CSC 390
Topics in Artificial Intelligence

"Unsupervised Machine Learning"

Fall 2016

Prof. Sara Mathieson
Smith College

Outline: 10/6

- Recap Homework 2
- Recap Lab 3 and finish PCA
- Autoencoders
- Reminders:
  - Office hours today 4-5pm (Ford 355)
  - Homework 3 due tonight

Recap Homework 2

Python tips

1) **i,j = minDist(distance_matrix)**

Python tips

1) **i,j = minDist(distance_matrix)**

2) **float(x)/y**, not float(x/y)

Python tips

1) **i,j = minDist(distance_matrix)**

2) **float(x)/y**, not float(x/y)

3) Parenthesis often unnecessary
   1) Order of operations: **a*b+c**, not (a*b)+c
   2) Conditionals: **if x == 2**, not if (x == 2)

Python tips

1) **i,j = minDist(distance_matrix)**

2) **float(x)/y**, not float(x/y)

3) Parenthesis often unnecessary
   1) Order of operations: **a*b+c**, not (a*b)+c
   2) Conditionals: **if x == 2**, not if (x == 2)

4) Lists: **new_cluster = cluster_i + cluster_j**
   1) i.e. [c,e] = [c]+[e]

Python tips

1) **i,j = minDist(distance_matrix)**

2) **float(x)/y**, not float(x/y)

3) Parenthesis often unnecessary
   1) Order of operations: **a*b+c**, not (a*b)+c
   2) Conditionals: **if x == 2**, not if (x == 2)

4) Lists: **new_cluster = cluster_i + cluster_j**
   1) i.e. [c,e] = [c]+[e]

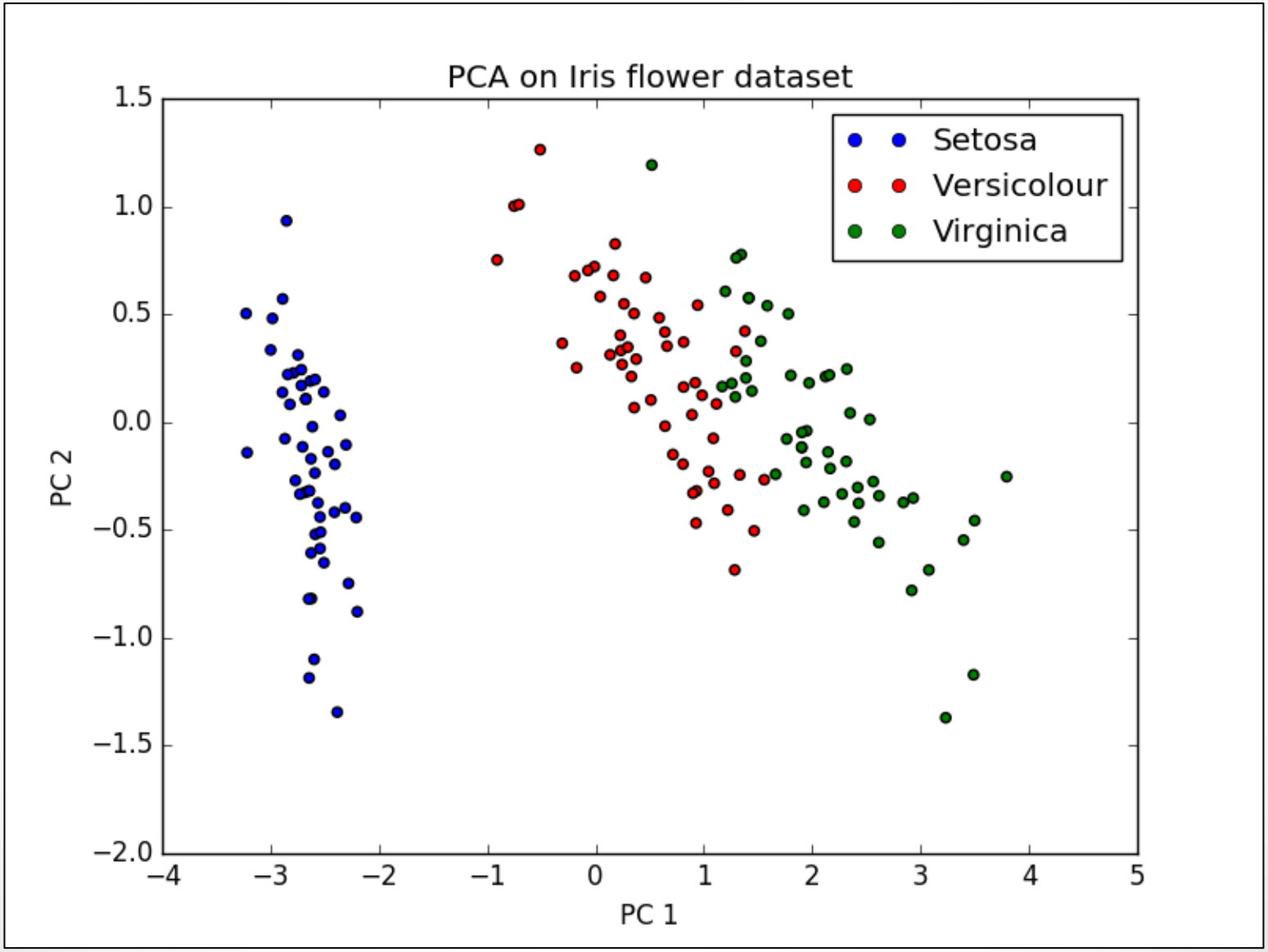
5) Camel case: functions, underscores: variables

Recap Lab 3

Lab 3 Solution

```
# load iris dataset
iris = datasets.load_iris()
X = iris.data
y = iris.target

# PCA
pca = decomposition.PCA(n_components=2)
pca.fit(X)
X_transform = pca.transform(X)

# list comprehension for colors
colors = [color_dict[y_value] for y_value in y]

# create a scatter plot
plt.scatter(X_transform[:,0], X_transform[:,1], c=colors)
plt.xlabel('PC 1')
plt.ylabel('PC 2')
plt.title('PCA on Iris flower dataset')

# create legend
leg_objects = []
for i in range(3):
    circle, = plt.plot([], color_dict[i] + 'o')
    leg_objects.append(circle)
plt.legend(leg_objects,names)

# show the plot
plt.show()
```

Lab 3 Figure


PCA on Iris flower dataset
Setosa
Versicolour
Virginica
PC 2
PC 1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−4
−3
−2
−1
0
1
2
3
4
5

Autoencoders

Motivation

- Data compression, especially for images
- Incorporate non-linear combinations of input features

Motivation

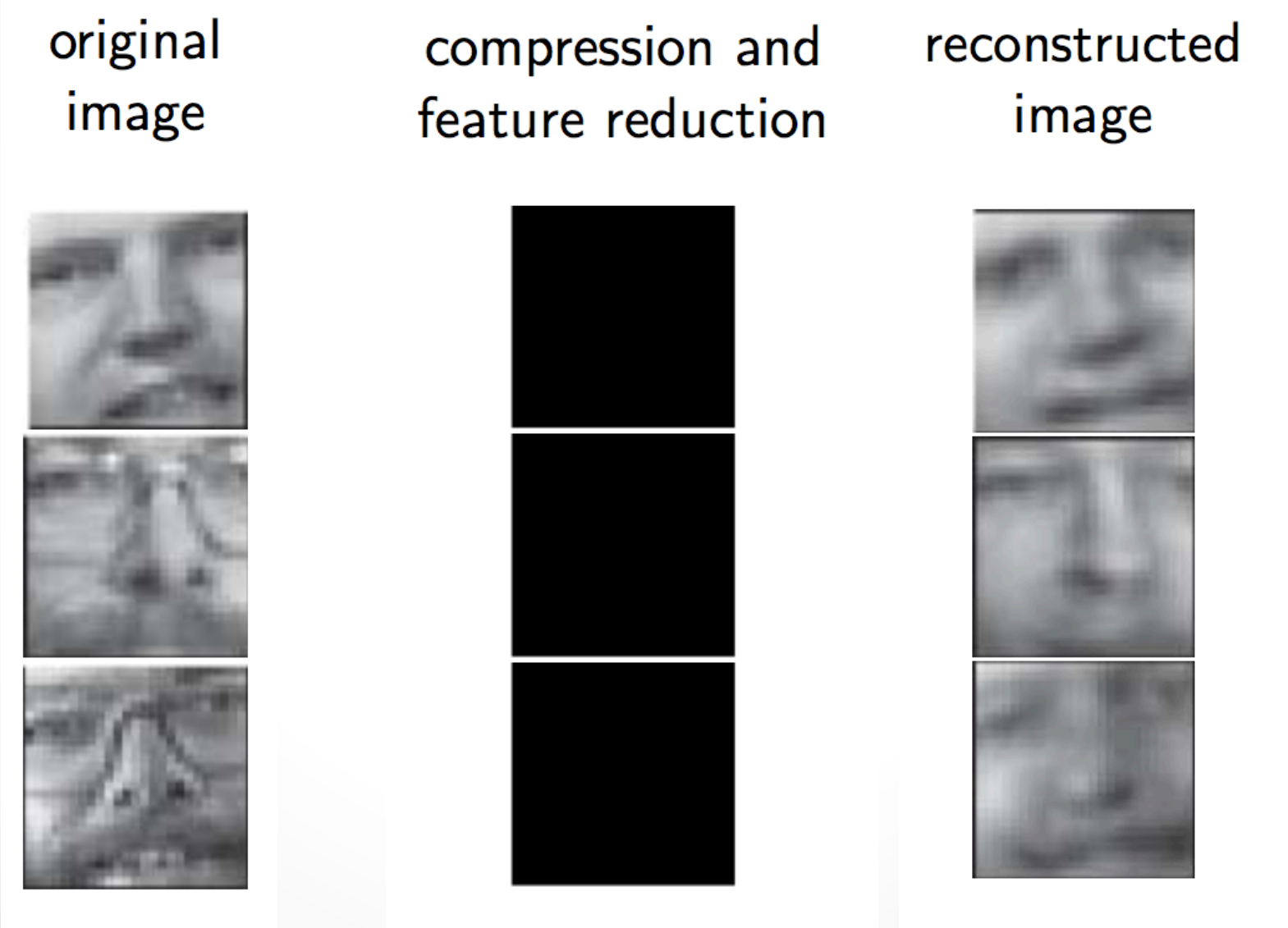
original
image

Motivation

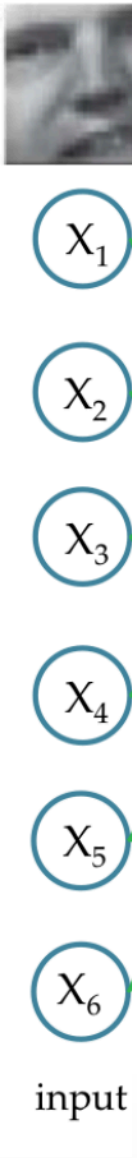
original image

compression and feature reduction

Motivation

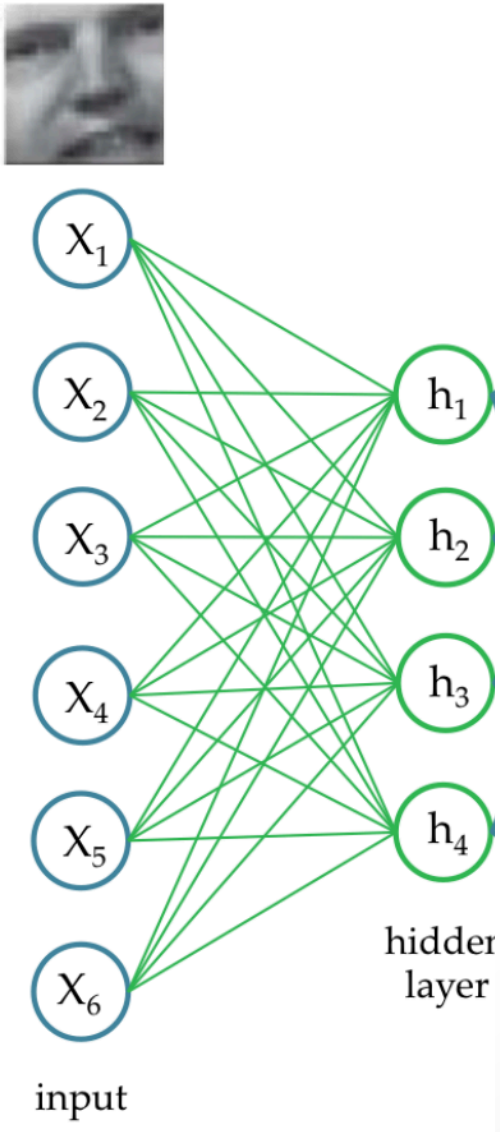

original image
compression and feature reduction
reconstructed image

Autoencoder algorithm



Autoencoder algorithm

1. Project data into a lower dimension:

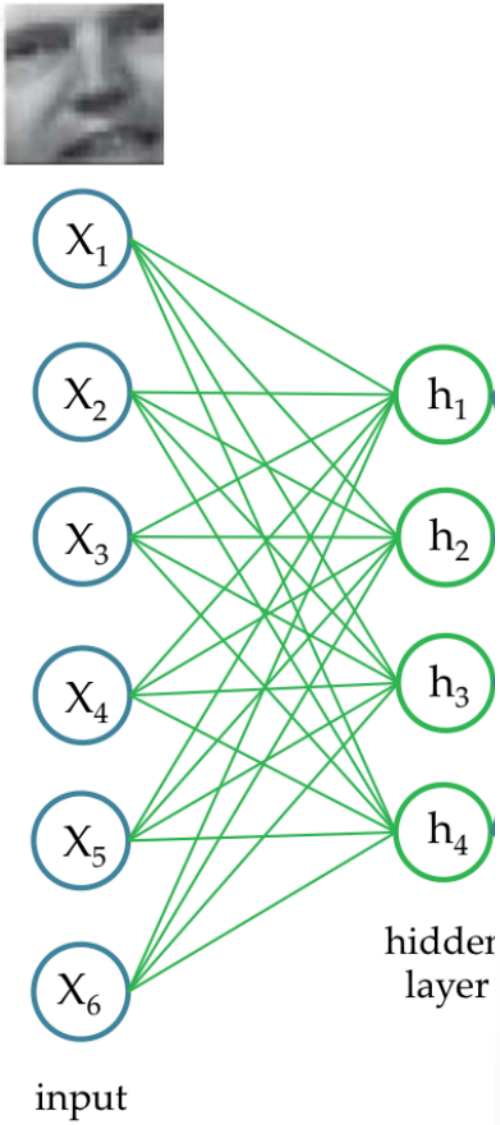


Autoencoder algorithm

1. Project data into a lower dimension:

$$h_j = \sigma\left(W_j^{(1)} \cdot x\right)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



Autoencoder algorithm

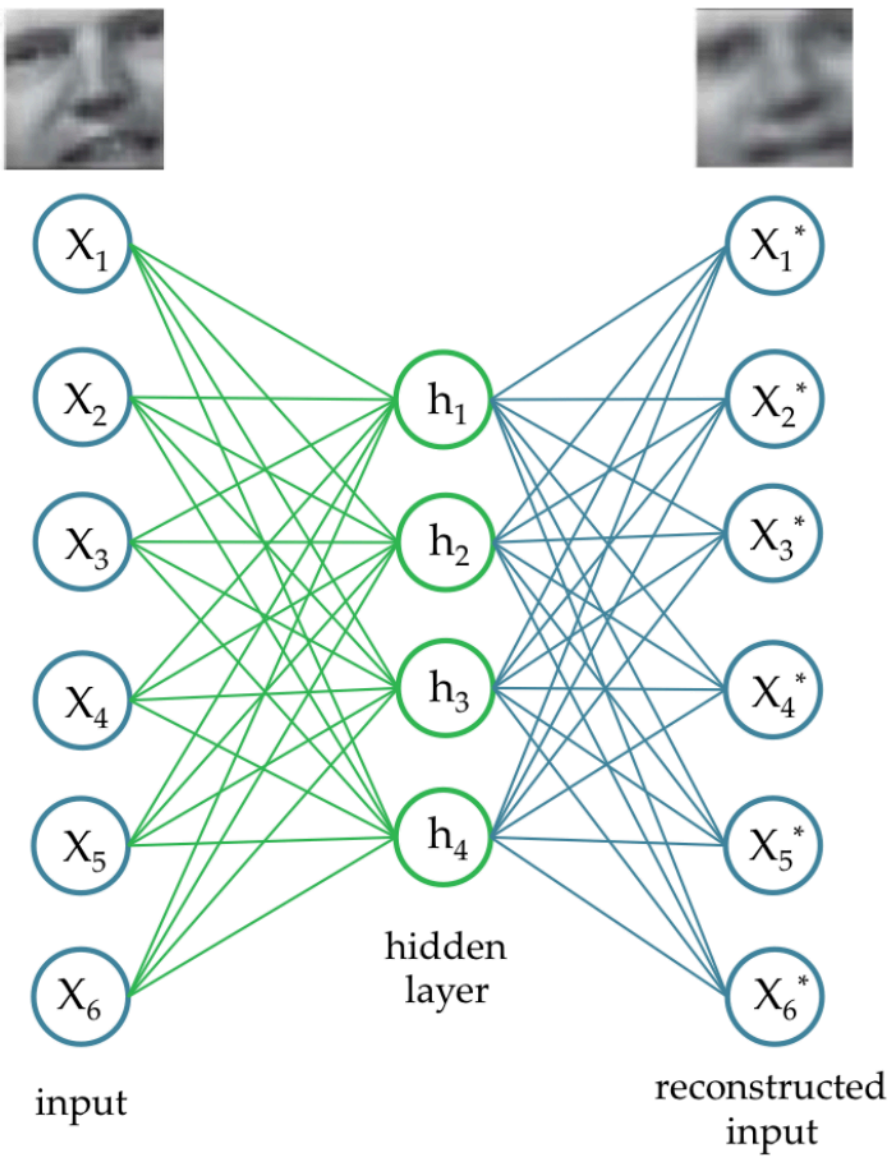
1. Project data into a lower dimension:

$$h_j = \sigma\left(W_j^{(1)} \cdot x\right)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

2. Minimize objective function:

$$J_x(W) = \frac{1}{2}||x - x^*||^2$$

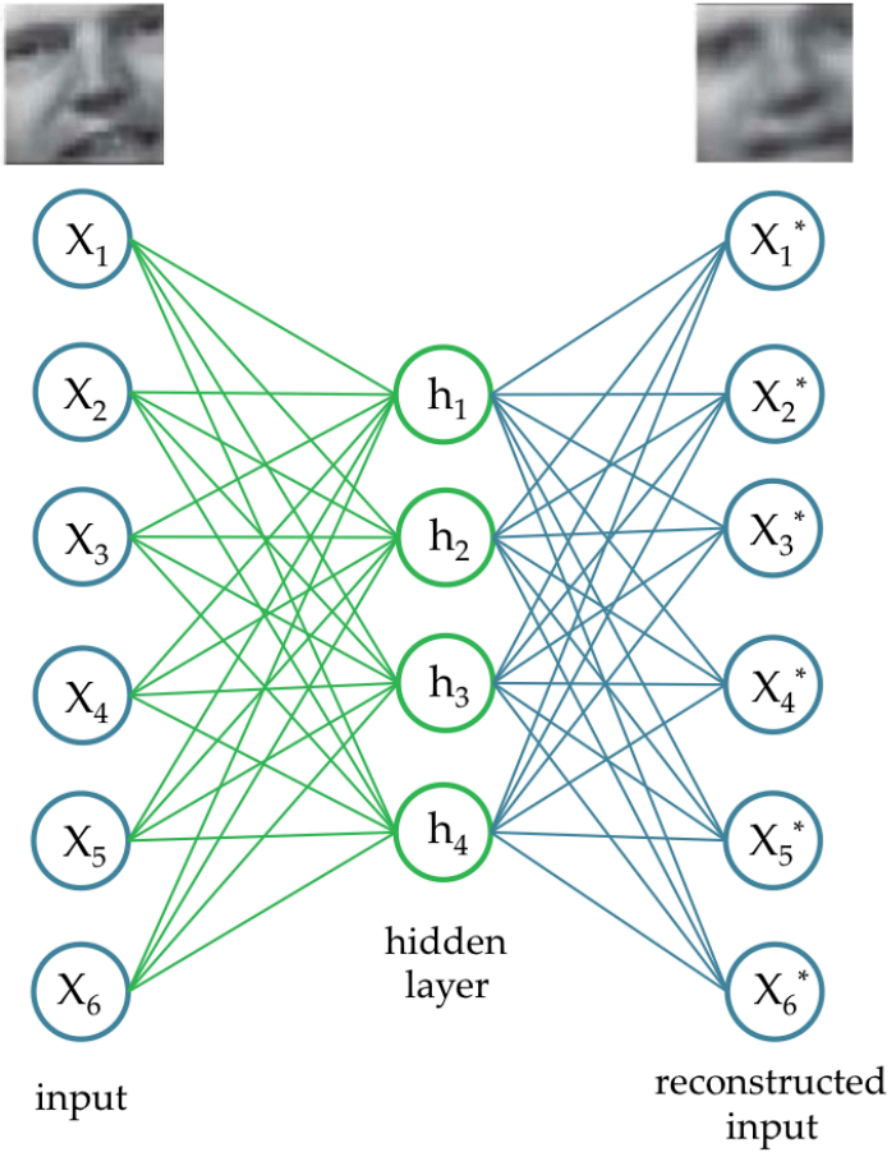


Autoencoder algorithm

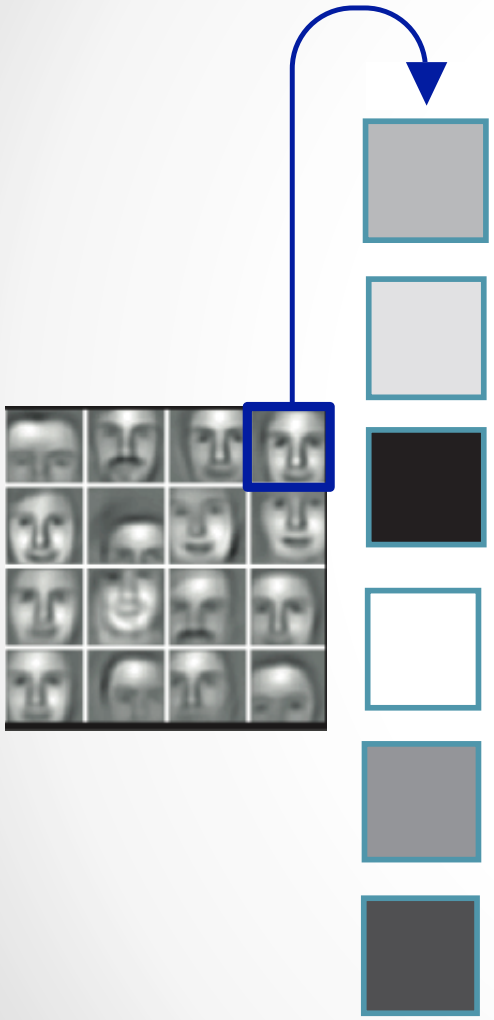
1. Project data into a lower dimension

2. Then try to reconstruct the data

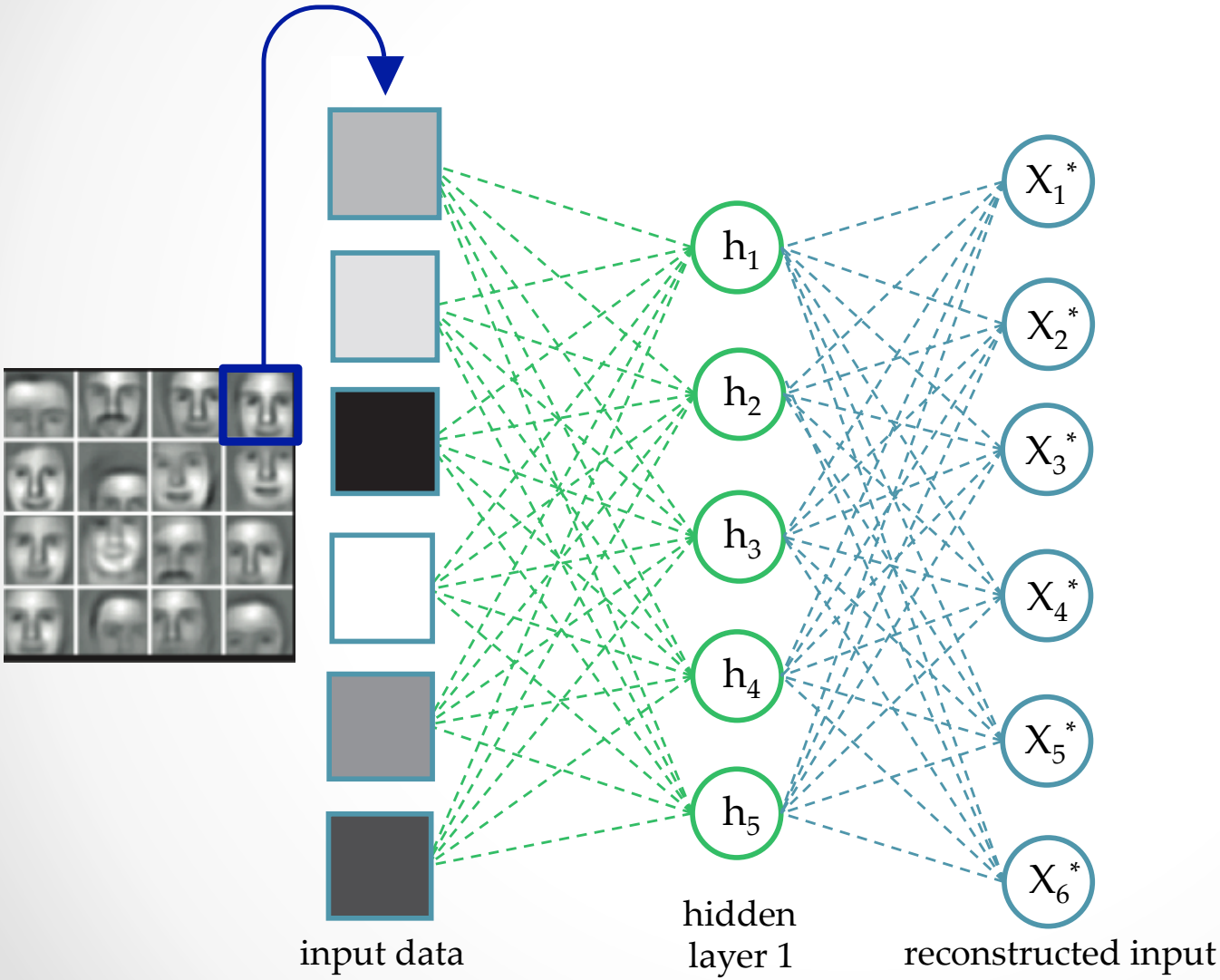
- learns a "basis" of features automatically
- robust to noise or uninformative data



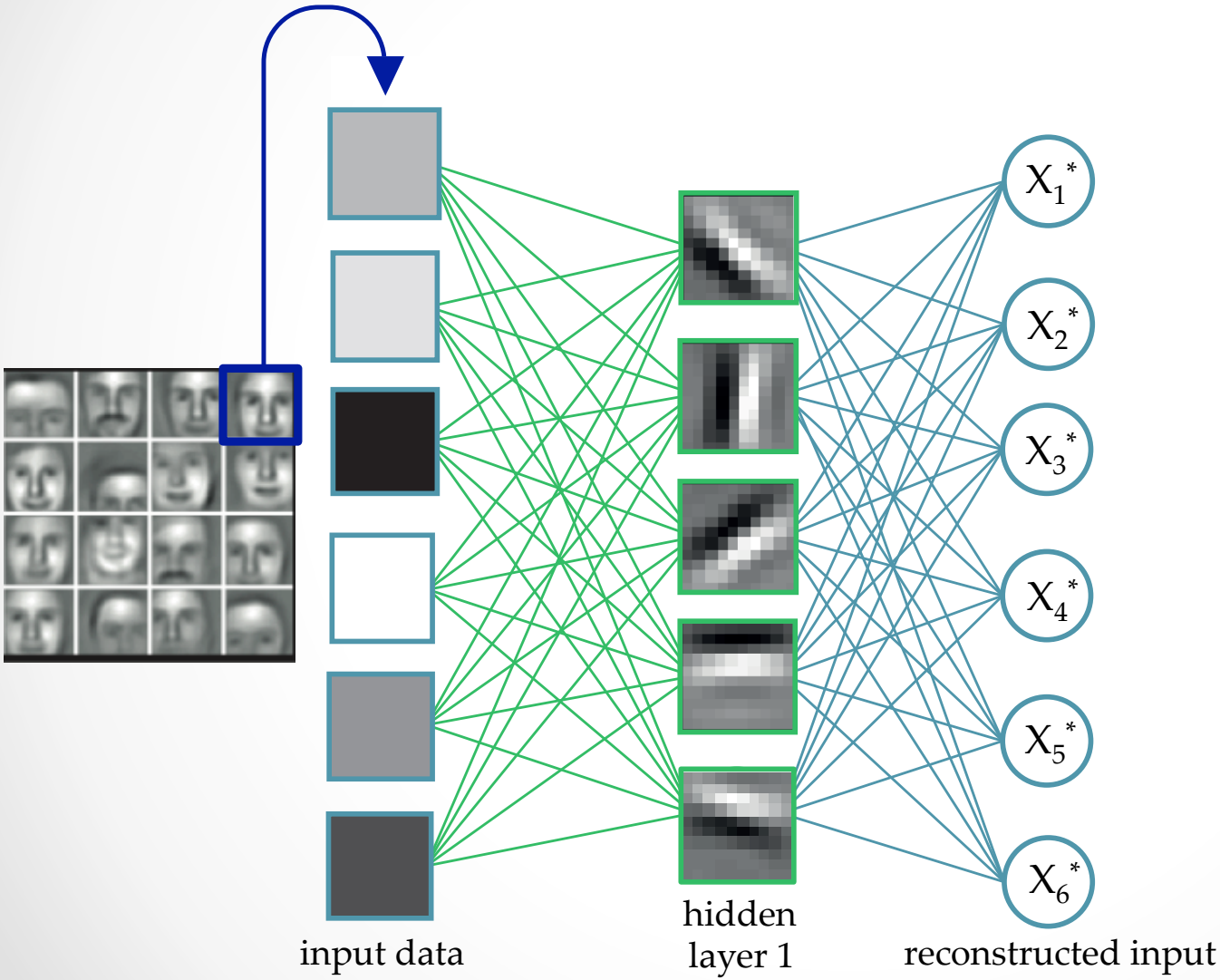
Autoencoder example



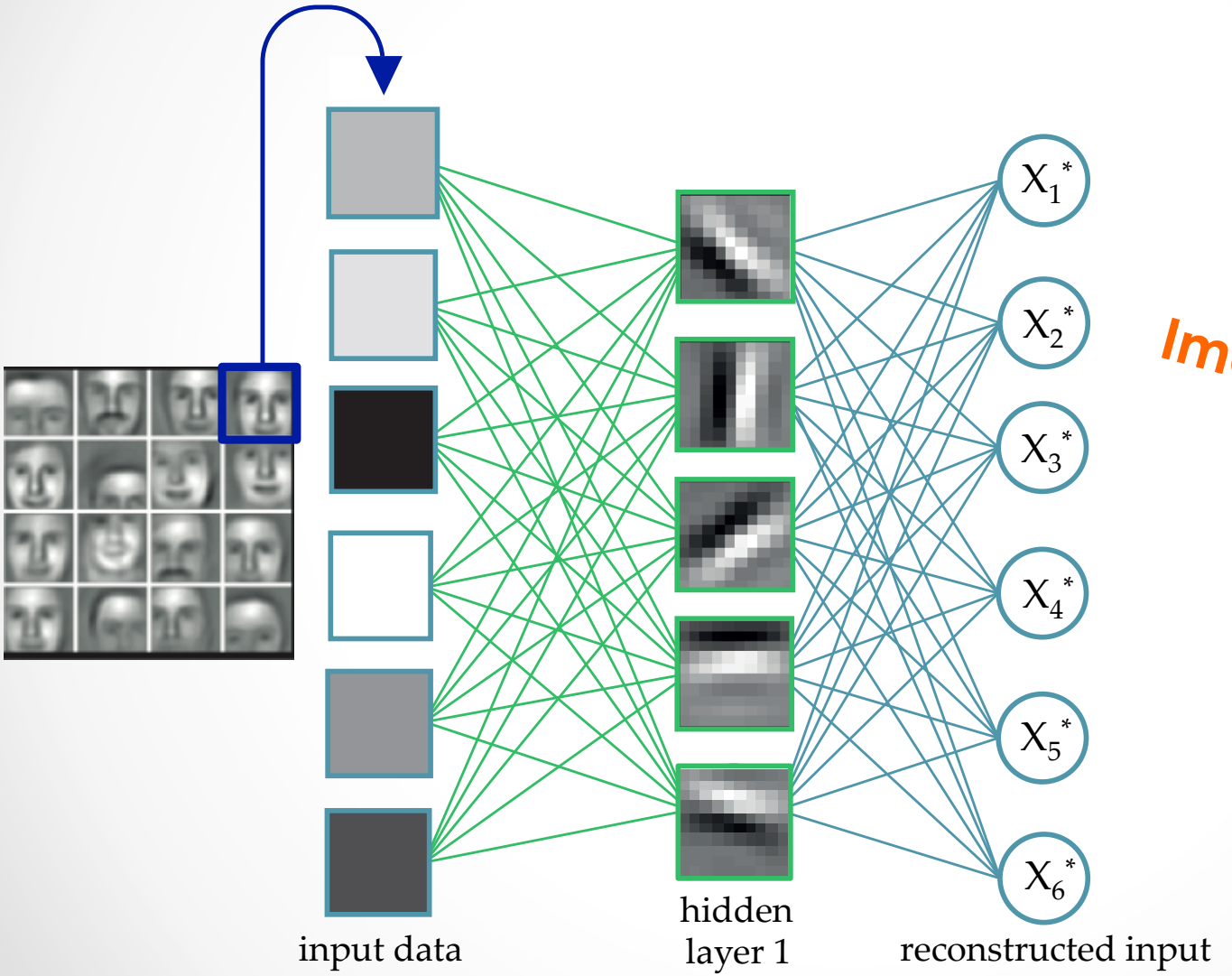
Autoencoder example



Autoencoder example



Autoencoder example



Images are made up of lines!