University of Nottingham
UK | CHINA | MALAYSIA

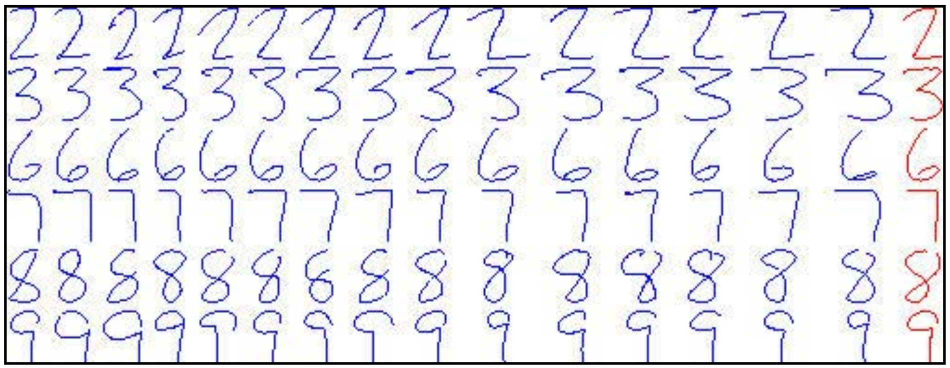
COMP3055
Machine Learning

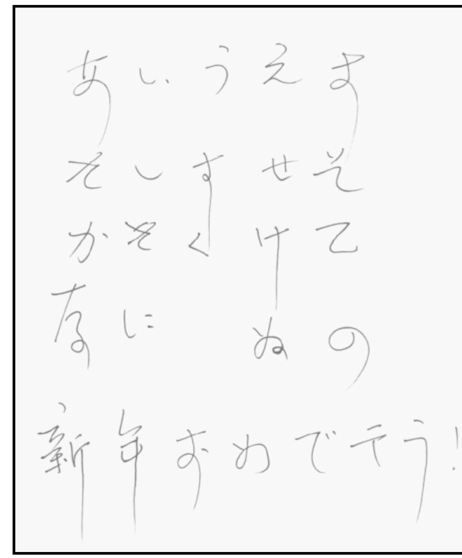
Topic 1 – Design a Learning System

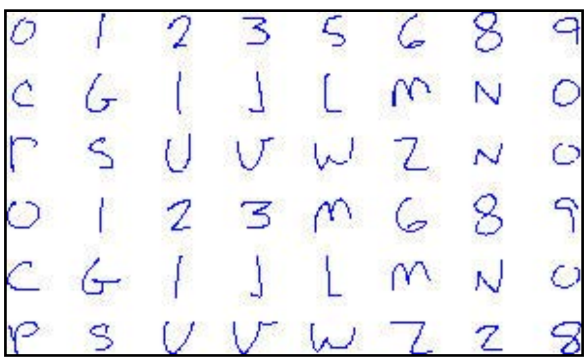
Dr. Zheng LU
2018 Autumn

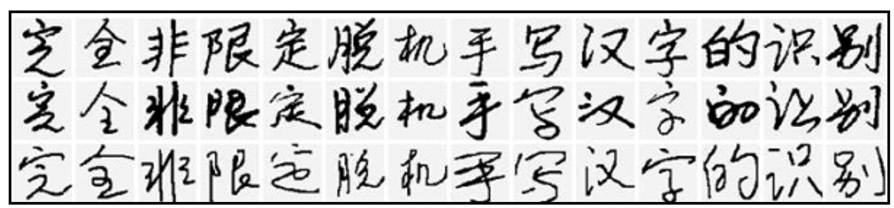
Motivating Problems

Handwritten Character Recognition









Motivating Problems

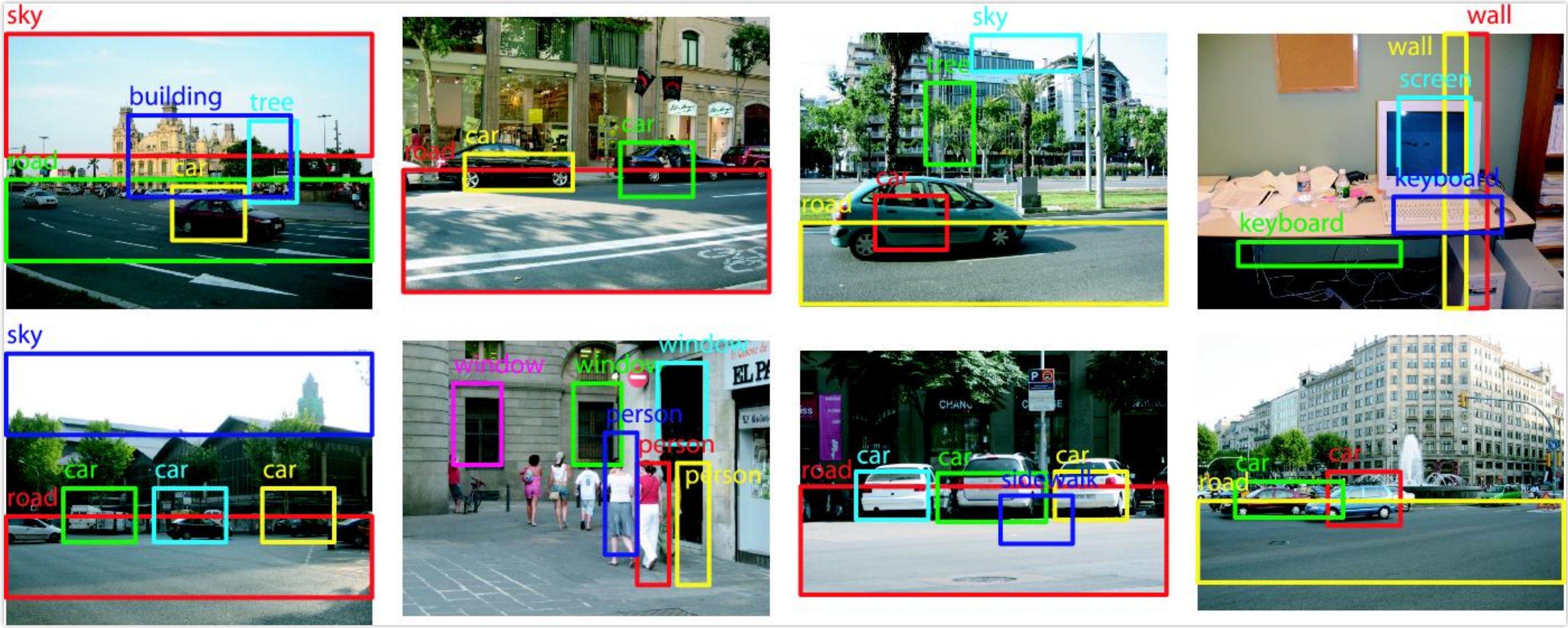
Fingerprint Recognition (e.g., border control)

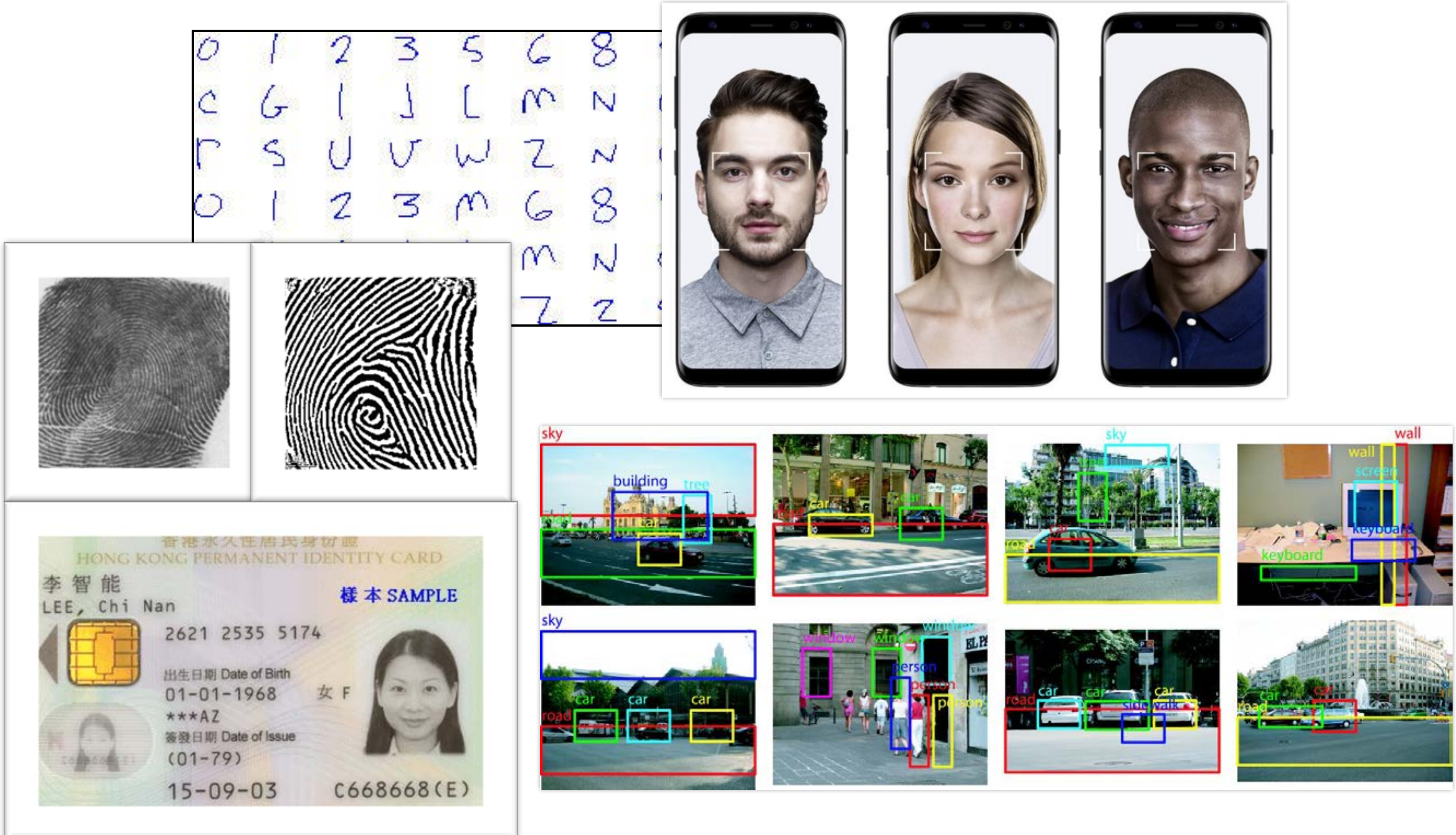
Motivating Problems

Face Recognition

Motivating Problems

Object Recognition





...

Can Machines Learn to Solve These Problems?

Definition of Learning

A computer program is said to learn from experience ***E*** with respect to some class of tasks ***T*** and performance measure ***P***, if its performance at tasks in ***T***, as measured by ***P***, improves with experience ***E***.

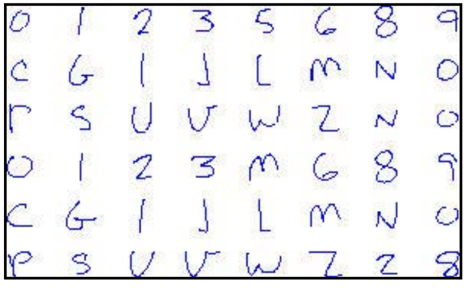
-- from Mitchell, Machine Learning, McGraw-Hill, 1997

Definition of Learning

What does this mean exactly?

For example, handwriting recognition problem

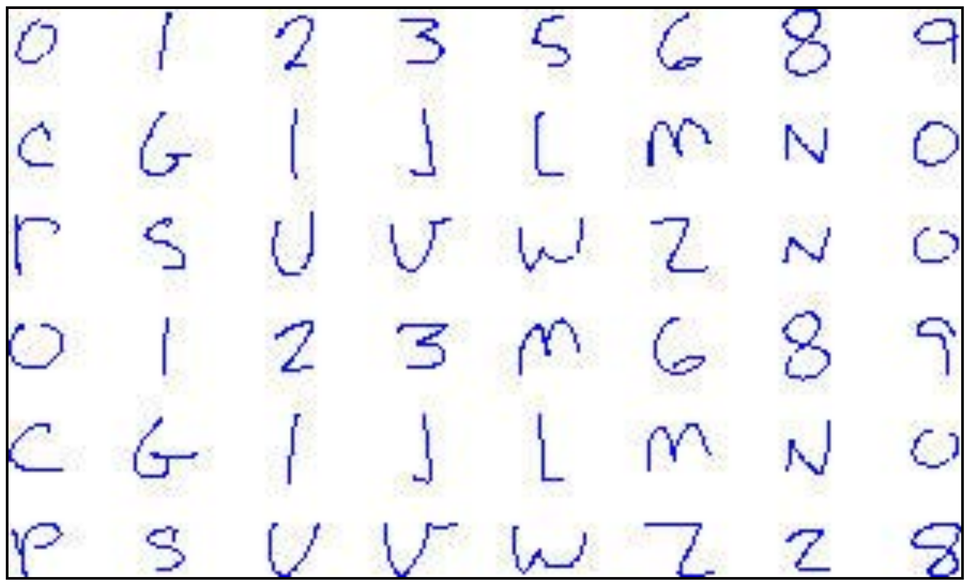
- Task ***T***: Recognizing hand written characters
- Performance measure ***P***: percent of characters correctly classified
- Training experience ***E***: a database of handwritten characters with given classifications



Definition of Learning

What are design issues and approaches?

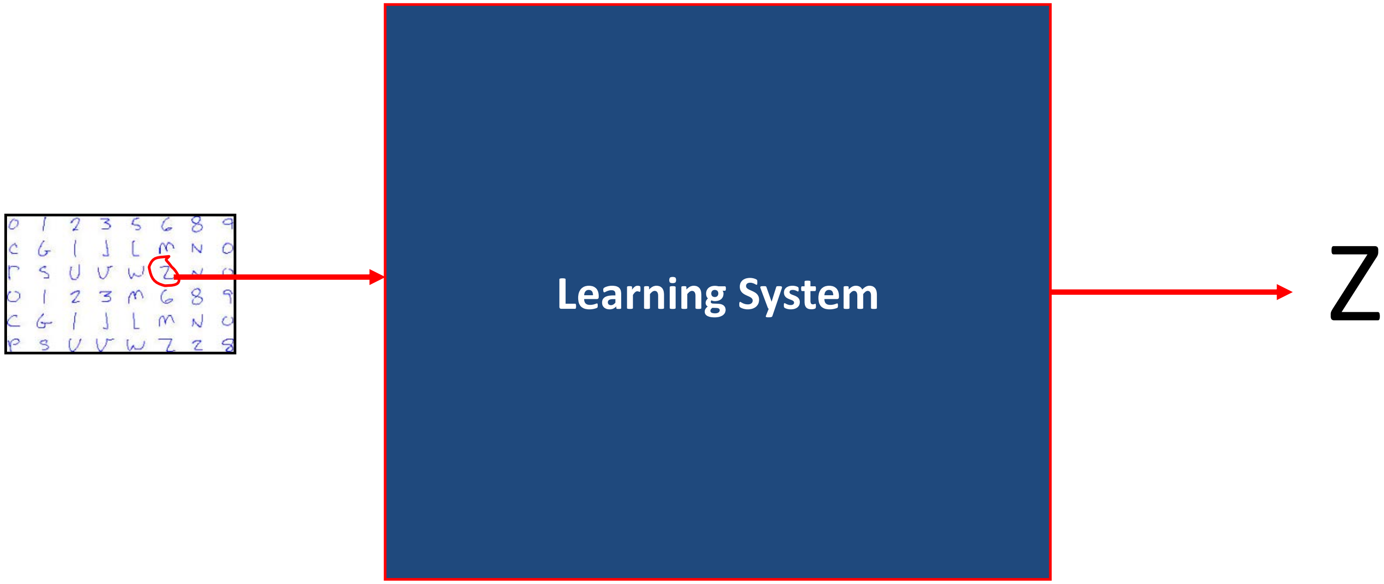
For example, handwriting recognition problem



Design a Learning System

Step 0:

– Lets treat the learning system as a black box

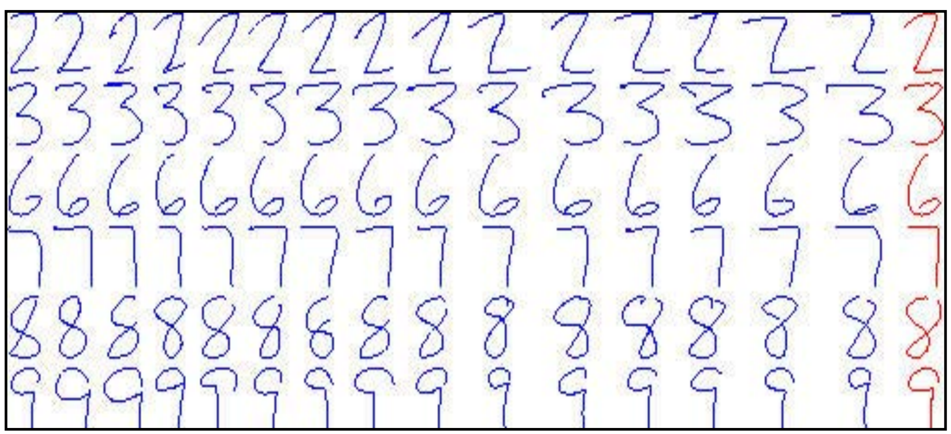


Design a Learning System

Step 1: Collect Training Examples (Experience).

- Without examples, our system will not learn
  - so-called learning from examples

Identify or Class or Label



2

3

6

7

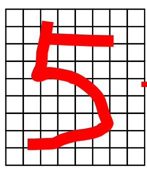
8

9

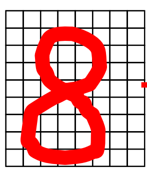
Design a Learning System

Step 2: Representing Experience

– Choose a representation scheme for the experience / examples



(1,1,0,1,1,1,1,1,1,1,0,0,0,0,1,1,1, 1,1,0, …., 1) 64-d Vector



(1,1,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1, 1,1,0, …., 1) 64-d Vector

The sensor input represented by an n-d vector,
called the **feature vector**, **X** = $(x_1, x_2, x_3, \ldots, x_n)$

Design a Learning System

Step 2: Representing Experience

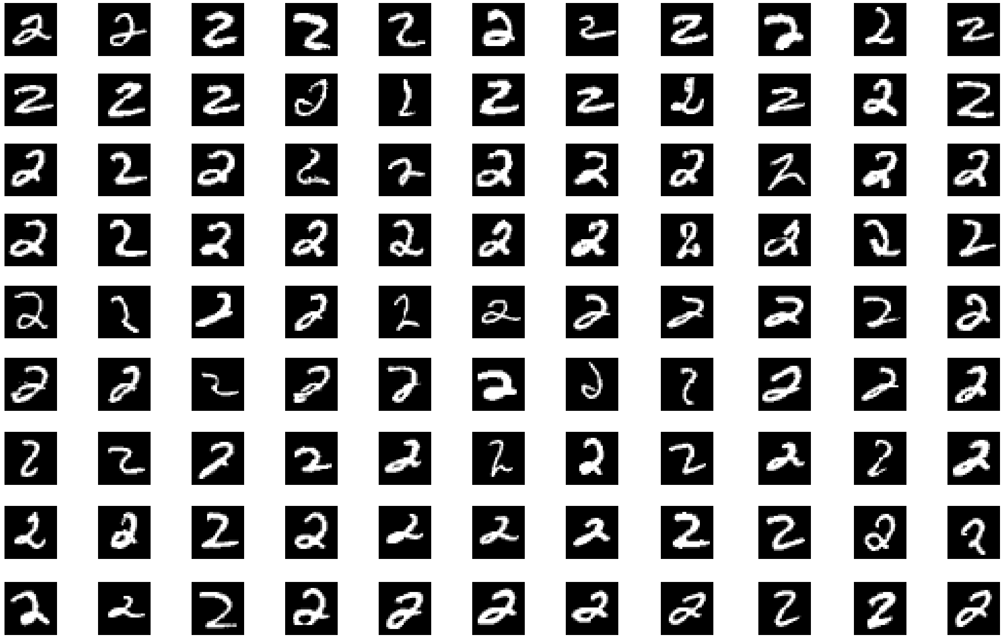
- THE MNIST DATABASE
  http://yann.lecun.com/exdb/mnist/

- The original black and white (bi-level) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. The images were centered in a 28x28 image by computing the center of mass of the pixels, and translating the image so as to position this point at the center of the 28x28 field.

Design a Learning System

Step 2: Representing Experience

- THE MNIST DATABASE
  http://yann.lecun.com/exdb/mnist/



The feature vector of input data is a 784 dimensional vector

Design a Learning System

Step 2: Representing Experience

- Choose a representation scheme for the experience/examples

  - The sensor input represented by an n-d vector, called the feature vector, **X** = (x1, x2, x3, …, xn)

  - To represent the experience, we need to know what **X** is.

  - So we need a corresponding vector **D**, which will record our knowledge (experience) about **X**.

  - The experience **E** is a pair of vectors **E** = (**X**, **D**).

Design a Learning System

Step 2: Representing Experience

- Choose a representation scheme for the experience/examples.

  - The experience **E** is a pair of vectors **E** = (**X**, **D**).

- So, what would **D** be like? There are many possibilities.

Design a Learning System

Step 2: Representing Experience

- So, what would **D** be like? There are many possibilities.
- Assuming our system is to recognise 10 digits only, then D can be a 10-d binary vector; each correspond to one of the digits.

D = (d0, d1, d2, d3, d4, d5, d6, d7, d8, d9)

e.g,

if X is digit 5, then d5=1; all others =0

If X is digit 9, then d9=1; all others =0

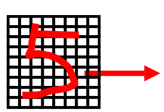
Design a Learning System

Step 2: Representing Experience

– So, what would **D** be like? There are many possibilities.

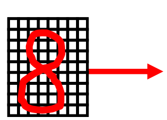
– Assuming our system is to recognise 10 digits only, then D can be a 10-d binary vector; each correspond to one of the digits.

D = (d0, d1, d2, d3, d4, d5, d6, d7, d8, d9)



X = (1,1,0,1,1,1,1,1,1,1,0,0,0,0,1,1,1, 1,1,0, …., 1); 64-d Vector

D= (0,0,0,0,0,1,0,0,0,0)



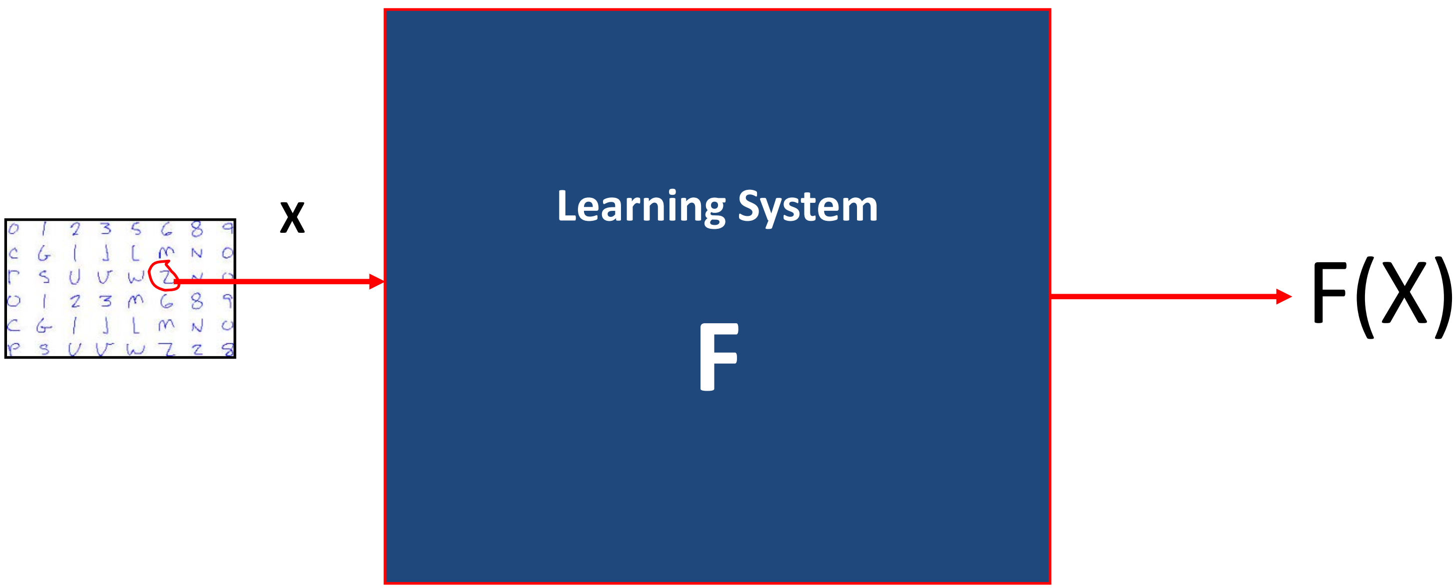
X= (1,1,1,1,1,1,1,1,1,1,0,0,1,1,1,1,1, 1,1,0, …., 1); 64-d Vector

D= (0,0,0,0,0,0,0,0,1,0)

Design a Learning System

Step 3: Choose a Representation for the Black Box

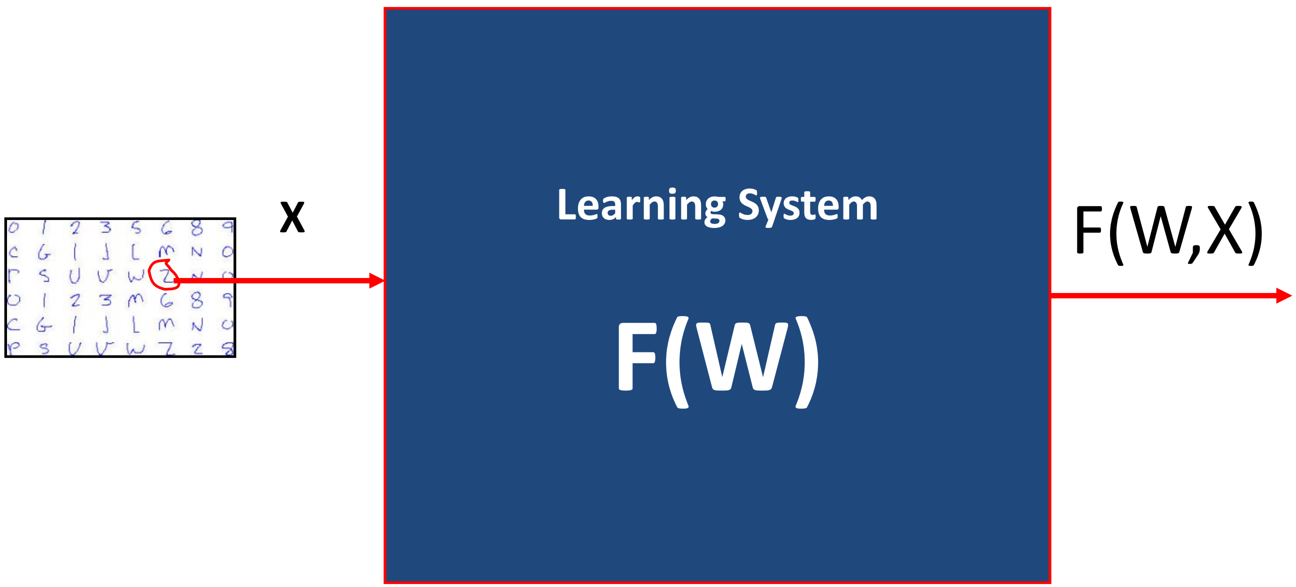
– We need to choose a function F to approximate the block box. For a given X, the value of F will give the classification of X. There are considerable flexibilities in choosing F.



Design a Learning System

Step 3: Choose a Representation for the Black Box

– F will be a function of some adjustable parameters, or weights, W = (w1, w2, w3, …w_N), which the learning algorithm can modify or learn



Design a Learning System

Step 4: Learning/Adjusting the Weights

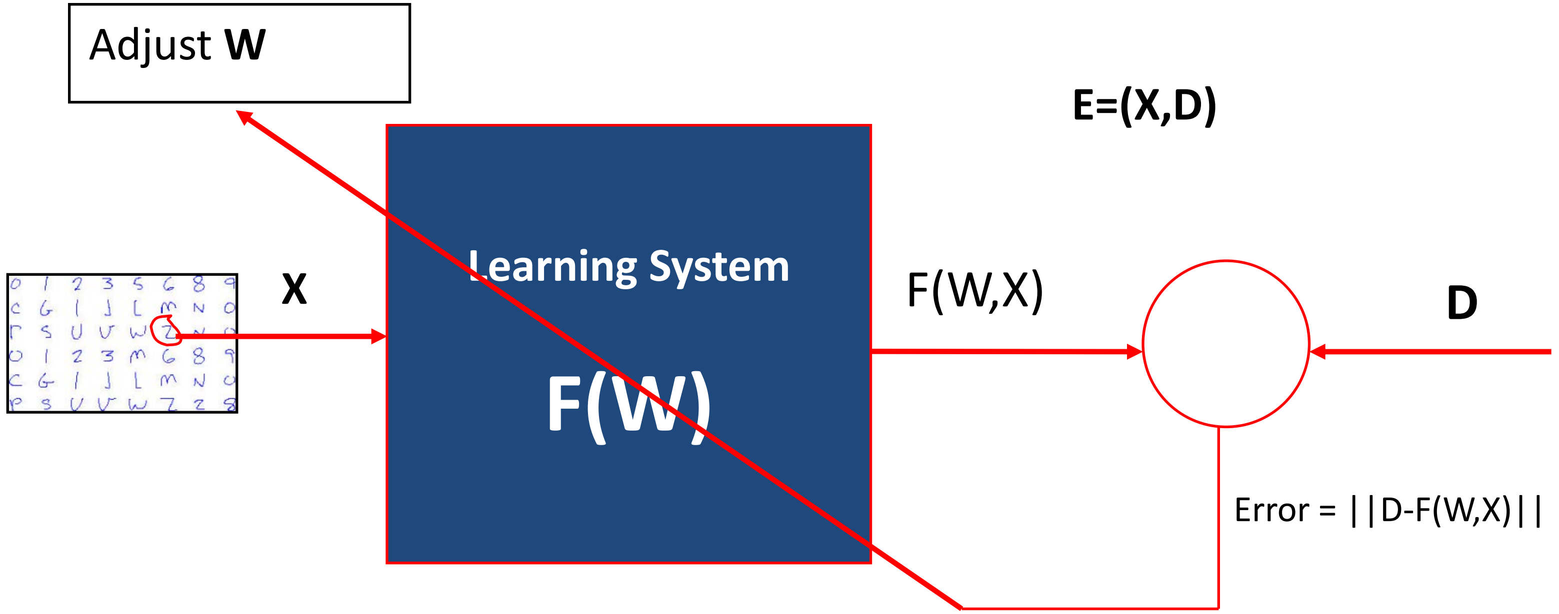
- We need a learning algorithm to adjust the weights such that the experience/prior knowledge from the training data can be learned into the system:

$$E=(X,D)$$

$$F(W,X) = D$$

Design a Learning System

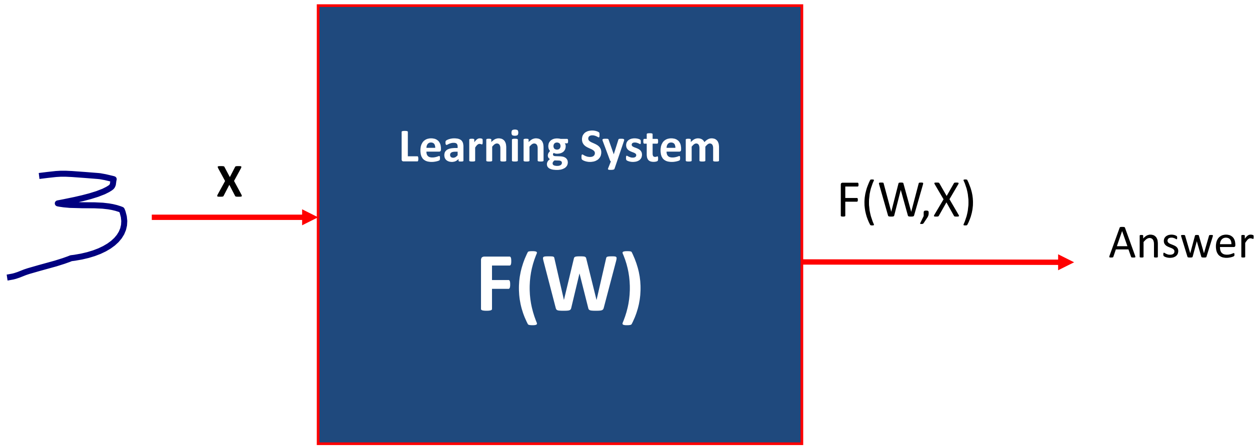
Step 4: Learning/Adjusting the Weights



Design a Learning System

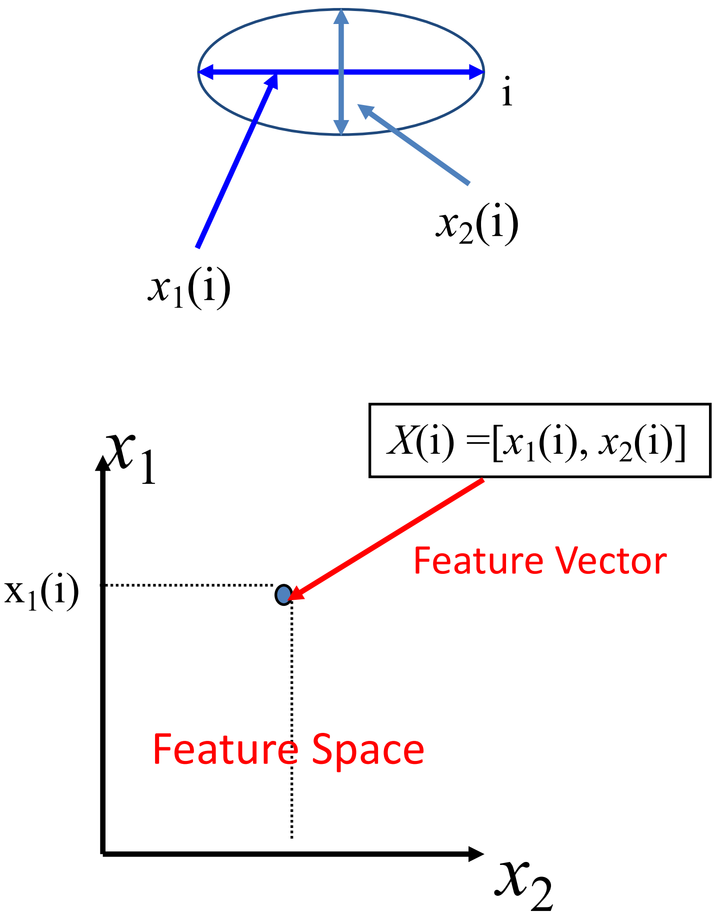
Step 5: Use/Test the System

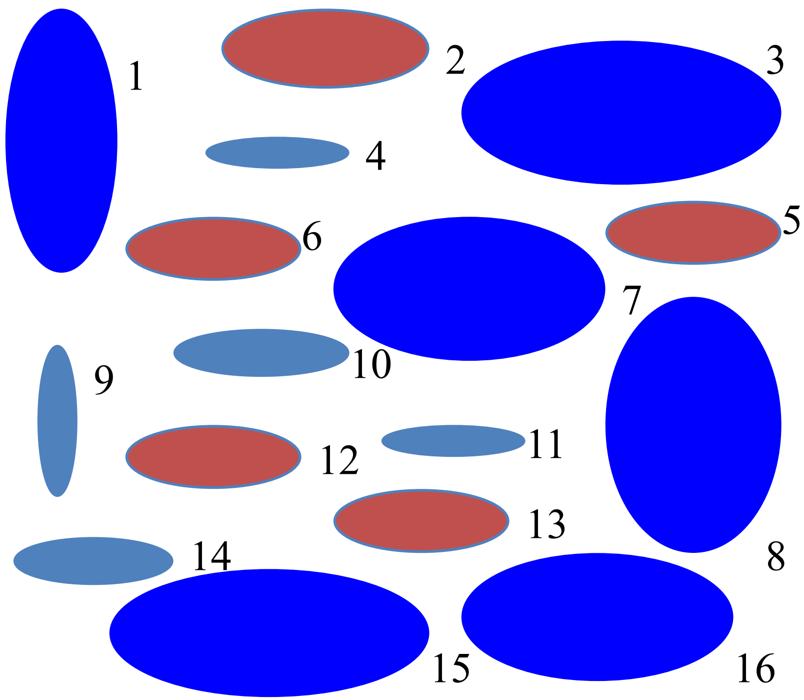
- Once learning is completed, all parameters are fixed. An unknown input **X** is presented to the system, the system computes its answer according to F(**W**,**X**)



Feature Space

Representing real world objects using **feature vectors**

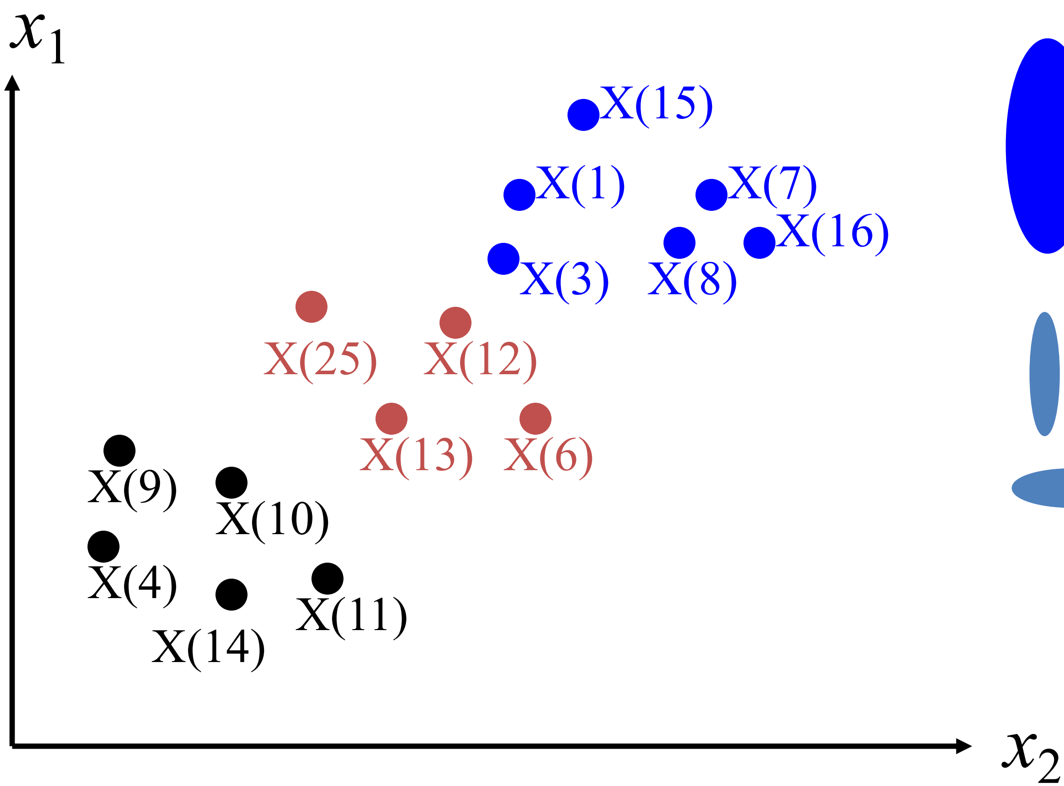


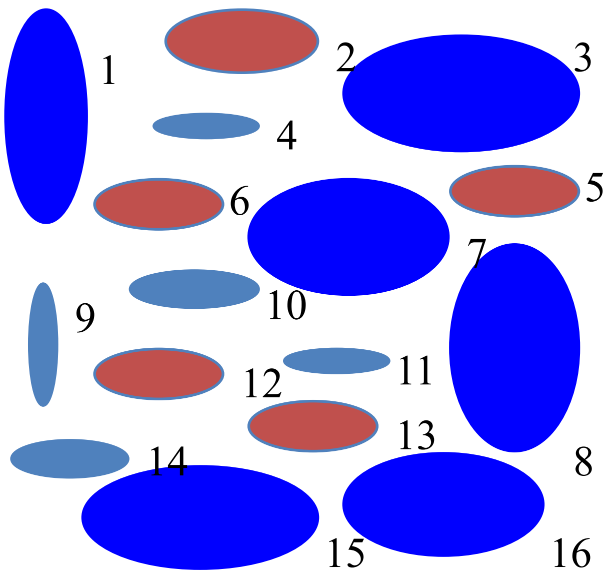


Elliptical blobs (objects)

Feature Space

From Objects to Feature Vectors to Points in the Feature Spaces

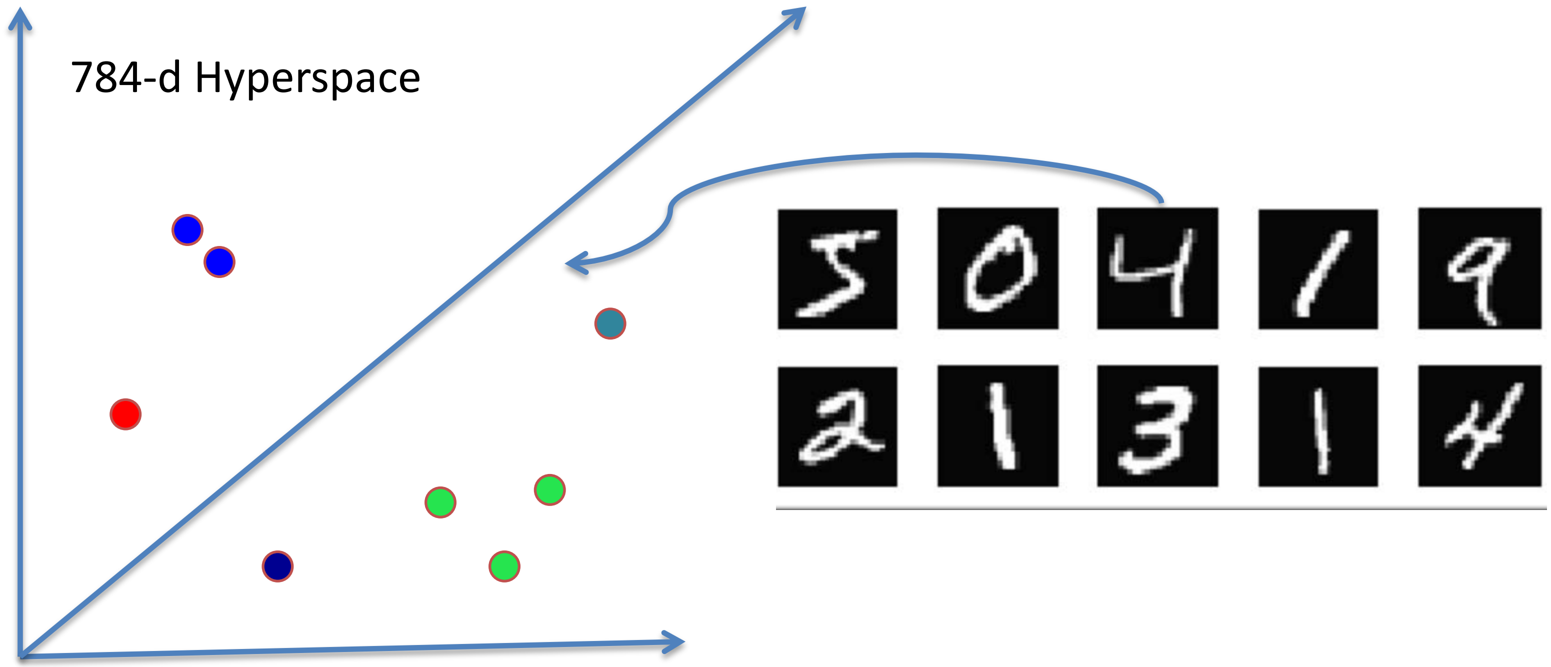




Elliptical blobs (objects)

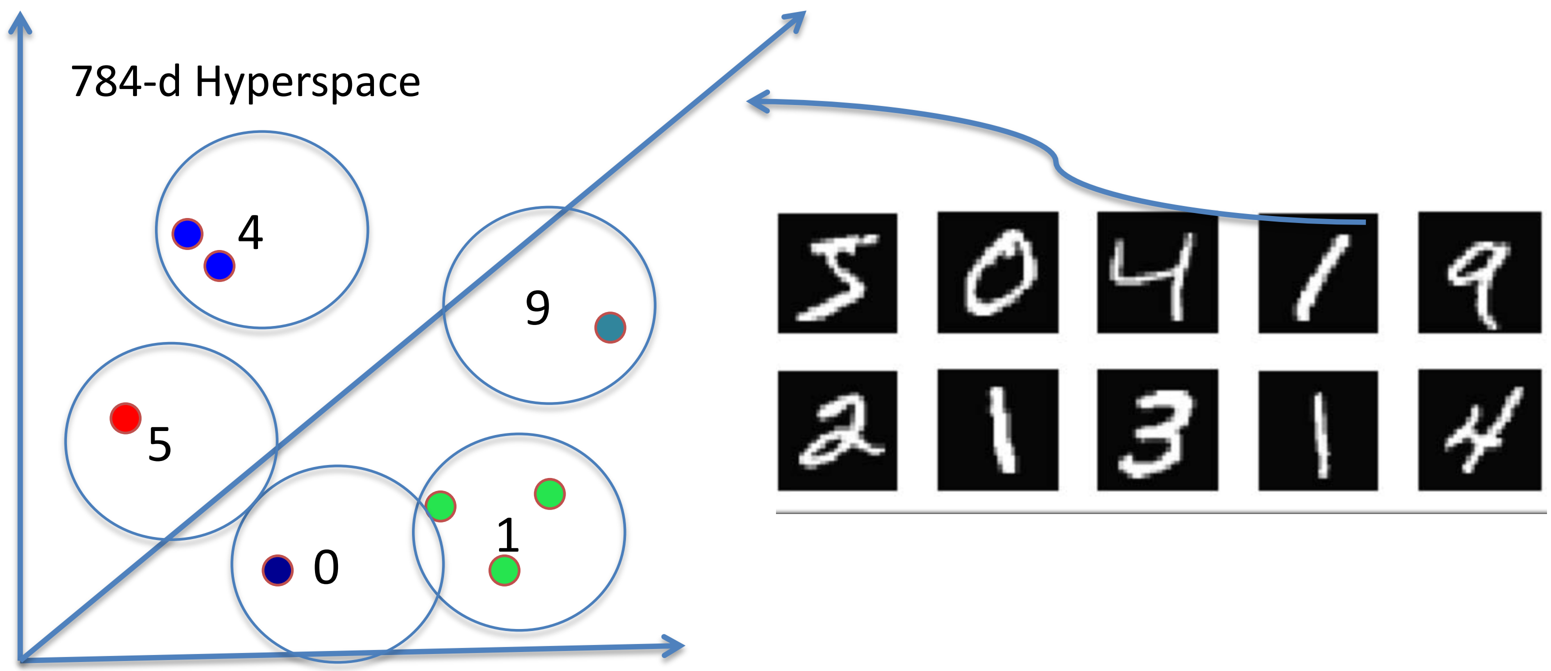
Feature Space

From Objects to Feature Vectors to Points in the Feature Spaces



Feature Space

From Objects to Feature Vectors to Points in the Feature Spaces



Representing General Objects

Feature vectors of

Faces

Cars

Fingerprints

Gestures

Emotions (a smiling face, a sad expression etc)

…

Further Reading

Chapter 1, T. M. Mitchell, Machine Learning, McGraw-Hill International Edition, 1997