

# Mathematics for AI

## Lecture 14

Noisy Homogeneous Equations, Affine spaces,  
affine maps, nonlinear functions, linearization and  
derivatives

# Eigen-decomposition and optimization



$A$  symmetric MA

$$\max_x \frac{x^T A x}{x^T x} = \max_x \frac{x^T A x}{\|x\|^2} \quad x^T x = 1$$
$$\max \quad x^T A x$$
$$\min \quad x^T A x \quad \text{subject to } \|x\| = 1$$

$\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$   
 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$   
 $\lambda_{\max} \quad \lambda_{\min}$

$$\lambda_{\min} \leq \frac{x^T A x}{\|x\|^2} \leq \lambda_{\max}$$
$$A = \sum_{i=1}^n \lambda_i v_i v_i^T$$
$$\lambda_{\max} = \max_x \frac{x^T A x}{x^T x} \quad v_{\max} = \operatorname{argmax}_x \frac{x^T A x}{x^T x}$$
$$\lambda_{\min} = \min_x \frac{x^T A x}{x^T x}$$

# SVD and optimization



$$\max_x \|Ax\| \quad \text{subject to } \|x\|=1$$

$$\max_x \frac{\|Ax\|}{\|x\|} = ?$$

$$\frac{v_i^T A v_i}{v_i^T v_i} = \frac{v_i^T (\lambda v_i)}{1}$$
$$\lambda_i v_i^T v_i = \lambda_i$$

$$\max_x \frac{\|Ax\|^2}{\|x\|^2} = \frac{(Ax)^T (Ax)}{x^T x} = \frac{x^T (A^T A) x}{x^T x}$$

$$\max_x \frac{\|Ax\|^2}{\|x\|^2} = \lambda_{\max}(A^T A) = \sigma_{\max}^2 = \sigma_1^2$$

$$\max_x \frac{\|Ax\|}{\|x\|} = \sigma_1 = \sigma_{\max}$$

$$\operatorname{argmax}_x \frac{\|Ax\|}{\|x\|} = \vec{v}_1 \quad A = U \Sigma \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix}$$

# SVD and optimization



$$\begin{aligned} \min_x \|Ax\| \quad \text{s.t.} \quad \|x\|=1 &= \sigma_{\min} \\ \operatorname{argmin}_x \|Ax\| \quad \text{s.t.} \quad \|x\|=1 &= V_{\min} \\ A = U \begin{bmatrix} \sigma_{\max} & & & \\ & \sigma_2 & & \\ & & \sigma_{\min} & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \\ \vdots \\ V_{\min}^T \end{bmatrix} \end{aligned}$$

# Remember: homogeneous equations



K. N. Toosi  
University of Technology

$$A\vec{x} = 0$$

if  $\vec{x}$  is a solution  ~~$\vec{x}$~~  so is  $\alpha\vec{x}$   
only the direction/orientation of  $\vec{x}$   
matters.

1D on the same

# Remember: homogeneous equations



$$A\vec{x} = 0$$

In many applications  $A$  has a 1D null space

~~$A \in \mathbb{R}^{m \times n}$~~

$\Rightarrow$   $A$  has  $n-1$   
independent columns -  
 $\text{rank}(A) = n-1$

$X = \text{null space}(A)$

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = 0$$

$$a_1^T x = 0$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = 0$$

$$a_2^T x = 0$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = 0$$

$$a_m^T x = 0$$

$$\begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{bmatrix} \vec{x} = 0$$

$A$  is of rank  $n-1$

$m \gg n$

Due to noise/computation errors

# Noisy homogeneous equations



In ~~practice~~ practice  $\text{rank}(A) = n$   $A \in \mathbb{R}^{m \times 3}$  MA14 (III)

In theory = 
$$[A] = [U] \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ v_3^T \end{bmatrix}$$

In practice = 
$$[A] = [U] \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ v_3^T \end{bmatrix}$$

in practice

$$\|Ax\| > 0$$

$\sigma_3$  small

Find  $x$  such that  $Ax=0$  theory



$$x^* = \underset{x}{\text{argmin}} \|Ax\| \quad \text{s.t. } \|x\|=1$$

# Noisy homogeneous equations



in practice

$$\|Ax\| > 0$$

$\sigma_3$  small

Find  $x$  such that  $Ax = 0$  } theory



$$x^* = \operatorname{argmin}_x \|Ax\| \quad \text{s.t. } \|x\| = 1$$

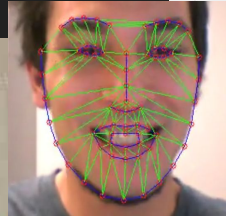
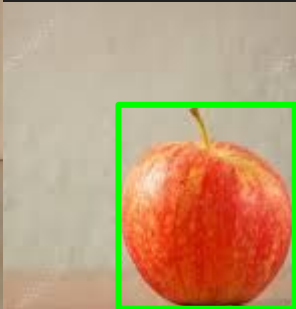
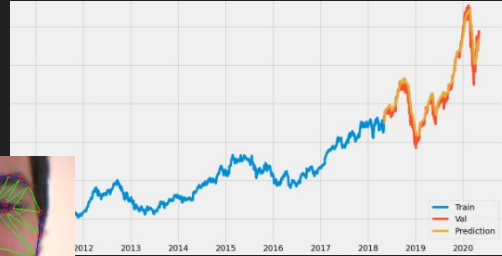
$x = v_{\min} \Rightarrow$  right singular vector corresponding to  $\sigma_{\min}$



# Machine Learning



K. N. Toosi  
University of Technology

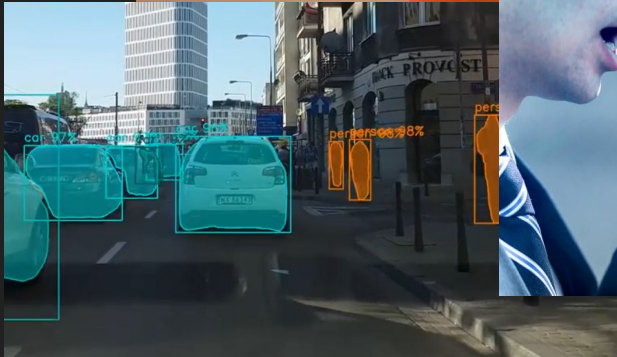


input



Model

output



# Classification



K. N. Toosi  
University of Technology



# Classification



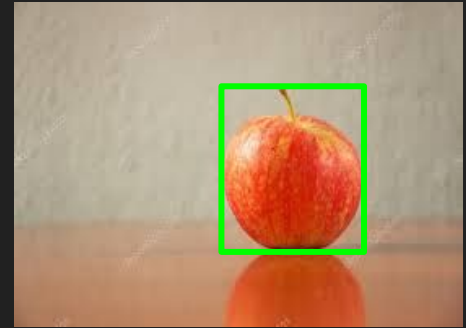
K. N. Toosi  
University of Technology



# Object detection



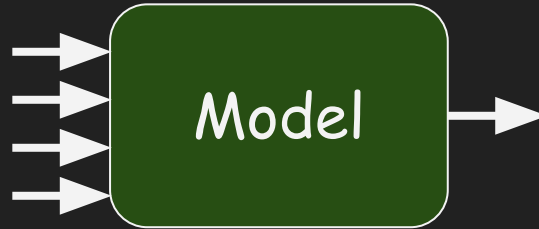
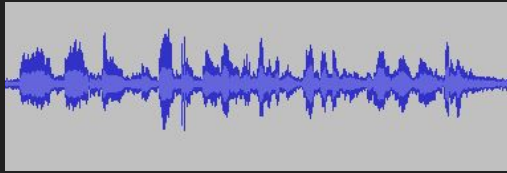
K. N. Toosi  
University of Technology



# Speech Recognition



K. N. Toosi  
University of Technology

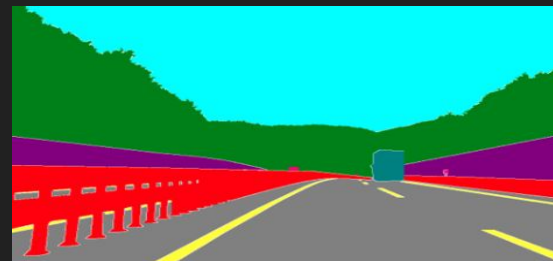
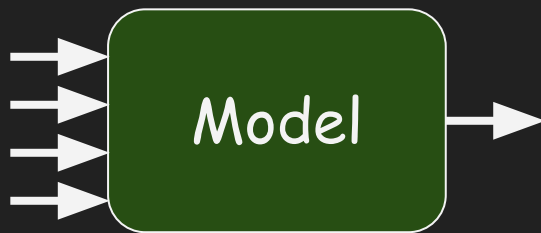


یکی بود یکی نبود.

# Segmentation



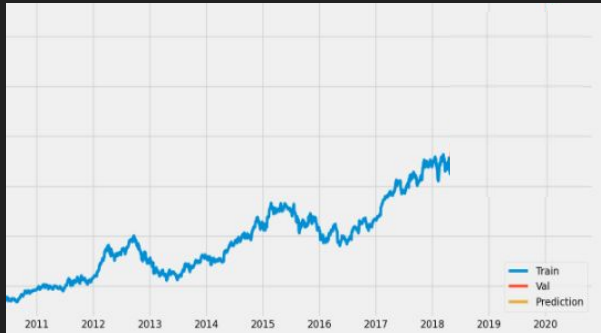
K. N. Toosi  
University of Technology



# Stock Market Prediction



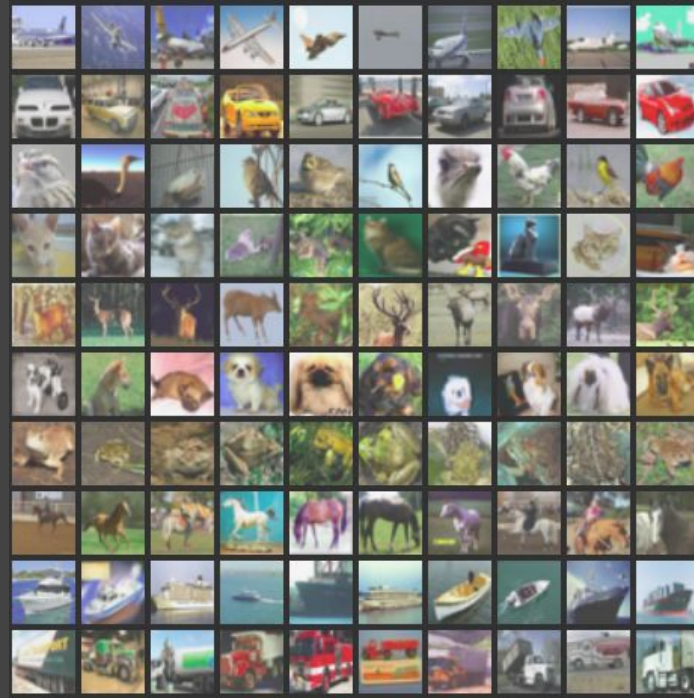
K. N. Toosi  
University of Technology



# Learning from data



K. N. Toosi  
University of Technology

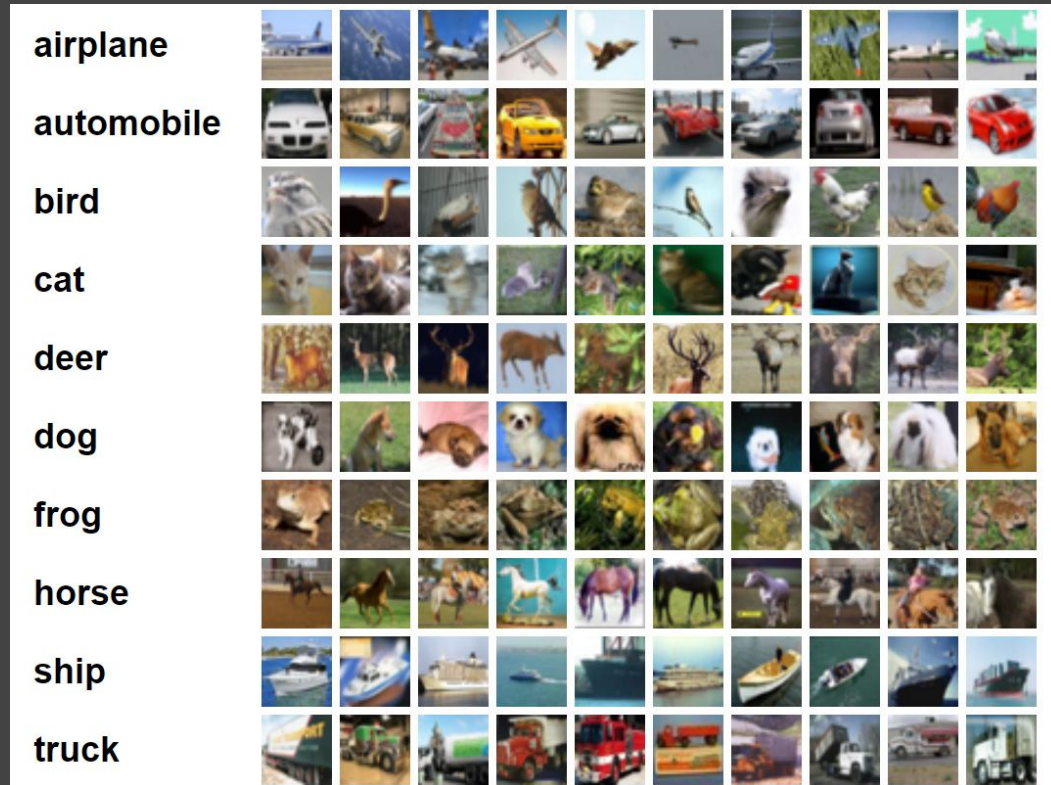




# Supervised Learning



K. N. Toosi  
University of Technology

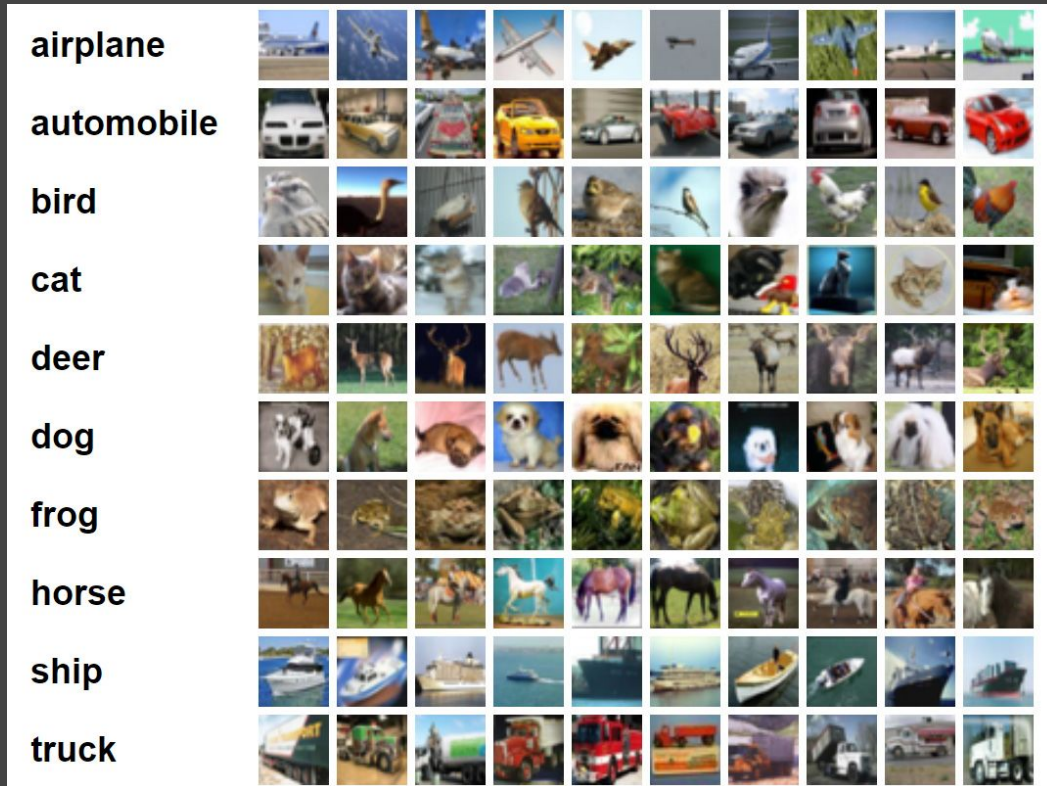


<http://seansoleyman.com/effect-of-dataset-size-on-image-classification-accuracy/>

# Supervised Learning



K. N. Toosi  
University of Technology



Training data:

$X_1, y_1$

$X_2, y_2$

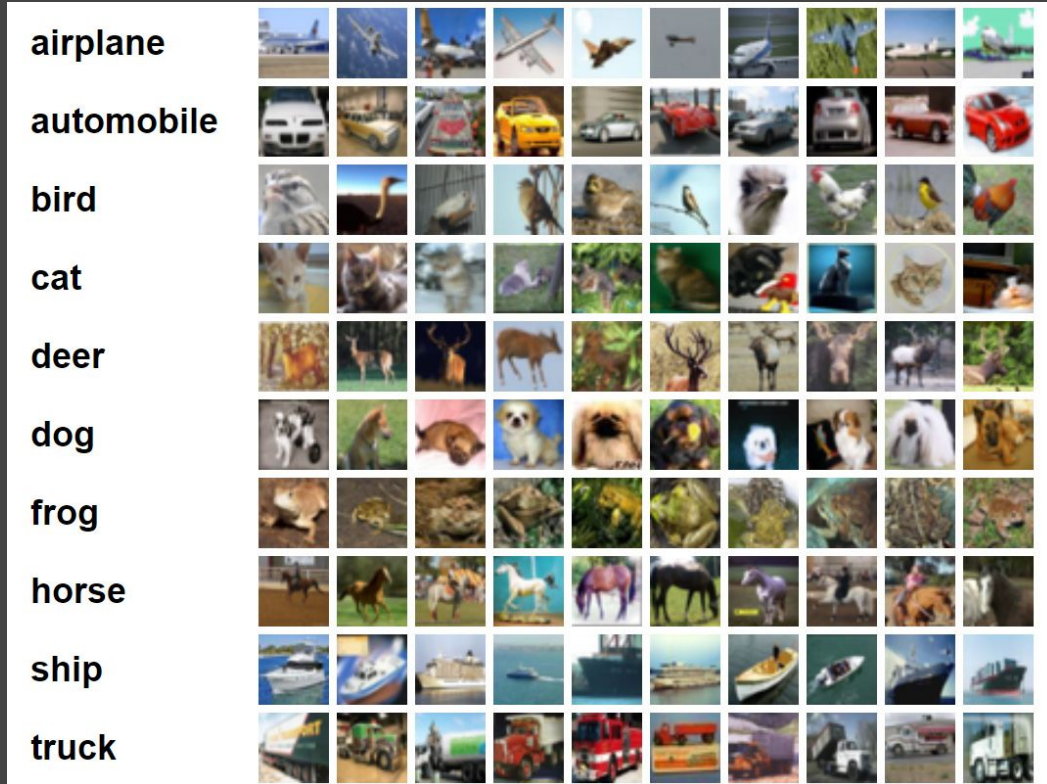
$X_3, y_3$

$\vdots$   
 $X_n, y_n$

# Supervised Learning



K. N. Toosi  
University of Technology



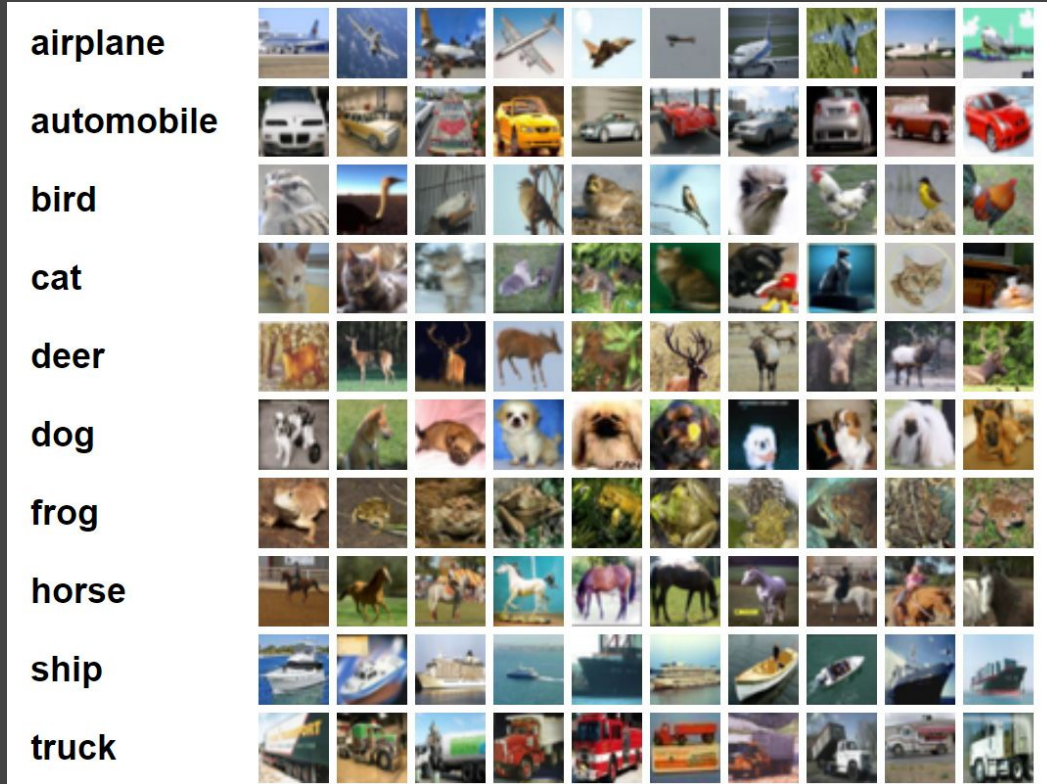
Training data:

	<b>Apple</b>
	<b>Apple</b>
	<b>Orange</b>
	<b>Orange</b>

# Supervised Learning



K. N. Toosi  
University of Technology



Training data:

	0
	0
	1
⋮	
	1

# Supervised Learning



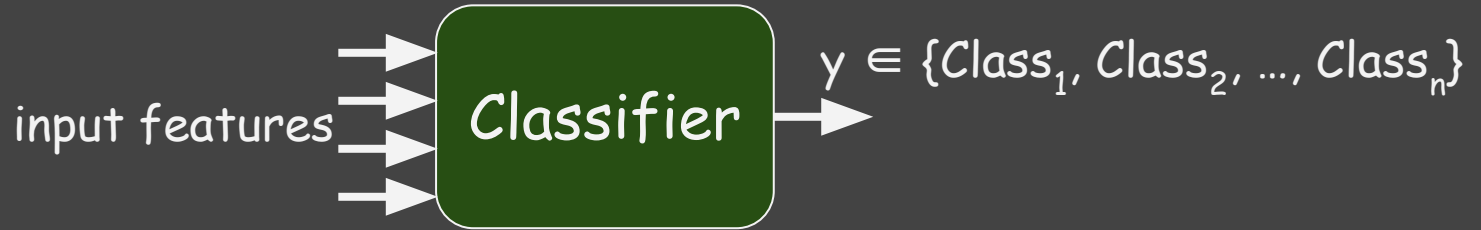
K. N. Toosi  
University of Technology



# Classification



K. N. Toosi  
University of Technology



# Classification



K. N. Toosi  
University of Technology



# Classification



K. N. Toosi  
University of Technology





# Regression



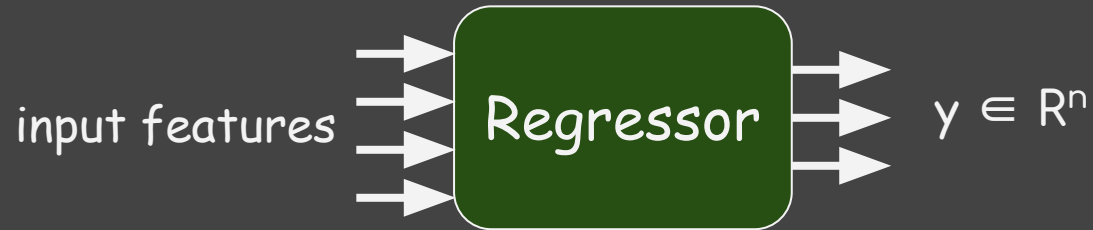
K. N. Toosi  
University of Technology



# Regression



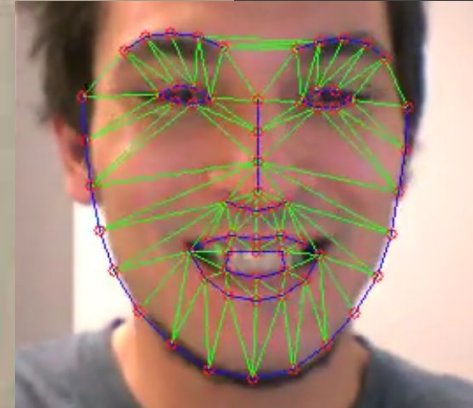
K. N. Toosi  
University of Technology



# Regression



K. N. Toosi  
University of Technology



# Learnable Models



K. N. Toosi  
University of Technology



# Learnable Models: Example



K. N. Toosi  
University of Technology



# Learnable Models: Example



K. N. Toosi  
University of Technology



# Learnable Models: Input-output map



K. N. Toosi  
University of Technology



$$y = f(x)$$

$$f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

# Learnable Models: Example

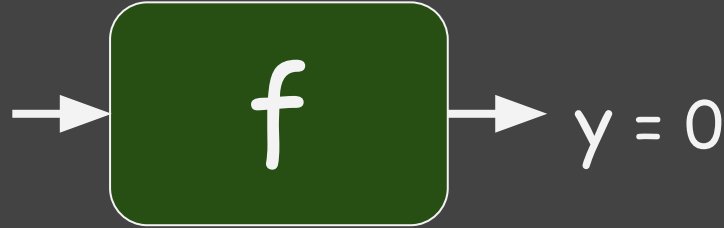


K. N. Toosi  
University of Technology



I

$x =$   
I.flatten()



$$y = f(x)$$

$$f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$



# Learnable Models: Example

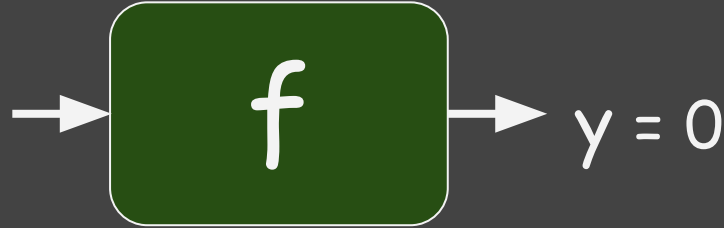


K. N. Toosi  
University of Technology



I

$x =$   
 $\text{features}(I)$



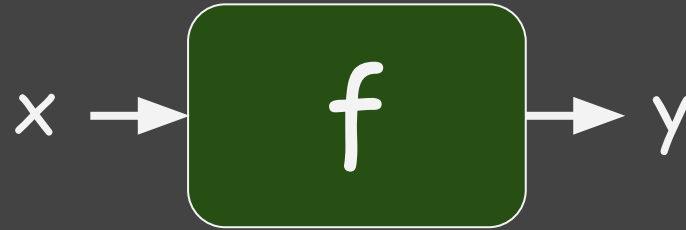
$$y = f(x)$$

$$f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

# Learnable Models: Example



K. N. Toosi  
University of Technology



$$y = f(x)$$

$$f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

# Learnable Models: Example



K. N. Toosi  
University of Technology



What about a linear function  $y = f(x) = A x$ ?

$$f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

# Learnable Models: Example



K. N. Toosi  
University of Technology



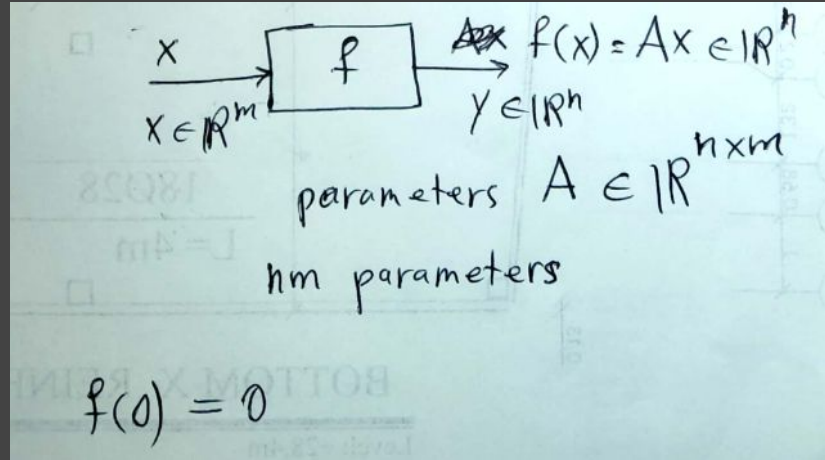
What about a linear function  $y = f(x) = A x$ ?

$$f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

# Learnable Models: Example



K. N. Toosi  
University of Technology



What about a linear function  $y = f(x) = A x$ ?

$$f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

# Learnable Models: Example



K. N. Toosi  
University of Technology



$$y = f(x), f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

How to make  $f$  learnable?

# Learnable Models: parameters



K. N. Toosi  
University of Technology



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

$\theta$ : model parameters

# Learnable Models: parameters



①  
23

$$f(x, y) = ax^2 + by^2 + cxy + dx + ey + f$$

$$f(x, \theta) \quad \theta = (a, b, c, d, e, f)^T$$
$$f\left(\begin{bmatrix} x \\ y \end{bmatrix}, \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}\right) = ax^2 + by^2 + cxy + dx + ey + f$$

training data  $(x_1, y_1, t_1), (x_2, y_2, t_2), (x_3, y_3, t_3), \dots$

$$\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}; t_1\right), \left(\begin{bmatrix} x_2 \\ y_2 \end{bmatrix}; t_2\right), \dots, \left(\begin{bmatrix} x_n \\ y_n \end{bmatrix}; t_n\right)$$

$$C(\theta, D) = C(\theta)$$



# Learnable Models: parameters



K. N. Toosi  
University of Technology



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

# Learning from data



K. N. Toosi  
University of Technology



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

# Learning from data



K. N. Toosi  
University of Technology



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

# Learning from data



K. N. Toosi  
University of Technology



- Training data  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$ 
  - choose  $\theta$  such that  $f(\theta, \mathbf{x}_i)$  is close to  $y_i$



# Learning from data: Cost function



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

Cost function

# Learning from data: Cost function



K. N. Toosi  
University of Technology



- Training data  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$ 
  - choose  $\theta$  such that  $f(\theta, \mathbf{x}_i)$  is close to  $y_i$
  - cost function:

$$C(\theta) = \sum_{i=1..N} d( f(\theta, \mathbf{x}_i), y_i )$$

# Learning from data: Cost function



K. N. Toosi  
University of Technology



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

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

↓  
data output

# Learning from data: Cost function



K. N. Toosi  
University of Technology



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

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



model output given  $x_i$



# Learning from data: Cost function



K. N. Toosi  
University of Technology



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

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

↓  
distance

# Learning from data: Cost function



K. N. Toosi  
University of Technology



- Training data  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$ 
  - choose  $\theta$  such that  $f(\theta, \mathbf{x}_i)$  is close to  $y_i$
  - cost function:

$$C(\theta) = \sum_{i=1..n} \| f(\theta, \mathbf{x}_i) - y_i \|^2$$

distance



# Learning from data: Cost function



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

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

# Learning from data: Cost function



K. N. Toosi  
University of Technology



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

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

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

# Learning from data: Cost function



K. N. Toosi  
University of Technology



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

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

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

# Cost function



$$y = f(\underline{x}, \underline{\theta})$$

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

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

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

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

training data  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$

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

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

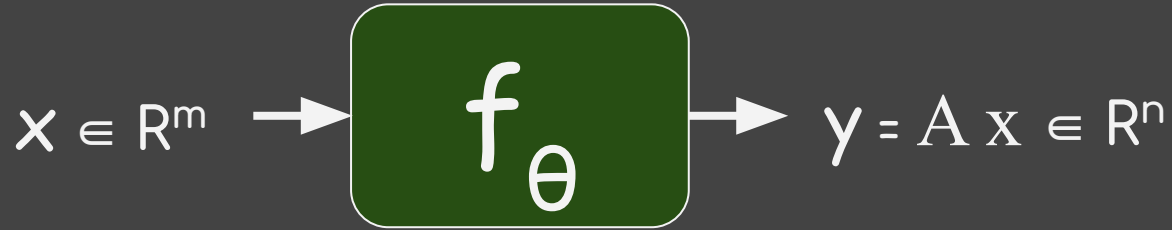
Example 
$$C(\theta) = \sum_{i=1}^N \|f(x_i, \theta) - y_i\|^2$$

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

# Example: Linear functions



K. N. Toosi  
University of Technology



# Affine functions



Affine functions

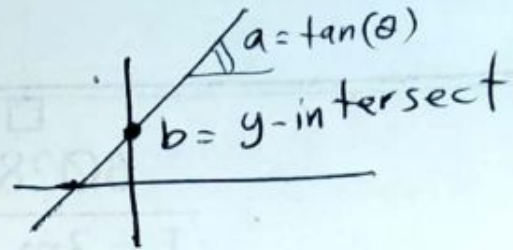
$$f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

$$f(x) = Ax + b$$

$\downarrow$                        $\downarrow$   
 $n \times m$                        $\in \mathbb{R}^n$   
 $\in \mathbb{R}$

$$m = n = 1 \Rightarrow f(x) = ax + b$$

line



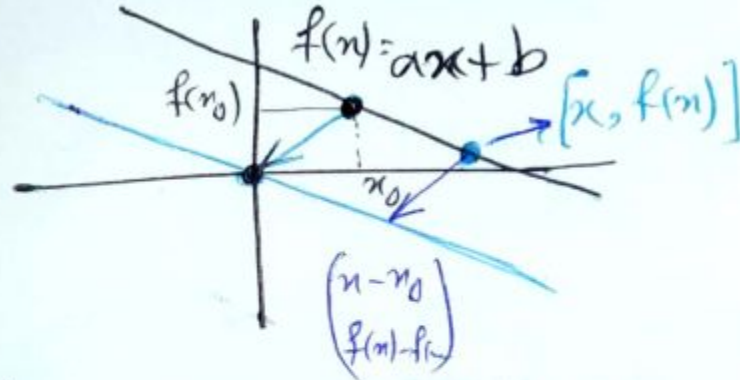


# Affine functions



$$\overbrace{f(n) - f(n_0)}^{y'} = a \overbrace{(n - n_0)}^{x'}$$

$$\begin{aligned} f(n) &= an - an_0 + f(n_0) \\ &= an + (f(n_0) - an_0) \\ &= an + b \end{aligned}$$



# Affine functions

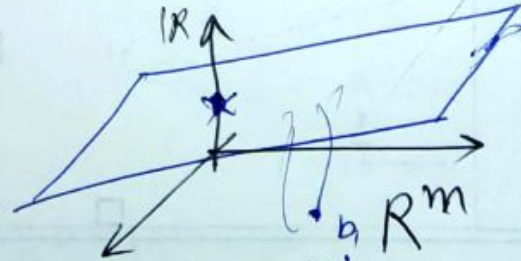


~~f~~  $f$ : affine

$\exists b$   $g(x) = f(x) - b$   
is a linear map.

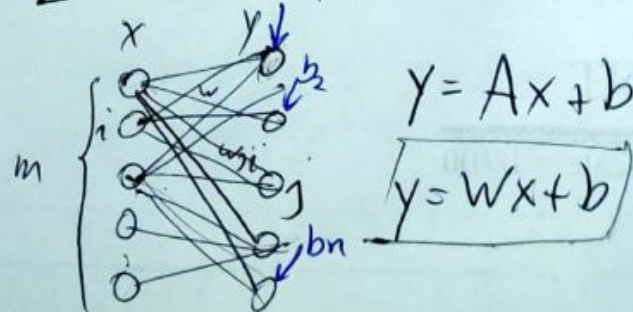
$$f: \mathbb{R}^m \rightarrow \mathbb{R}$$

$$f(x) = \underbrace{a^T}_{a^T \in \mathbb{R}^{1 \times m}} x + b \in \mathbb{R}$$



$$f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

$$f(x) = Ax + b$$



# Affine functions



$$f(x) = Ax + b \quad f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$
$$f(\underline{\theta}, x) = Ax + b \quad f: \mathbb{R}^{mn+n} \times \mathbb{R}^m \rightarrow \mathbb{R}^n$$

$\downarrow$                        $\downarrow$

~~$\mathbb{R}^m \times \mathbb{R}^n$~~        $n$                        $\mathbb{R}^{m \times n} \times \mathbb{R}^n$

$$\theta = (A, b)$$

$b \neq 0 \Rightarrow$   ~~$f$~~   $f$ ; is not linear in  $x$ .  $\checkmark$   $f(\alpha x) \neq f(x)$   
 $\Rightarrow$  is  $f$  linear in  $\theta = (A, b)$ ?

# Affine functions



$b \neq 0$  ~~for~~  $f$ ; is not linear in  $x$ .  $\checkmark$   $f(\alpha x) \neq f(x)$   
 $\Rightarrow$  is  $f$  linear ~~in~~  $\Theta = (A, b)$ ?

$$\Theta = (A, b) \quad \alpha\Theta = (\alpha A, \alpha b) \quad \text{scalar product}$$

$$\begin{aligned} \Theta_1 &= (A_1, b_1) \\ \Theta_2 &= (A_2, b_2) \end{aligned} \quad \Theta_1 + \Theta_2 = (A_1 + A_2, b_1 + b_2) \quad \text{vector addition}$$

is  $f(\Theta, x) = Ax + b$  ~~linear~~ linear is  $\Theta = (A, b)$ ?

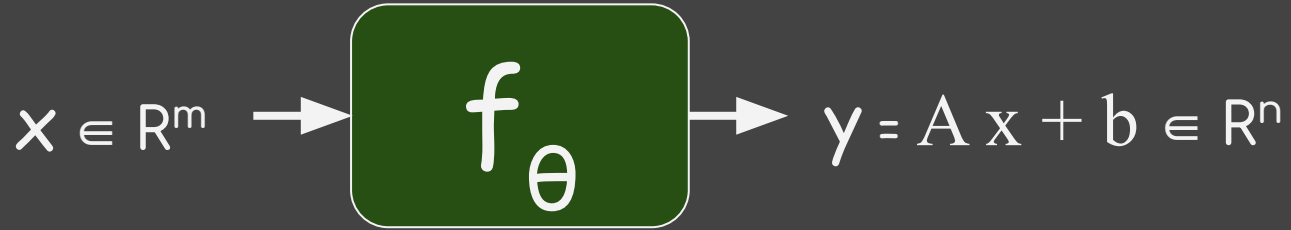
$$f(\alpha\Theta, x) = (\alpha A)x + \alpha b = \alpha(Ax + b) = \alpha f(\Theta, x)$$

$$f(\Theta_1 + \Theta_2, x) = f(\Theta_1, x) + f(\Theta_2, x)$$

# Example: Linear Regression



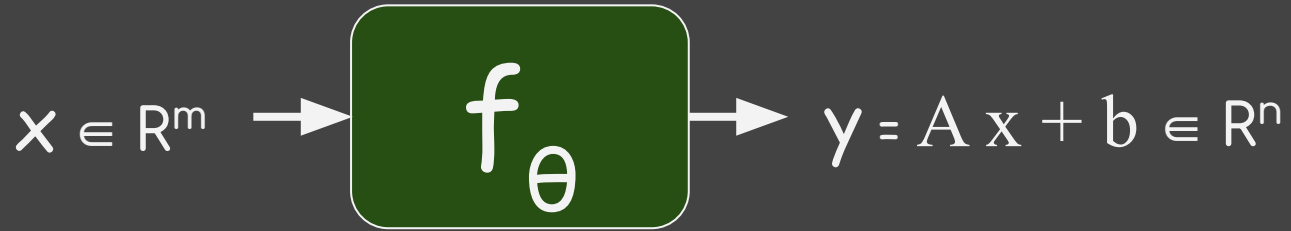
K. N. Toosi  
University of Technology



# Example: Linear Regression



K. N. Toosi  
University of Technology



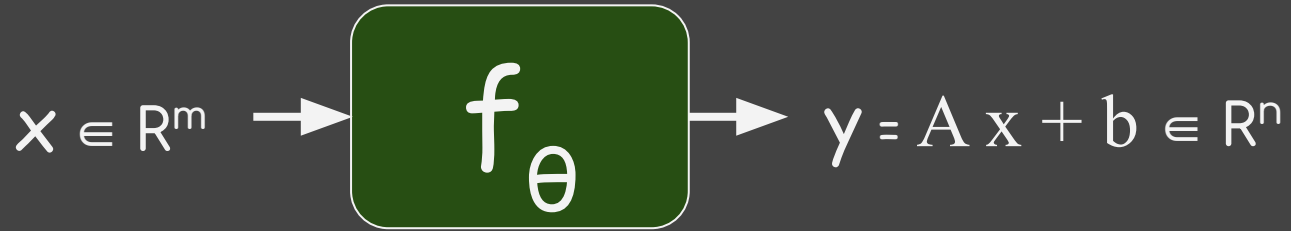
A: ? by ? matrix

b: ?-D vector

# Example: Linear Regression



K. N. Toosi  
University of Technology



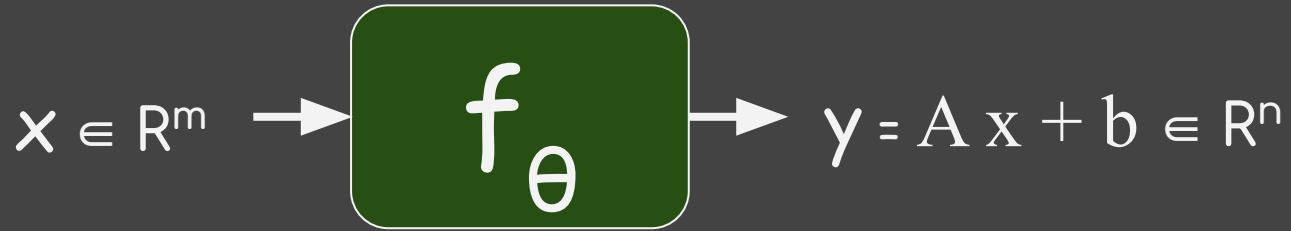
$\mathbf{A}$ :  $n$  by  $m$  matrix

$\mathbf{b}$ :  $n$ -D vector

# Example: Linear Regression



K. N. Toosi  
University of Technology



$$\mathbf{y} = \mathbf{f}(\theta, \mathbf{x})$$

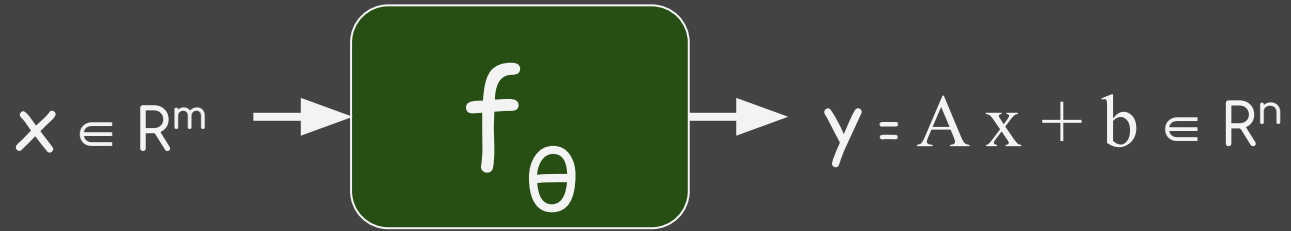
$$\theta = ?$$



# Example: Linear Regression



K. N. Toosi  
University of Technology



$$\mathbf{y} = \mathbf{f}(\theta, \mathbf{x})$$

$$\theta = (\mathbf{A}, \mathbf{b})$$

# Example: Linear Regression



$$f(x) = Ax + b$$

$$f: \mathbb{R}^m \rightarrow \mathbb{R}^n$$

is  $f$  linear?

$$f(\alpha x) = A(\alpha x) + b = \alpha Ax + b$$

$$\alpha f(x) = \alpha Ax + \alpha b$$

$\Rightarrow f$  is not linear in general ( $b \neq 0$ )

# Affine maps



K. N. Toosi  
University of Technology

$$\cancel{g(x+x_0)} \quad g(x) = f(x+x_0) - f(x_0) = A(x+x_0) + b - (Ax_0 + b) = Ax$$

$$g(x) = f(x) - f(0) = Ax$$
$$g(x) = f(x) - b = Ax$$

$$f(x) = Ax + b$$

affine function

$$f(x) : \mathbb{R}^m \rightarrow \mathbb{R}^n$$

$$\exists v \in \mathbb{R}^n$$

$$g(x) = f(x) - v$$

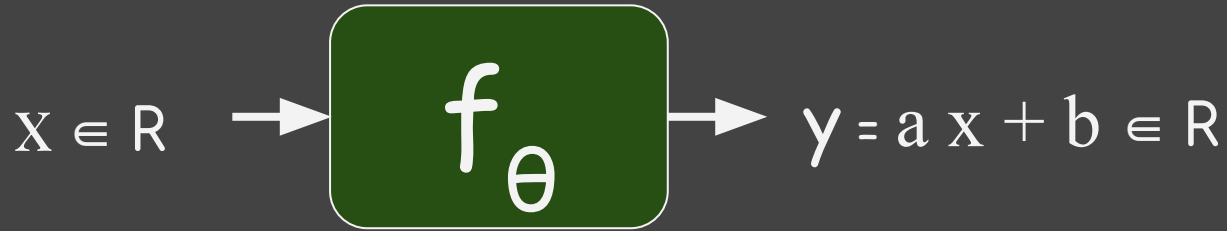
linear

$$f(x-x_0) - f(x_0)$$

# Example: Linear Regression



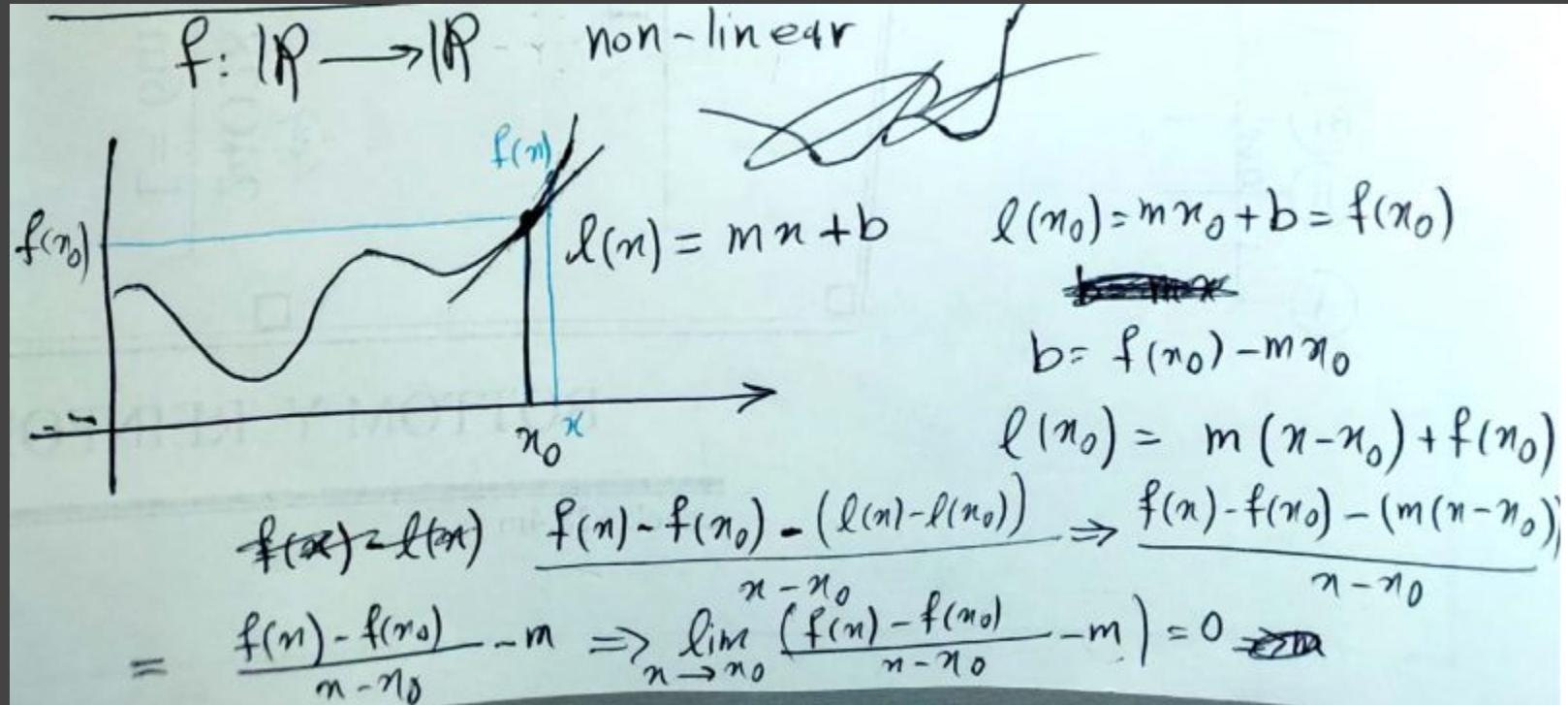
K. N. Toosi  
University of Technology



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

$$\theta = ?$$

# nonlinear functions



# nonlinear functions: linearization



K. N. Toosi  
University of Technology

$$m = \lim_{n \rightarrow n_0} \frac{f(n) - f(n_0)}{n - n_0} = f'(n_0)$$

