mathbb{R}^3 \rightarrow z^{(i)} \in \mathbb{R}^2$

$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$.

Principal Component Analysis (PCA) algorithm

Reduce data from n-dimensions to k-dimensions

Compute "covariance matrix":

$$\Sigma = \frac{1}{m} \sum_{i=1}^{n} (x^{(i)})(x^{(i)})^T$$

Compute "eigenvectors" of matrix Σ:

```
[U,S,V] = svd(Sigma);
```

Principal Component Analysis (PCA) algorithm

From `[U,S,V] = svd(Sigma)`, we get:

$$\rightarrow U = \begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(n)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times n}$$

k

$$x \in \mathbb{R}^n \longrightarrow z \in \mathbb{R}^k$$

$$\begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(k)} \\ | & | & & | \end{bmatrix}$$

Principal Component Analysis (PCA) algorithm

From `[U,S,V] = svd(Sigma)`, we get:

$$\rightarrow U = \begin{bmatrix} \underbrace{\begin{matrix} | & | \\ u^{(1)} & u^{(2)} \\ | & | \end{matrix}}_{k} & \dots & \begin{matrix} | \\ u^{(n)} \\ | \end{matrix} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

$$x \in \mathbb{R}^n \longrightarrow z \in \mathbb{R}^k$$

$$z = \underbrace{\begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(k)} \\ | & | & & | \end{bmatrix}}_{\substack{n \times k \\ U_{reduce}}}^{T} x = \underbrace{\underbrace{\begin{bmatrix} — (u^{(1)})^T — \\ \vdots \\ — (u^{(k)})^T — \end{bmatrix}}_{k \times n} \underbrace{x}_{n \times 1}}_{k \times 1}$$

$$z \in \mathbb{R}^k$$

Andrew Ng

Principal Component Analysis (PCA) algorithm summary

- After mean normalization (ensure every feature has zero mean) and optionally feature scaling:

$$\texttt{Sigma} = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)})(x^{(i)})^T$$

```
[U,S,V] = svd(Sigma);
```

Principal Component Analysis (PCA) algorithm summary

→ After mean normalization (ensure every feature has zero mean) and optionally feature scaling:

$$\text{Sigma} = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)})(x^{(i)})^T$$

```
[U,S,V] = svd(Sigma);
```

$$\underline{X} = \begin{bmatrix} — & x^{(1)T} & — \\ & \vdots & \\ — & x^{(m)T} & — \end{bmatrix}$$

$\boxed{\text{Sigma} = (1/m) * X' * X;}$

Principal Component Analysis (PCA) algorithm summary

- After mean normalization (ensure every feature has zero mean) and optionally feature scaling:

```
Sigma =
```
$$\frac{1}{m}\sum_{i=1}^{m}(x^{(i)})(x^{(i)})^T$$

```
[U,S,V] = svd(Sigma);
Ureduce = U(:,1:k);
z = Ureduce'*x;
```

$$X = \begin{bmatrix} — x^{(1)T} — \\ \vdots \\ — x^{(m)T} — \end{bmatrix}$$

$\boxed{\texttt{Sigma} = (1/m) * X' * X;}$

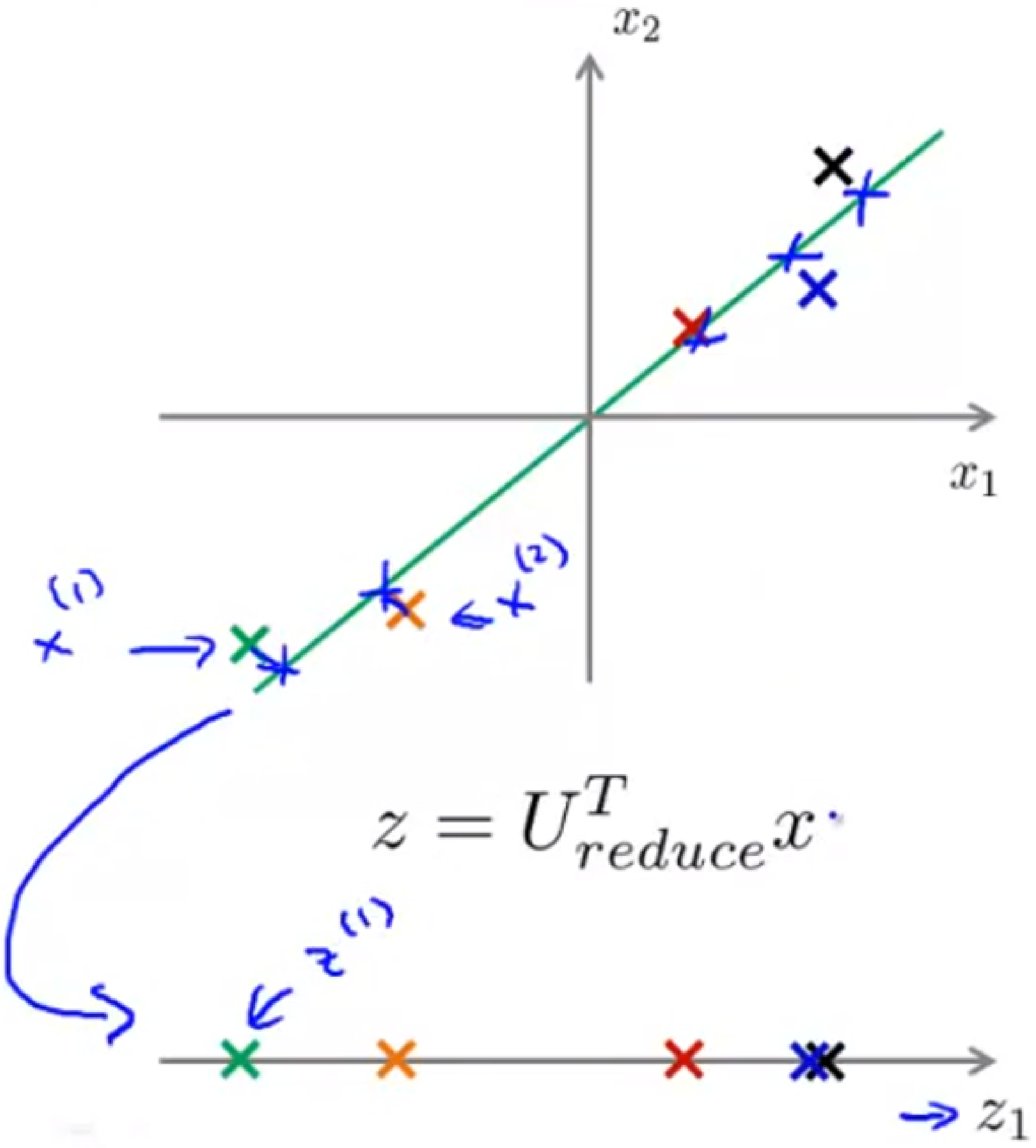
$x \in \mathbb{R}^n$ $\cancel{x_0 = 1}$

Reconstruction from compressed representation

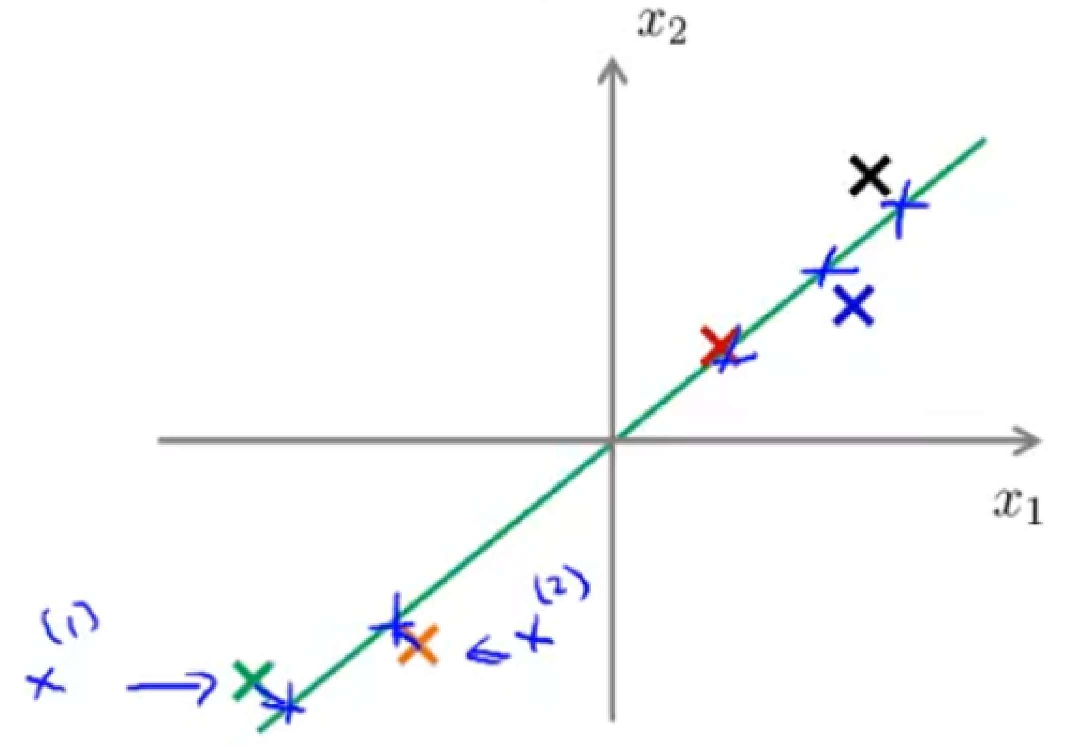
Principal Component Analysis

Unsupervised Learning

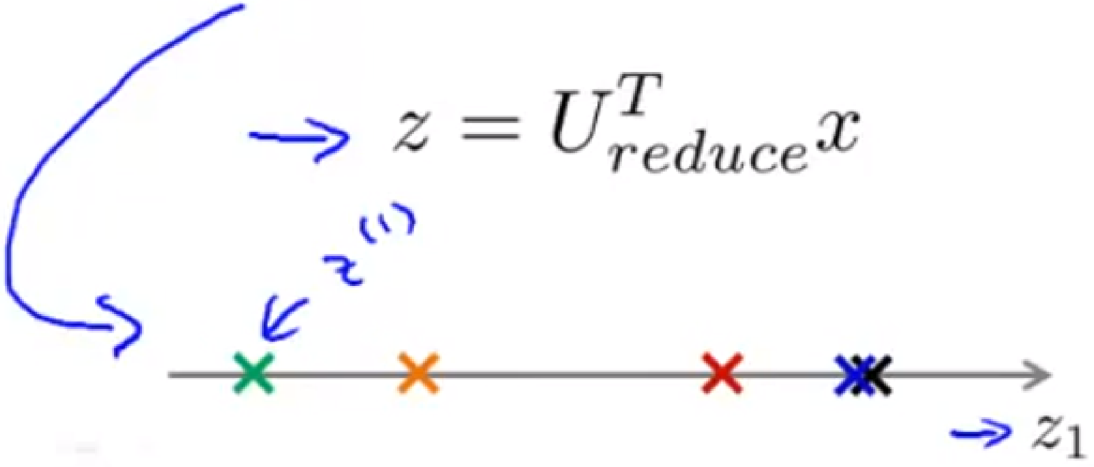
Reconstruction from compressed representation

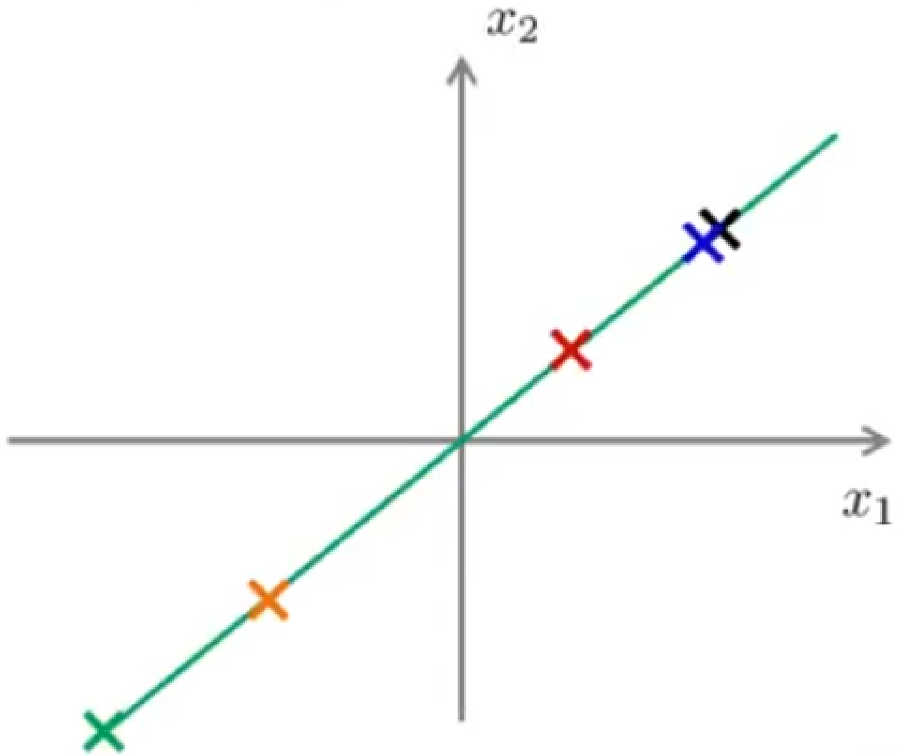


Reconstruction from compressed representation



$$\rightarrow z = U_{reduce}^T x$$



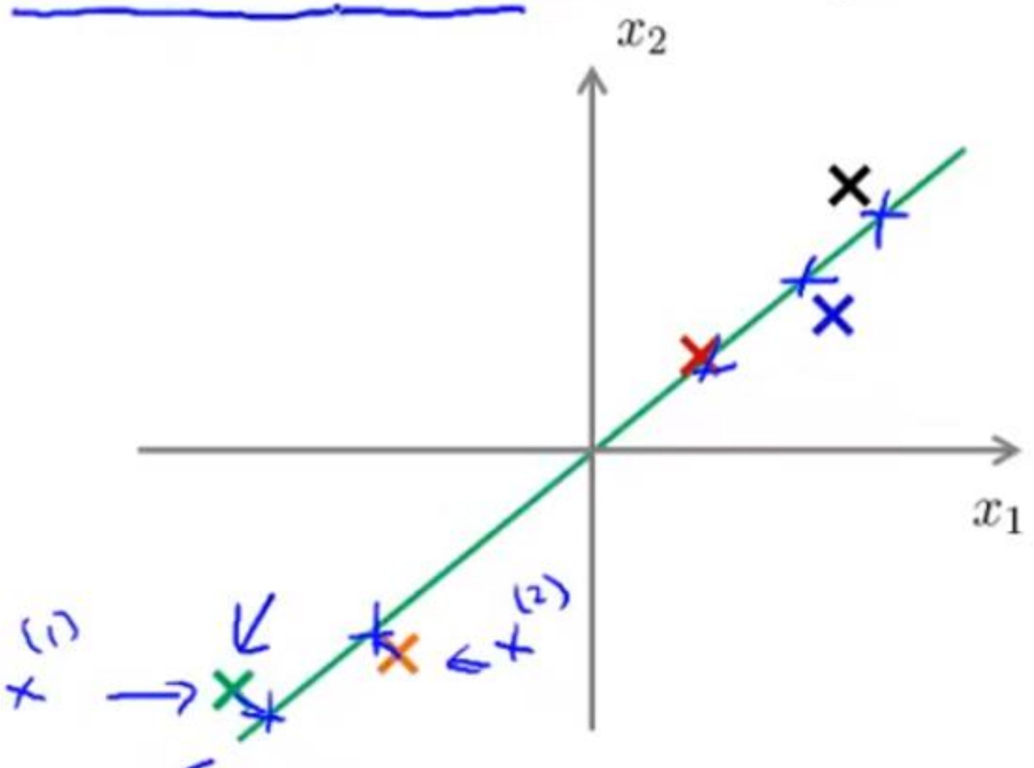


$$z \in \mathbb{R} \rightarrow x \in \mathbb{R}^2$$

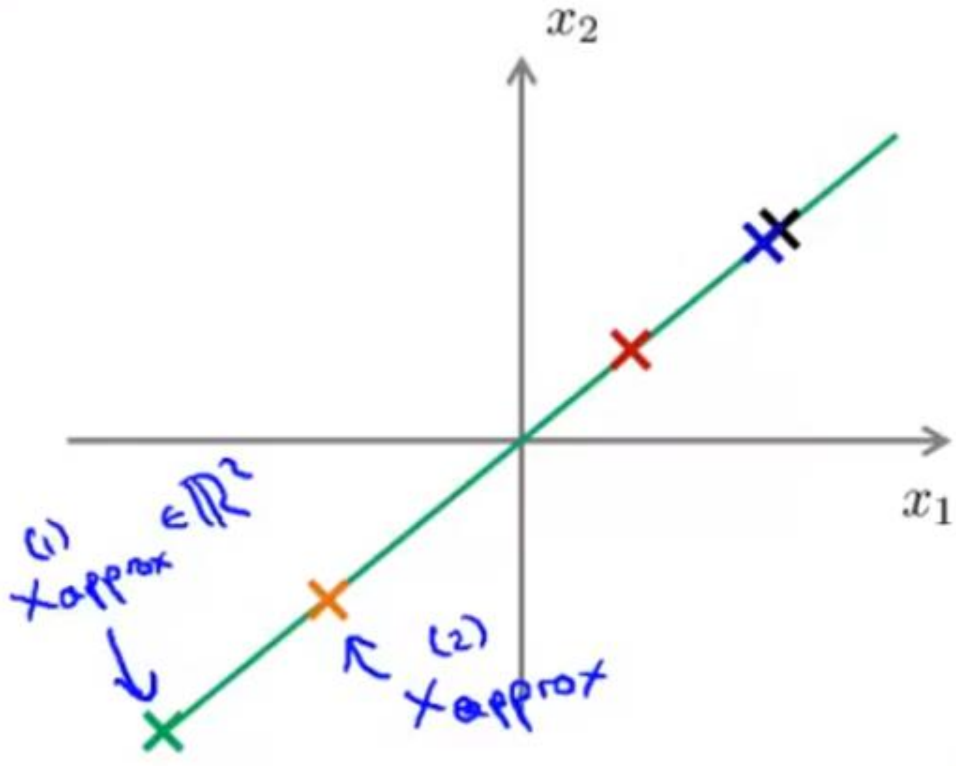
$$\approx x$$

$$\underbrace{x_{approx}}_{\in \mathbb{R}^n} = \underbrace{\underbrace{U_{reduce}}_{n \times k} \cdot \underbrace{z}_{k \times 1}}_{n \times 1}$$

Reconstruction from compressed representation



$$\rightarrow z = U_{reduce}^T x$$



$$z \in \mathbb{R} \rightarrow x \in \mathbb{R}^2$$

$$x \approx \underbrace{x_{approx}^{(1)}}_{\mathbb{R}^n} = \underbrace{\underbrace{U_{reduce}}_{n \times k} \cdot \underbrace{z^{(1)}}_{k \times 1}}_{n \times 1}$$