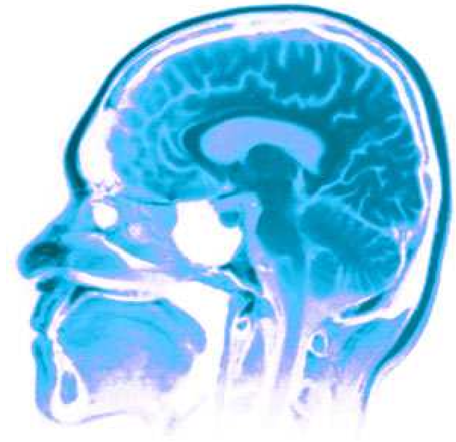
CPSC340



Principal Component Analysis (PCA)

Nando de Freitas

October, 2012

University of British Columbia

Outline of the lecture

This lecture explains one of the most widely used techniques for dimensionality reduction: Principal Component Analysis (PCA) . You will learn:

- ❑ How PCA **dimensionality reduction** with the SVD
- ❑ To Apply PCA for **visualization** tasks
- ❑ That PCA is a linear method that minimizes reconstruction error. That is: the SVD emerges as the optimal solution to an obvious optimization problem, namely ***what you imagine should look like what you see!***

data

m attributes

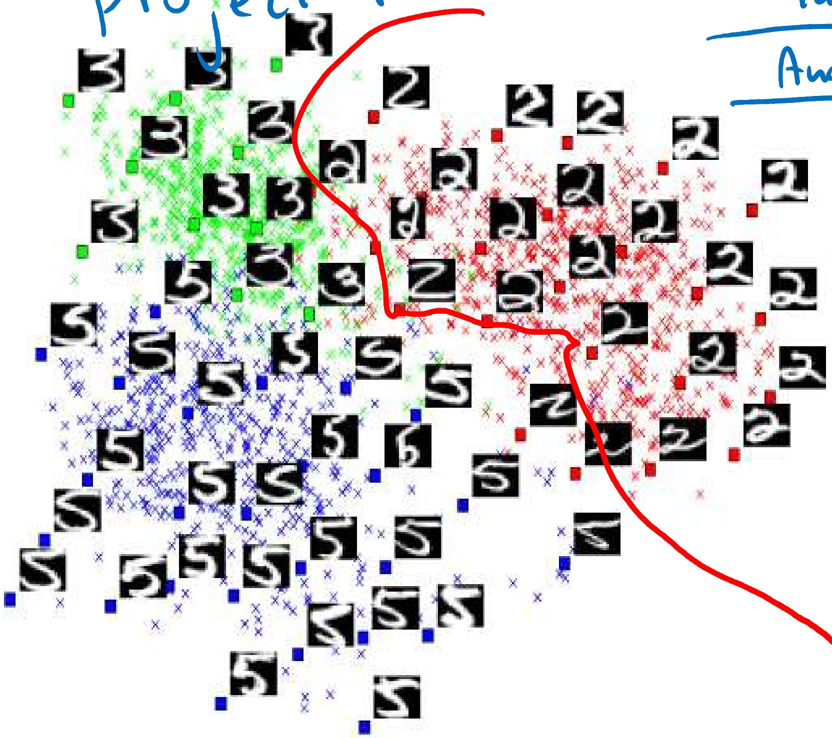
Country	area	income	Population	etc
Canada	1	0.2	400	...

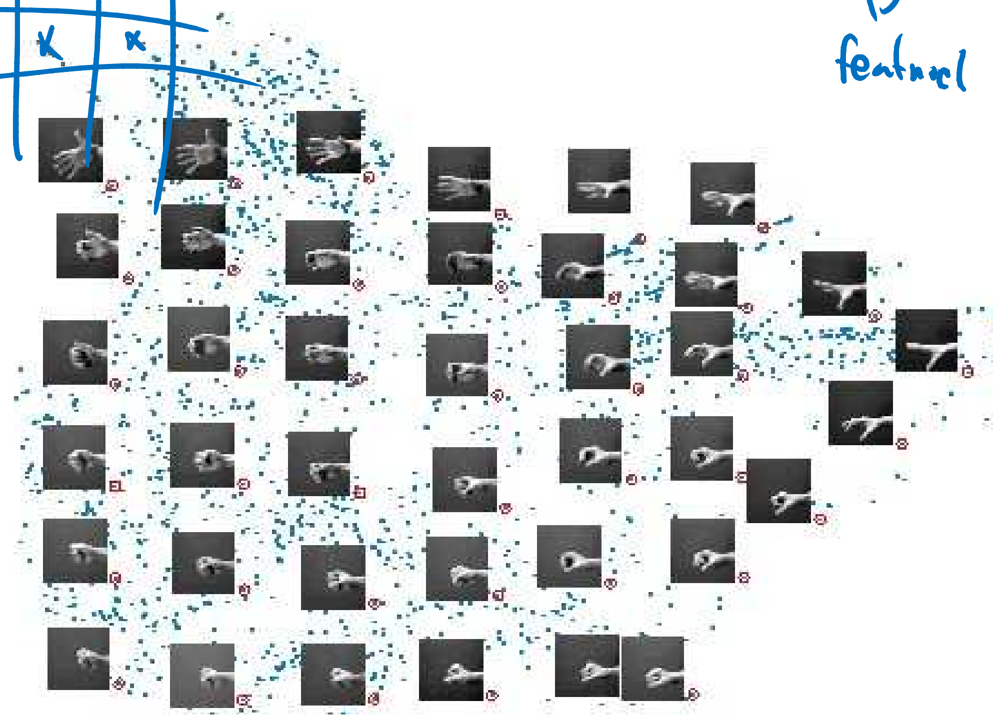
feature2

feature1

Project to d=2.

country	feature1	feature2
Canada	x	x
Austria	x	x





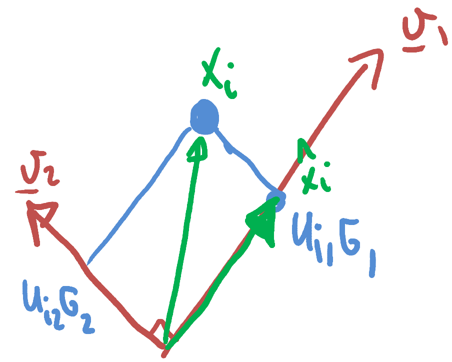
PCA derivation: 2D to 1D

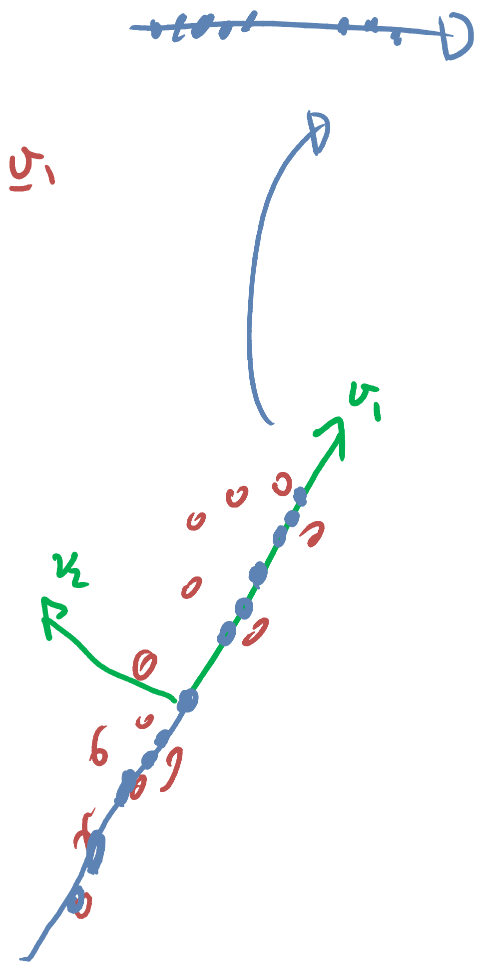
$$
\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{i1} & x_{i2} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{bmatrix}
=
\begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \\ \vdots & \vdots \\ u_{i1} & u_{i2} \\ \vdots & \vdots \\ u_{n1} & u_{n2} \end{bmatrix}
\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}
\begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{bmatrix}
$$

x_i

$$\mathbf{x}_i = u_{i1}\sigma_1\mathbf{v}_1^T + \overset{1\times 1}{u_{i2}}\,\overset{1\times 1}{\sigma_2}\,\overset{1\times 2}{\mathbf{v}_2^T}$$

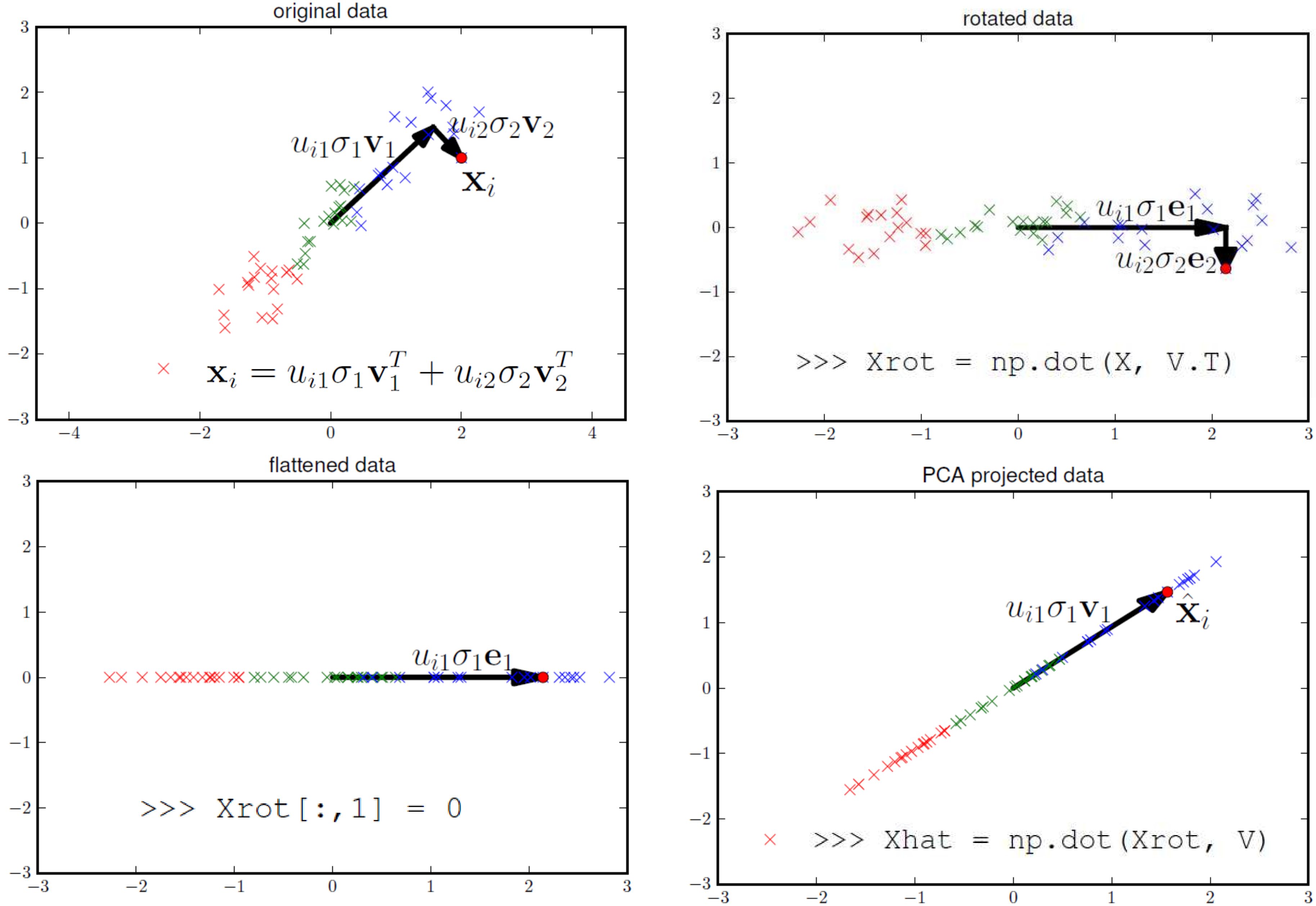
$$\widehat{\mathbf{x}}_i = u_{i1}\sigma_1\mathbf{v}_1^T \Leftarrow \text{truncation}$$



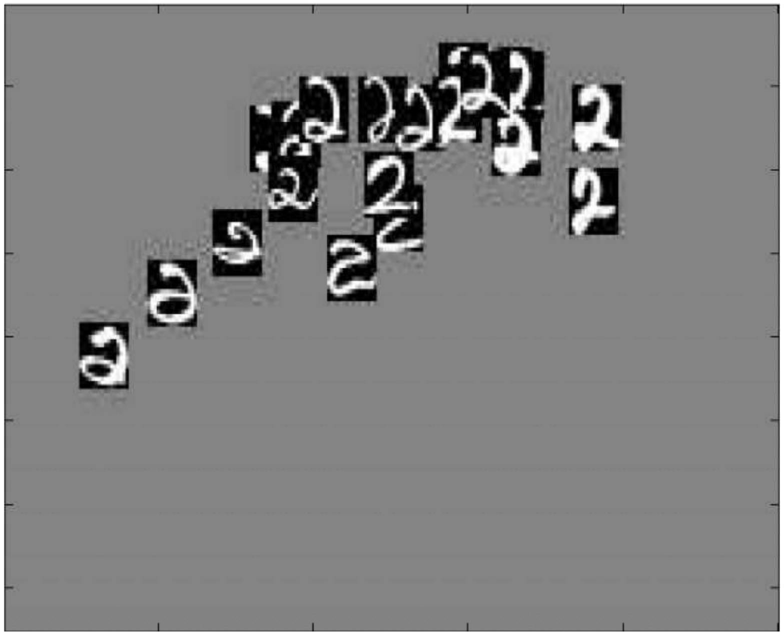


How PCA works:

$$\mathbf{X}_{\text{rot}} = \begin{bmatrix} (\mathbf{V}\mathbf{x}_1)^T \\ \vdots \\ (\mathbf{V}\mathbf{x}_n)^T \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^T \mathbf{V}^T \\ \vdots \\ \mathbf{x}_n^T \mathbf{V}^T \end{bmatrix} = \mathbf{X}\mathbf{V}^T$$



PCA for 2D visualization



I_i: image i, which is 40×40

1, 2, 3 → [1][2][3][ ] . . . [40] ← I_i

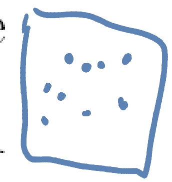
1×1600

Next,

$$A = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_m \end{bmatrix}$$

database of m images of size 1600.

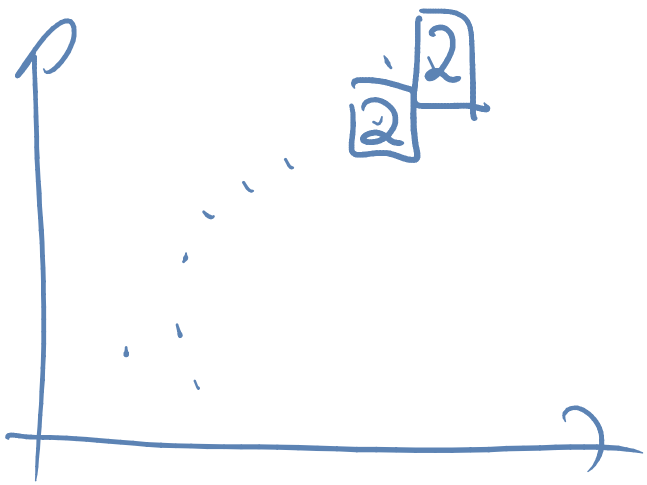
For example, we can take several 16×16 images of the digit 2 and project them to 2D. The images can be written as vectors with 256 entries. We then from the matrix $\mathbf{A} \in \mathbb{R}^{n \times 256}$, carry out the SVD and truncate it to $k = 2$. Then the components $\mathbf{U}_k \mathbf{\Sigma}_k$ are 2 vectors with n data entries. We can plot these 2D points on the screen to visualize the data.



PCA in python

```
>>> U, S, V = svd(A)
>>> k = 2
>>> Z = dot(U[:,:k], eye(k) * S[:k])
>>> figure(1)
>>> plot(Z[:,0], Z[:,1], ’ro’)
>>> grid()
```

V^T



Standardize the data first!

*It is a good idea to standardize the data before doing PCA so that all entries of matrix **X** have similar magnitude. We do it as follows:*

$$\overline{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$$

$$\sigma_j^2 = \frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2$$

$$x_{ij} = \frac{x_{ij} - \overline{x}_j}{\sigma_j}$$

	cat	fun	educational
web1	100	4	1
web2	1000	3	0
web2	0	0	1
web7	5000	2	0

$$X \quad \begin{array}{l} \text{web1} \\ \text{web2} \end{array} \begin{vmatrix} 0.3 & 0.3 & 1 \\ 0.7 & 0.9 & 0.2 \end{vmatrix}$$

Advanced: PCA as orthogonal reconstruction

Suppose we are given the n by d data matrix ***X*** *and that we want to find a* ***linear reconstruction*** *of this matrix in terms of a set of neural responses* ***z*** *and synaptic weights* ***w****. We can do this by minimizing the following quadratic difference:*

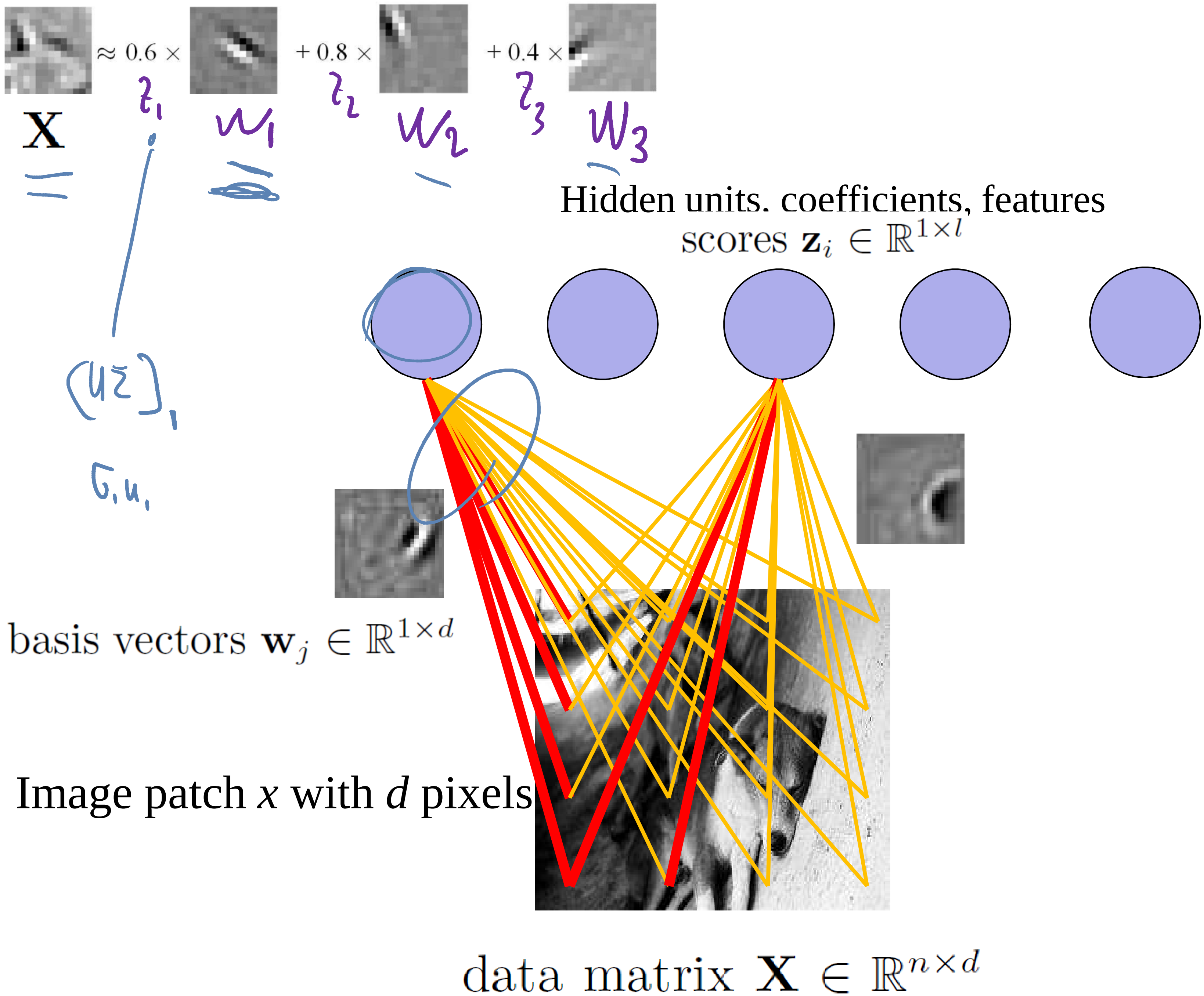
$$J(\mathbf{W}, \mathbf{Z}) = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2 = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{z}_i \mathbf{W}||^2$$

Subject to the constraint that ***W*** *is orthogonal.*

If we solve the above optimization problem, the answer is:

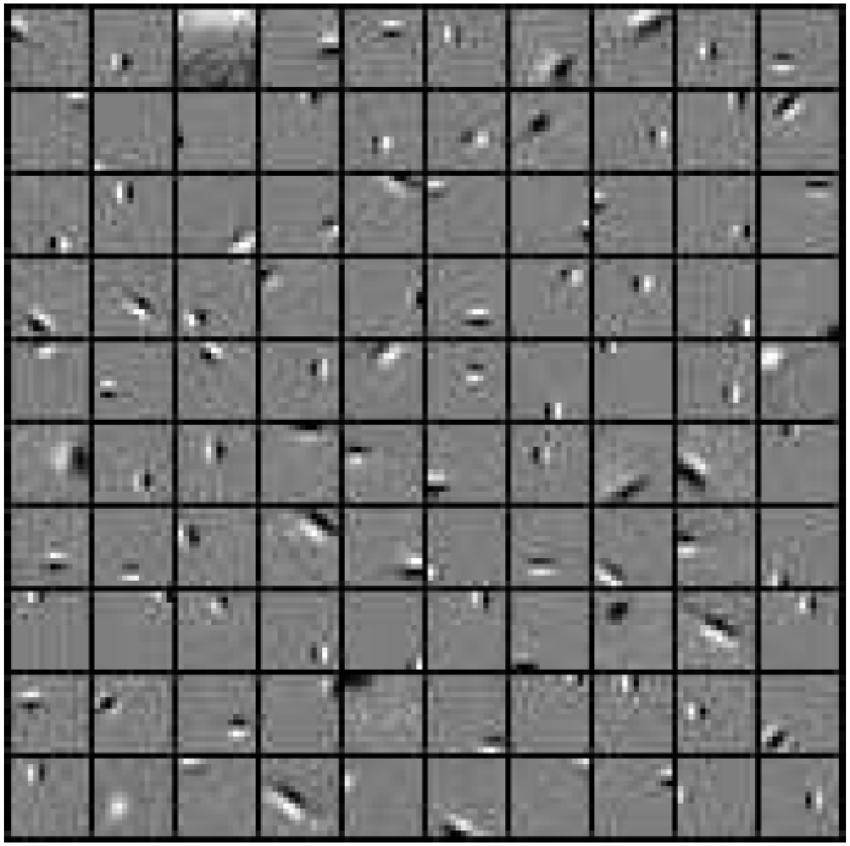
$$\boldsymbol{W = V^T}$$
$$\boldsymbol{Z = U\,\Sigma}$$

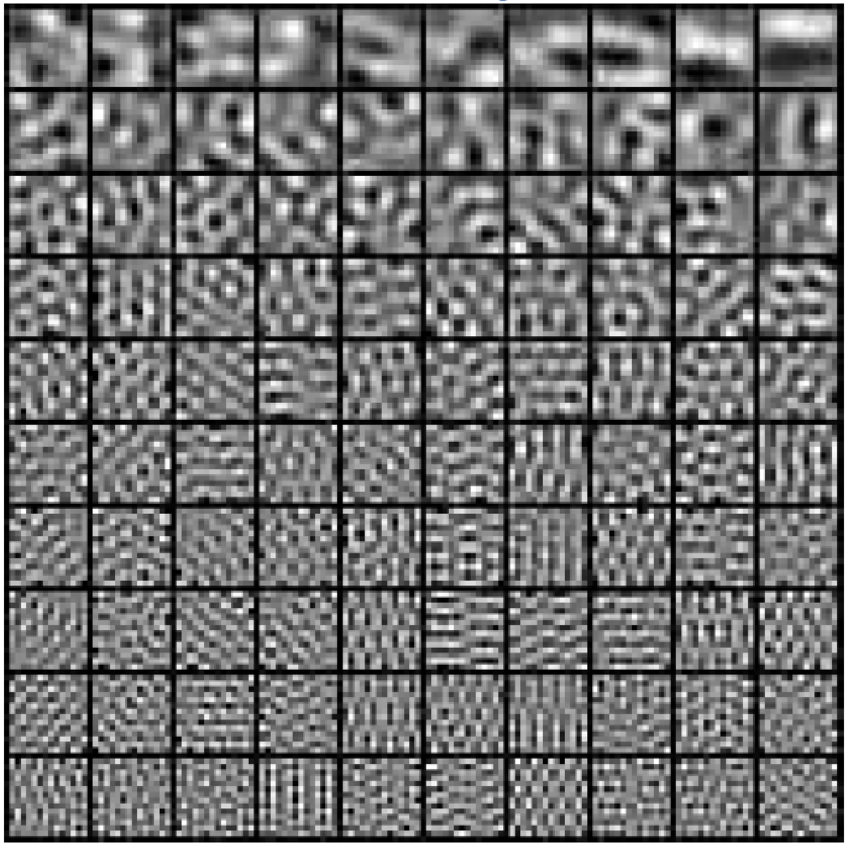
That is, the SVD gives the optimal linear reconstruction.



$\mathbf{X} \approx 0.6 \times z_1 W_1 + 0.8 \times z_2 W_2 + 0.4 \times z_3 W_3$
$(U\Sigma)_1$
$\sigma_1 u_1$
Hidden units. coefficients, features
scores $\mathbf{z}_i \in \mathbb{R}^{1\times l}$
basis vectors $\mathbf{w}_j \in \mathbb{R}^{1\times d}$
Image patch *x* with *d* pixels
data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$

The weights found with **sparse coding** and **PCA**





$$\mathbf{B}^*, \mathbf{C}^* = \arg\min_{\mathbf{B},\mathbf{C}} ||\mathbf{X} - \mathbf{BC}||_2^2 + \lambda \, ||\mathbf{C}||_1$$

$$s.t. \quad ||\mathbf{b}_j||_2^2 = 1, \ \forall j.$$

Next lecture

In the next lecture we begin our introduction to supervised learning.