Linear Algebra for Computer Science

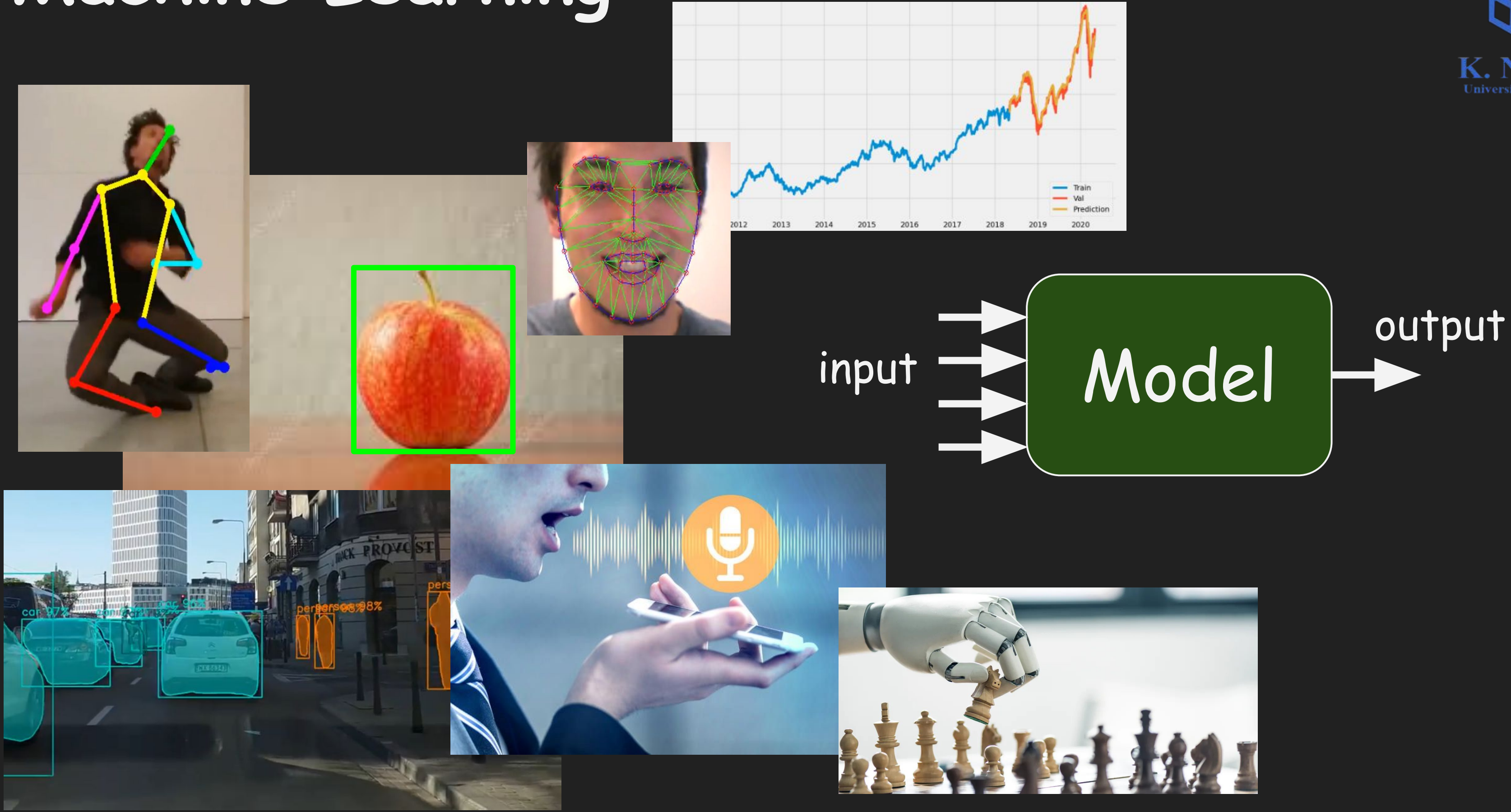
Lecture 30

Introduction to Machine Learning

and learning from data

Machine Learning

K. N. Toosi
University of Technology

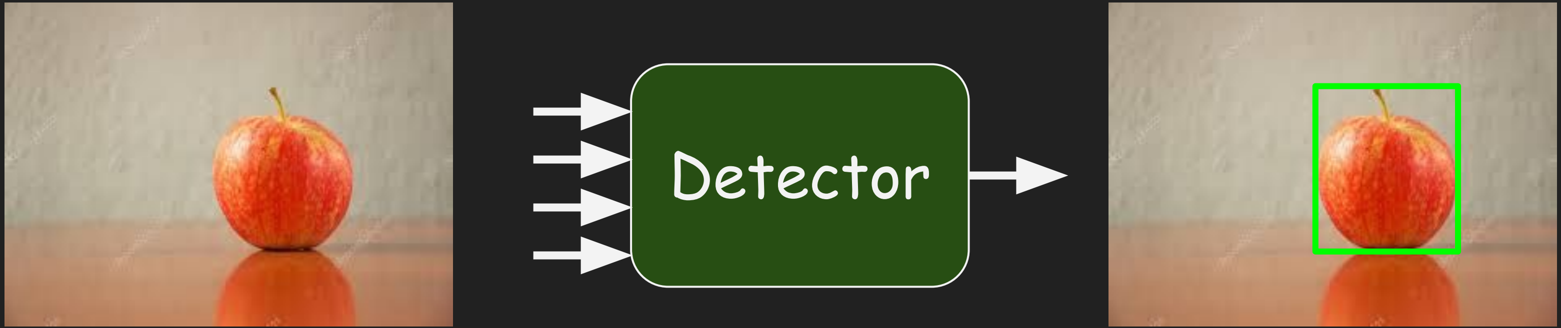


Classification

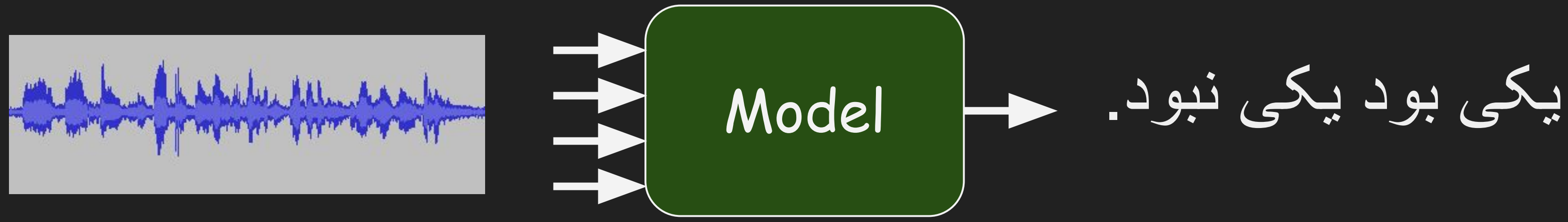
Classification

Object detection

K. N. Toosi
University of Technology

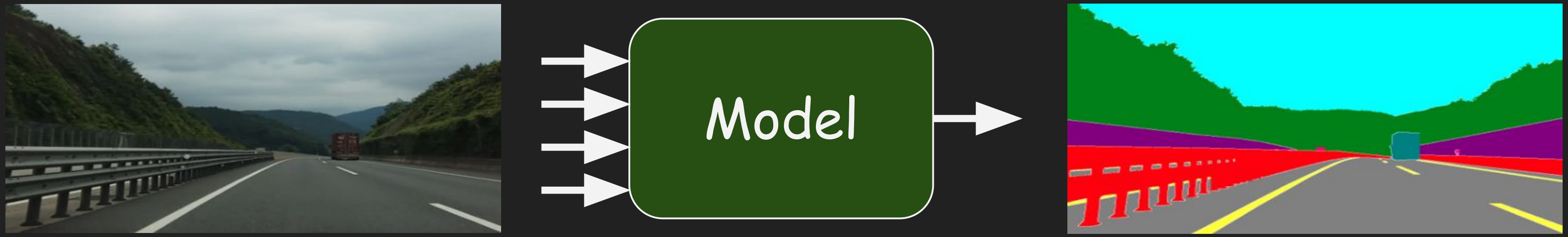


Speech Recognition



Segmentation

K. N. Toosi
University of Technology

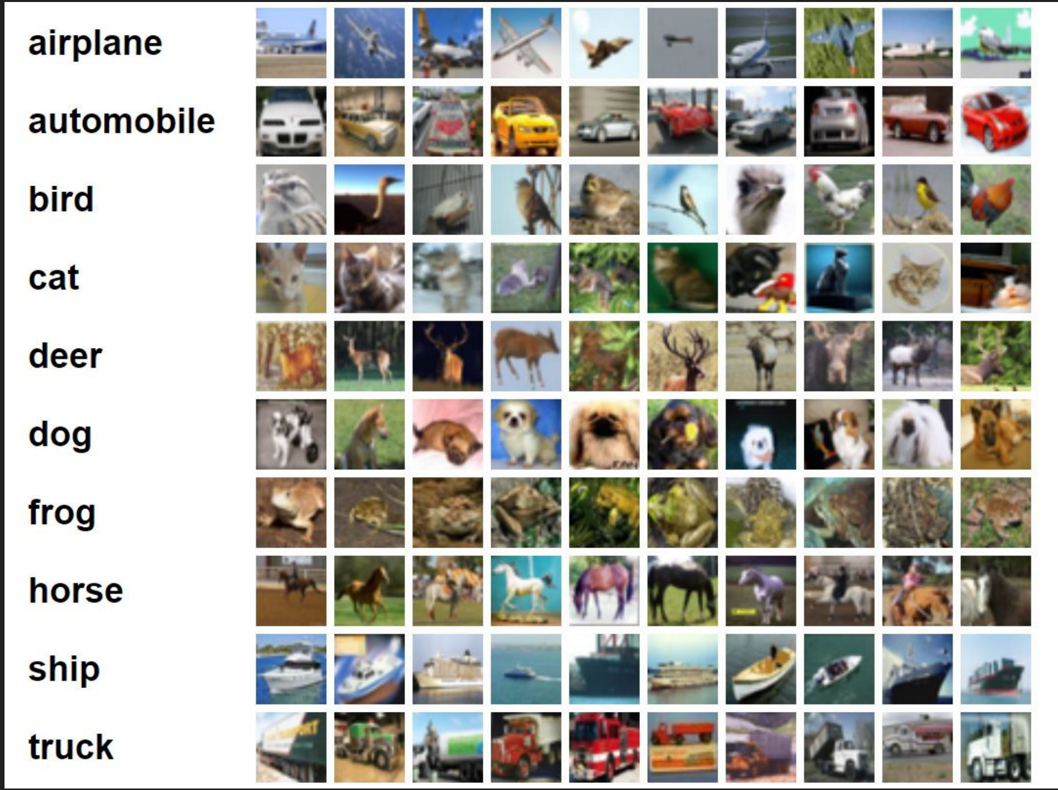


Stock Market Prediction

Learning from data

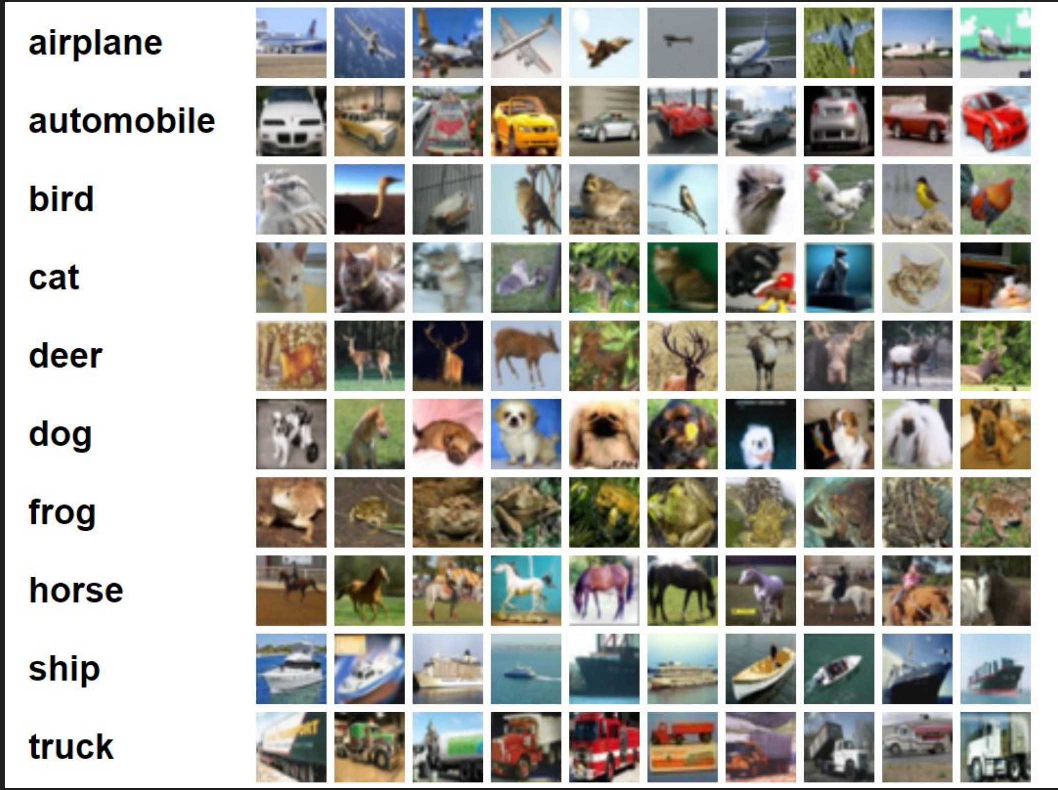
https://www.analyticsvidhya.com/blog/2018/03/comprehensive-collection-deep-learning-datasets/

Supervised Learning



http://seansoleyman.com/effect-of-dataset-size-on-image-classification-accuracy/

Supervised Learning



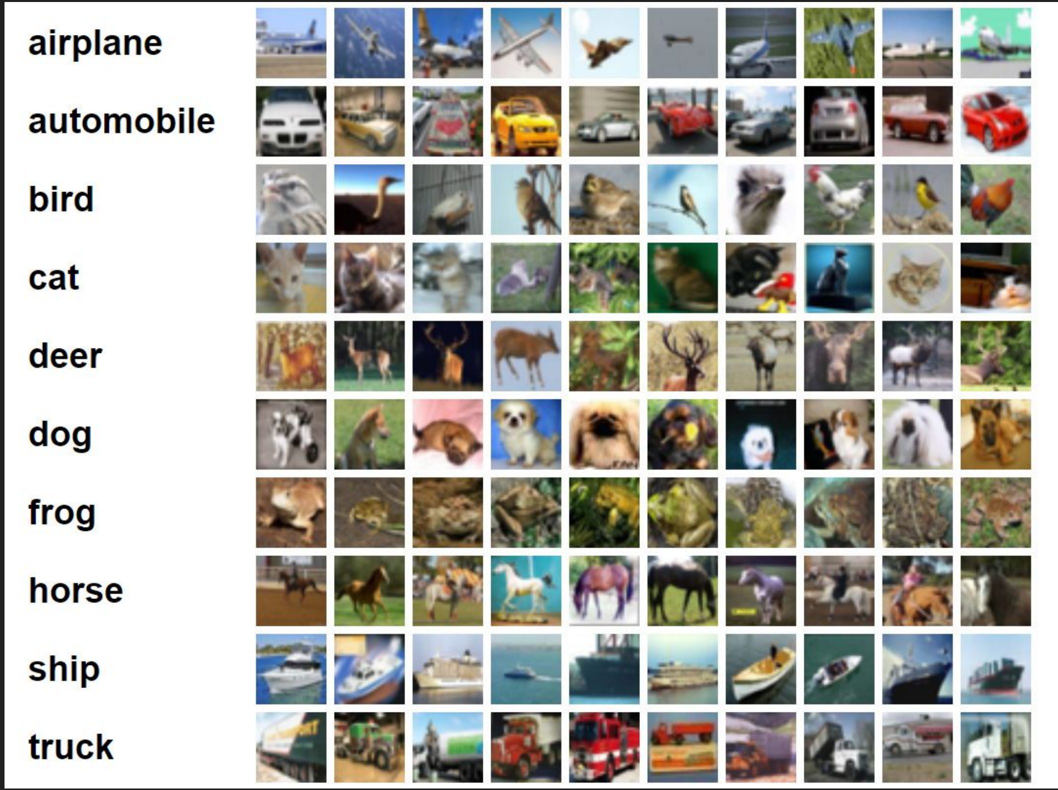
http://seansoleyman.com/effect-of-dataset-size-on-image-classification-accuracy/

Training data:

X_1, y_1
X_2, y_2
X_3, y_3
$\vdots$
X_n, y_n

Supervised Learning

K. N. Toosi
University of Technology

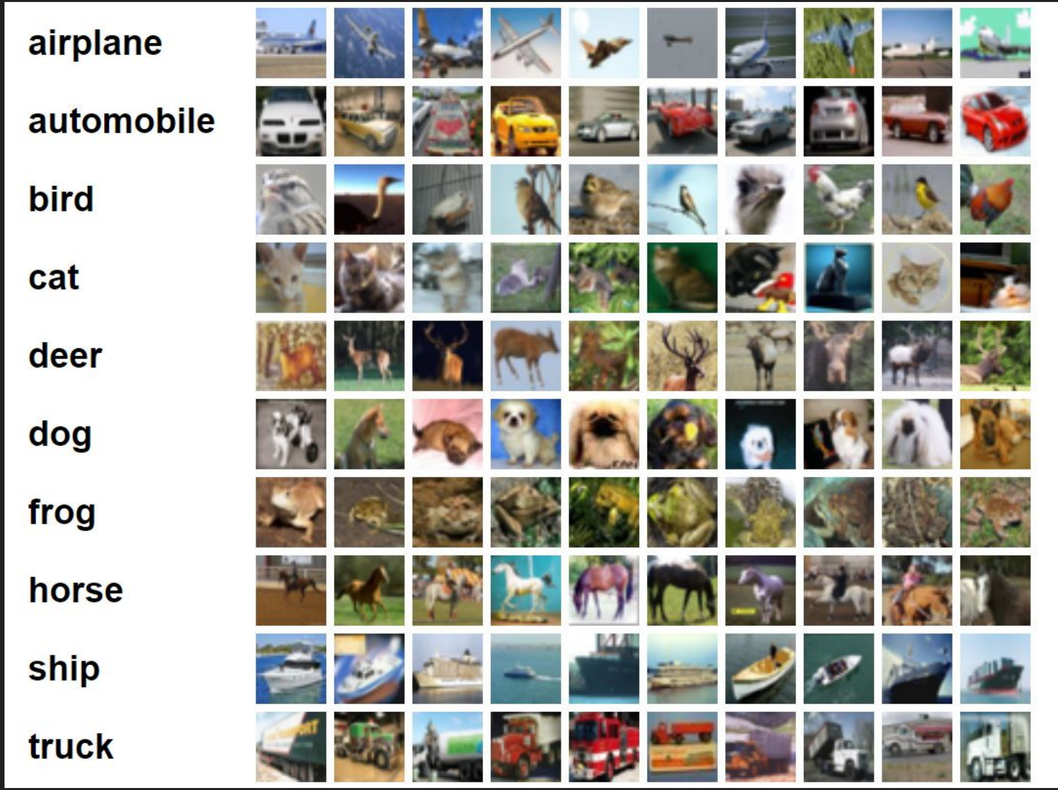


http://seansoleyman.com/effect-of-dataset-size-on-image-classification-accuracy/

Training data:

	Apple
	Apple
	Orange
⋮	
	Orange

Supervised Learning



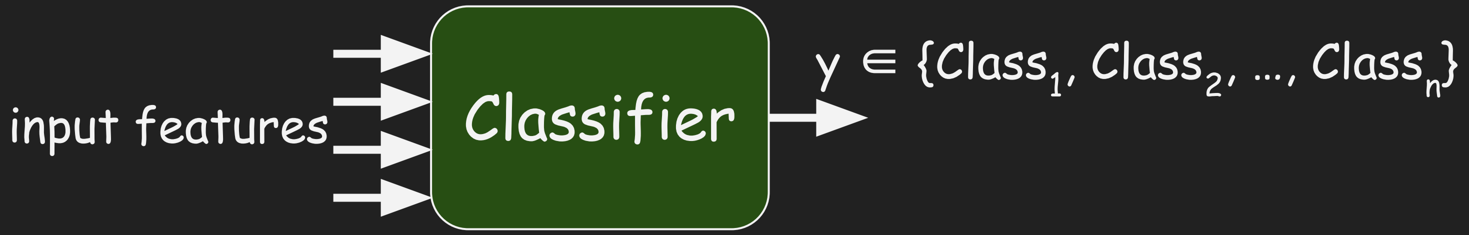
http://seansoleyman.com/effect-of-dataset-size-on-image-classification-accuracy/

Training data:

	0
	0
	1
⋮	
	1

Supervised Learning

Classification



Classification

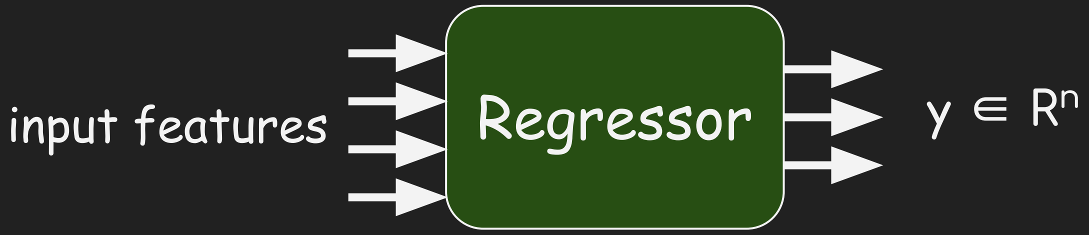
K. N. Toosi
University of Technology

Classification

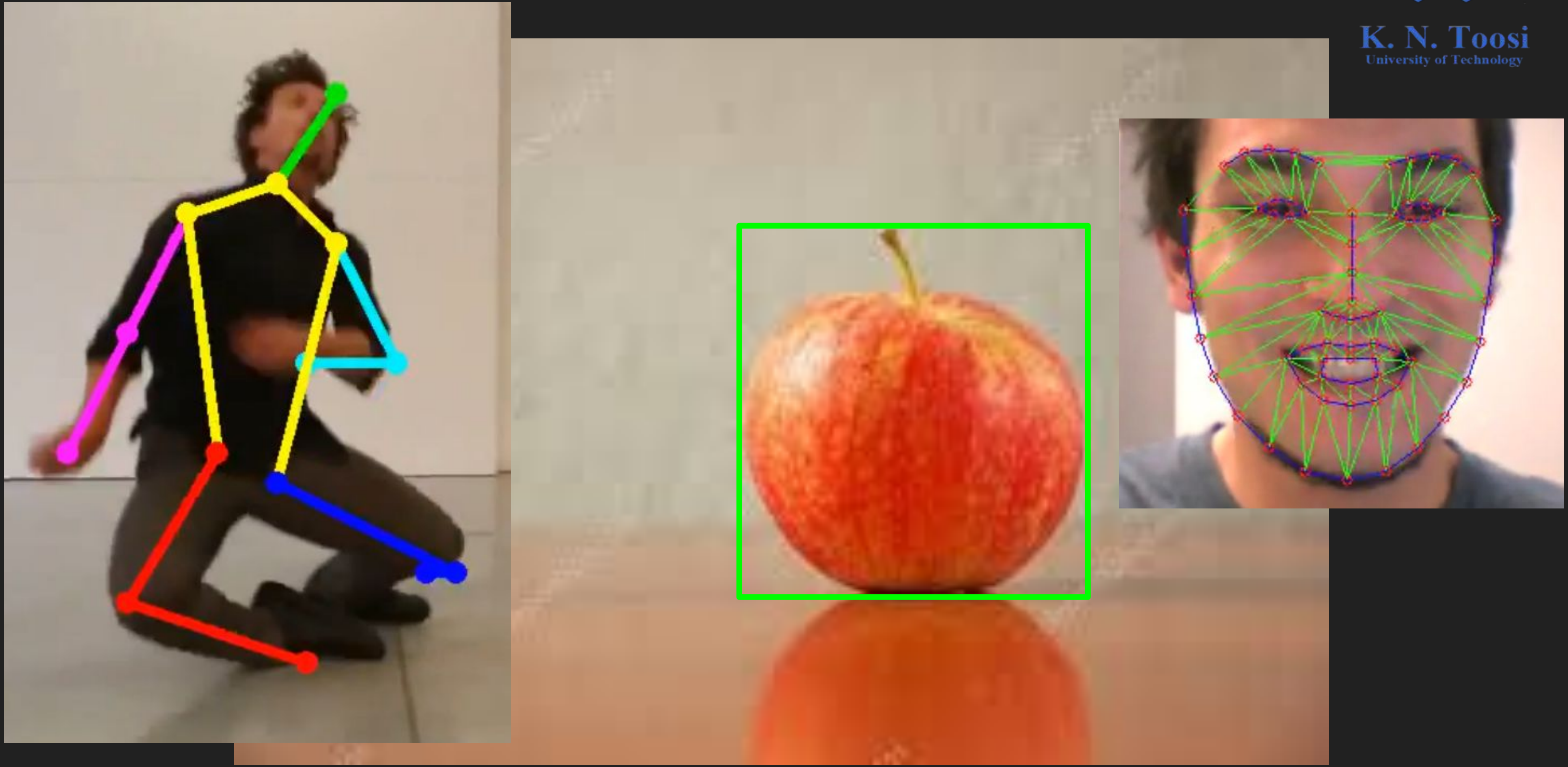
K. N. Toosi
University of Technology

Regression

Regression



Regression



Learnable Models

Learnable Models: Example

K. N. Toosi
University of Technology

Learnable Models: Example

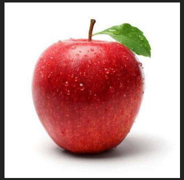
K. N. Toosi
University of Technology

Learnable Models: Input-output map

$$y = f(x)$$

$$f: R^m \rightarrow R^n$$

Learnable Models: Example



I

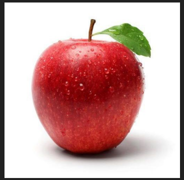
x = I.flatten() → f → y = 0

$$y = f(x)$$

$$f: R^m \rightarrow R^n$$

Learnable Models: Example

K. N. Toosi
University of Technology



I

x = features(I) → f → y = 0

$$y = f(x)$$

$$f: R^m \rightarrow R^n$$

Learnable Models: parameters

K. N. Toosi
University of Technology

$$y = f(\theta, x)$$

θ: model parameters

Learnable Models: parameters

$y = f(x, \theta)$

$f: \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n$

$\mathbb{R}^2 \times \mathbb{R}^6 \to \mathbb{R}^1$

$\begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} \mapsto ax^2 + by^2 + cyx + dx + ey + f$

$f(x, \theta) \in \mathbb{R}^n$

training data $(x_1, y_1); (x_2, y_2) \dots (x_N, y_N)$

$x_1, x_2, \dots, x_n \in \mathbb{R}^m$

$y_1, y_2, \dots, y_n \in \mathbb{R}^N$

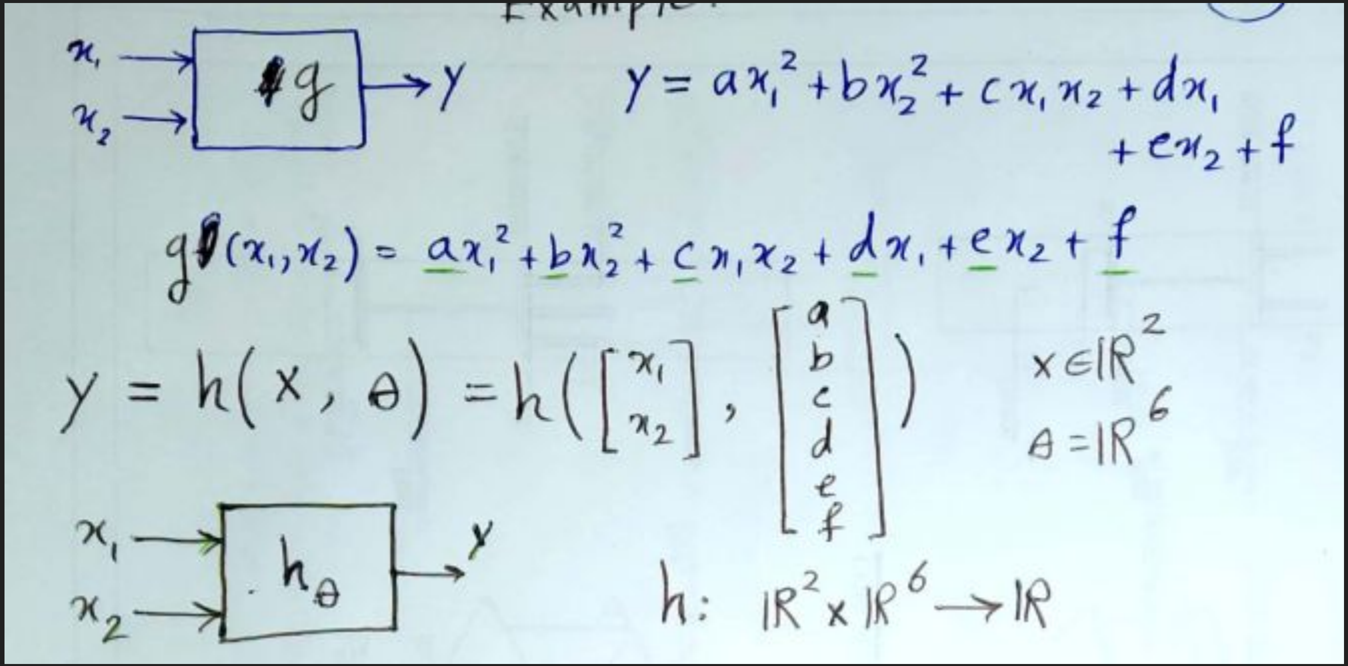
Example

$$C(\theta) = \sum_{i=1}^{N} \| f(x_i; \theta) - y_i \|^2$$

$$= \sum_{i=1}^{N} \left(f(x_i; \theta) - y_i \right)^T \left(f(x_i; \theta) - y_i \right)$$

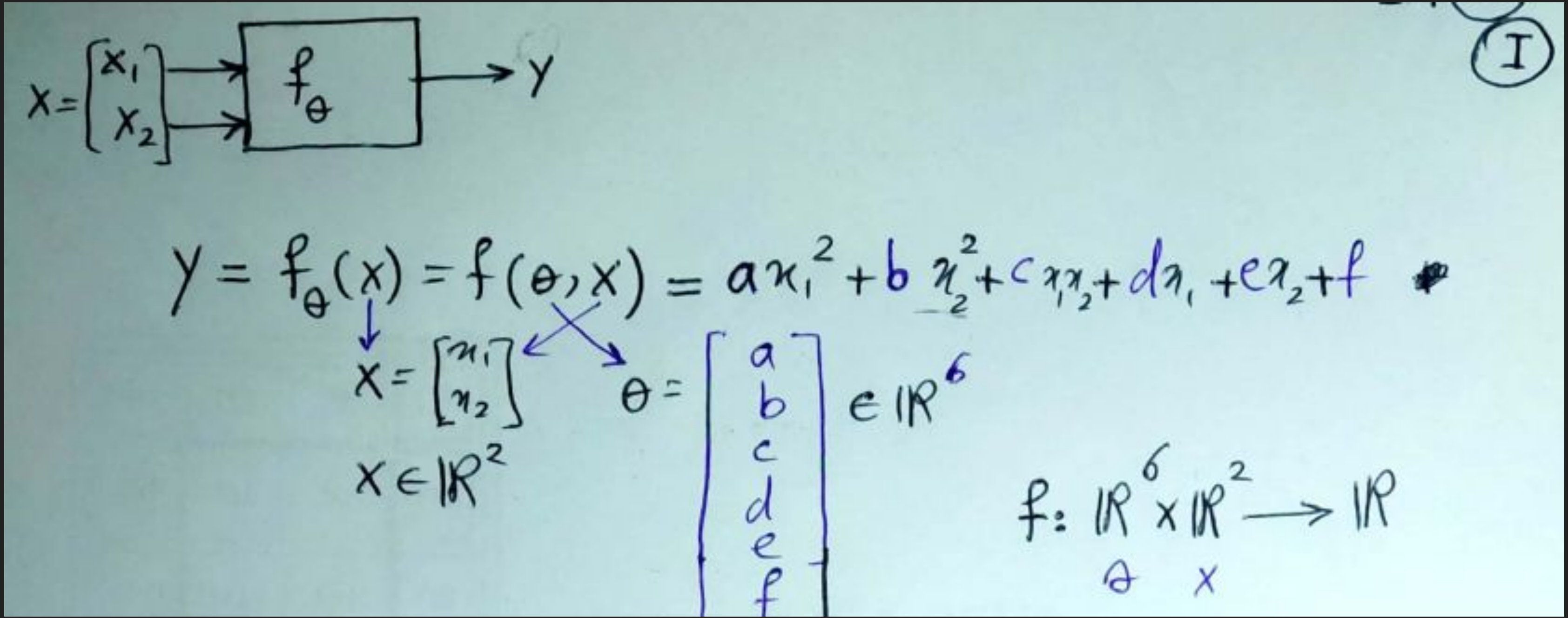
Learnable Models: parameters

K. N. Toosi
University of Technology



Example: A Quadratic Function

K. N. Toosi
University of Technology



Example: A Quadratic Function

K. N. Toosi
University of Technology

$$y = f_\theta(x) = f(\theta, x) = a x_1^2 + b x_2^2 + c x_1 x_2 + d x_1 + e x_2 + f$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \theta = \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} \in \mathbb{R}^6$$

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

$$f: \underset{\theta}{\mathbb{R}^6} \times \underset{x}{\mathbb{R}^2} \longrightarrow \mathbb{R}$$

$$f(\theta, x) = \underbrace{\begin{bmatrix} a & b & c & d & e & f \end{bmatrix}}_{\theta^T} \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \\ x_1 \\ x_2 \\ 1 \end{bmatrix}$$

$$= \theta^T \phi(x) \qquad \phi(x) = \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \\ x_1 \\ x_2 \\ 1 \end{bmatrix}$$

Quadratic Function: Alternative Param

$$f(\theta, x) = a x_1^2 + b x_2^2 + c x_1 x_2 + d x_1 + e x_2 + f$$

$$= \underbrace{\begin{bmatrix} x_1 & x_2 \end{bmatrix}}_{x^T} \underbrace{\begin{bmatrix} a & c/2 \\ c/2 & b \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} d & e \end{bmatrix}}_{v^T} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x} + f$$

$$= x^T A x + v^T x + f \qquad (A \text{ symmetric})$$

$$\theta = (A, v, f)$$

$A \rightarrow$ sym, $v \in \mathbb{R}^2$, $f \in \mathbb{R}$

Learnable Models: parameters

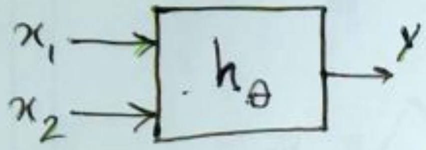
- Parameter Learning:
  - A collection of input-output paris $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$,
  - choose θ such that $y = f(\theta, x)$ is a reasonable output for any input x.

Learnable Models: parameters

K. N. Toosi
University of Technology

$$g_\theta(x_1, x_2) = a x_1^2 + b x_2^2 + c x_1 x_2 + d x_1 + e x_2 + f$$

$$y = h(x, \theta) = h\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}\right) \qquad x \in \mathbb{R}^2, \quad \theta = \mathbb{R}^6$$



$$h: \mathbb{R}^2 \times \mathbb{R}^6 \to \mathbb{R}$$

$$D = \text{traing data} = \{[(x_1^1, x_2^1), y^1], [(x_1^2, x_2^2), y^2], \dots, [(x_1^n, x_2^n), y^n]\}$$

$$D = \{(x^1, y^1), (x^2, y^2), \dots, (x^n, y^n)\}$$

Learning from data

- Parameter Learning:
  - A collection of input-output paris $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$,
  - choose θ such that $y = f(\theta, x)$ is a reasonable output
    - for training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
    - for unseen data (generalization)

Learning from data

K. N. Toosi
University of Technology

- Parameter Learning:
  - A collection of input-output paris $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$,
  - choose θ such that $y = f(\theta, x)$ is a reasonable output
    - for training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
    - for unseen data (generalization)

Learning from data

- Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
  - choose θ such that $f(\theta, x_i)$ is close to y_i

K. N. Toosi
University of Technology

Learning from data: Cost function

- Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
  - choose θ such that $f(\theta, x_i)$ is close to y_i

Cost function

Learning from data: Cost function

- Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
  - choose θ such that $f(\theta, x_i)$ is close to y_i
  - cost function:

$$C(\theta) = \Sigma_{i=1..N}\, d(\, f(\theta, x_i), y_i\,)$$

Learning from data: Cost function

- Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
  - choose θ such that $f(\theta, x_i)$ is close to y_i
  - cost function:

$$C(\theta) = \Sigma_{i=1..N}\, d(\, f(\theta, x_i), y_i \,)$$

data output

Learning from data: Cost function

K. N. Toosi
University of Technology

- Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
  - choose θ such that $f(\theta, x_i)$ is close to y_i
  - cost function:

$$C(\theta) = \Sigma_{i=1..N}\, d(\, f(\theta, x_i), y_i\,)$$

model output given x_i

Learning from data: Cost function

- Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
  - choose θ such that $f(\theta, x_i)$ is close to y_i
  - cost function:

$$C(\theta) = \Sigma_{i=1..n}\ d(\ f(\theta, x_i), y_i\ )$$

d → distance

Learning from data: Cost function

- Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
  - choose θ such that $f(\theta, x_i)$ is close to y_i
  - cost function:

$$C(\theta) = \Sigma_{i=1..n} \| f(\theta, x_i) - y_i \|^2$$

distance

Learning from data: Cost function

- Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
  - choose θ such that $f(\theta, x_i)$ is close to y_i
  - cost function:

$$C(\theta) = \Sigma_{i=1..n}\, d(\, f(\theta, x_i), y_i\,)$$

Learning from data: Cost function

$$x \in R^m \rightarrow f_\theta \rightarrow y = f(\theta, x)$$

- Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
  - choose θ such that $f(\theta, x_i)$ is close to y_i
  - cost function:

$$C(\theta) = \Sigma_{i=1..n}\, d(\, f(\theta, x_i), y_i\,)$$

choose θ such that $C(\theta)$ is small

Learning from data: Cost function

K. N. Toosi
University of Technology

- Training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$
  - choose θ such that $f(\theta, x_i)$ is close to y_i
  - cost function:

$$C(\theta) = \Sigma_{i=1..n}\, d(\, f(\theta, x_i), y_i\,)$$

$$\theta^* = \text{argmin}_\theta\, C(\theta)$$