Supervised Learning

+ definitions
+ Linear Regression
+ Batch & stochastic gradient descent
+ Normal Equations

Supervised learning

Prediction

$h: X \longrightarrow Y$

Image	Contains cat
Text	Is hate speech?
House data	Price

Given: Training set

$\{(x^{(1)}, y^{(1)}) \ldots (x^{(n)}, y^{(n)})\} \quad x^{(i)} \in X,\ y^{(i)} \in Y$

Do: find "good" $h: X \to Y$ hypothesis

this job of training algorithm

We use h on new data (x)

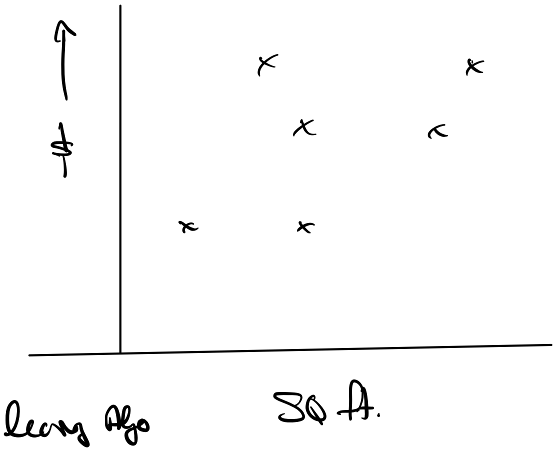
call this prediction, we are very interested in $x \notin$ training set

if y is discrete $\Rightarrow$ classification

y is continuous $\Rightarrow$ regression

Example Data (House Prices)

SQ ft	Price (1k)
2100	400
2500	900
1127	800
⋮	⋮



TRAINING SET → Learning Algo

h: SQ ft → Price

How do we represent h?

$$h(x) = \theta_0 + \theta_1 x_1 \qquad \text{(affine fn.)}$$

	$x_1^{(i)}$ Size	x_2 Bedroom	x_3 Lot size	Price
$x^{(1)}$	2104	4 $x_2^{(1)}$	45k	400
$x^{(2)}$	2500	3	30k	900

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots$$

$$= \sum_{j=0}^{3} \theta_j x_j \qquad \underline{\text{NB}}\ x_0 \text{ identically } 1$$

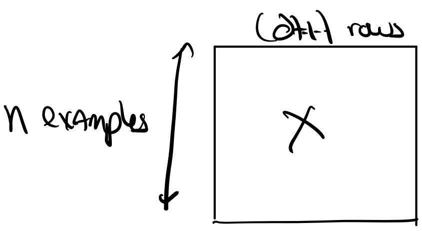
$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \qquad x^{(1)} = \begin{bmatrix} x_0^{(1)} \\ x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \end{bmatrix} \begin{array}{l} 1 \\ \text{Size} \\ \text{Bedrooms} \\ \text{Lot size} \end{array} \qquad y^{(1)} \text{ is Price}$$

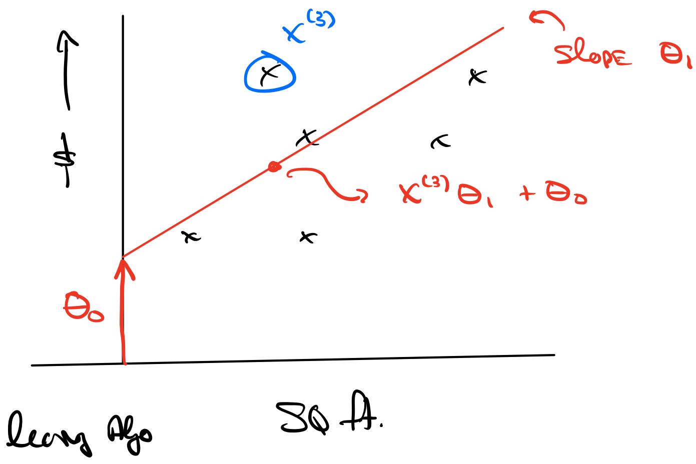
Parameters — Inputs / Features — Output / Target

(x, y) is a training example

$(x^{(i)}, y^{(i)})$ is i^{th} example $\quad i$ runs $1 \ldots n$

n examples and d features $\Rightarrow$ $x_i^{(i)}, \theta$ are $\underline{d+1}$ dimensional





$$h_\theta(x) = \sum_{j=0}^{d} \theta_j x_j$$

WANT TO CHOOSE θ s.t. $h_\theta(x) \approx y$

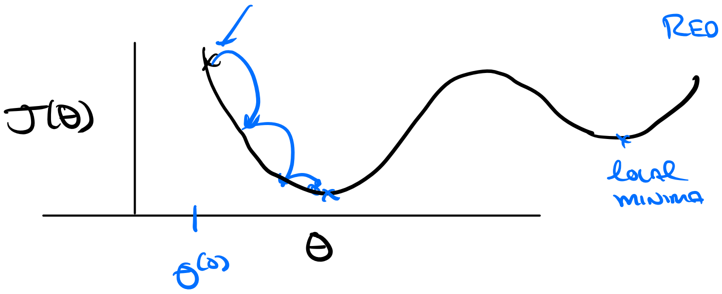
IDEA: $$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$ Cost function (least squares)

$$\min_\theta J(\theta)$$

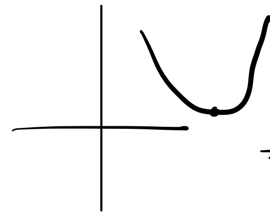
Gradient Descent

START $\theta^{(0)}$ AT RANDOM OR ZERO

REDUCE USING GRADIENT



if J is nice (convex)



local min
$\Rightarrow$ global min!

$$\theta^{(0)} := 0$$

LEARNING RATE

$$\theta_j^{(t+1)} := \theta_j^{(t)} - \alpha \frac{\partial}{\partial \theta_j} J(\theta^{(t)}) \qquad j = 1 \ldots d$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \sum_{i=1}^{n} \frac{1}{2} \frac{\partial}{\partial \theta_j} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

$$= \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \frac{\partial}{\partial \theta_j} h_\theta(x^{(i)})$$

$$h_\theta(x^{(i)}) = \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \cdots + \theta_d x_d^{(i)}$$

$$\frac{\partial h_\theta(x^{(i)})}{\partial \theta_j} = x_j^{(i)}$$

$$\theta_j^{(t+1)} := \theta_j^{(t)} - \alpha \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

SOMETIMES WRITE AS $\theta^{(t+1)} := \theta^{(t)} - \alpha \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)}$

vector notation

BATCH VERSUS STOCHASTIC MINIBATCH

$$\theta^{(t+1)} := \theta^{(t)} - \alpha \sum_{i=1}^{n} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)}$$

MINIBATCH: Randomly select $b < n$ points AND Estimate gradient

1. Pick b points $\{i_1 \ldots i_b\} = \mathcal{B}$

2.

$$\theta^{(t)} := \theta^{(t)} - \alpha_b \sum_{k \in \mathcal{B}} \left(h_\theta(x^{(k)}) - y^{(k)} \right) x^{(k)}$$

ONE DETAIL Scale α AND α_b differently.

TRADEOFF: NOISIER BUT much faster

faster: Imagine if TRAINING SET contains 100 copies of SAME point

$\Rightarrow$ Not AS RIDICULOUS AS it seems (NEAR COPIES)

How do you choose $\mathcal{B}$? Sadly, WHATEVER WORKS

Normal Equation

$$\nabla_\theta J(\theta) = \begin{bmatrix} \frac{\partial}{\partial \theta_0} J(\theta) \\ \frac{\partial}{\partial \theta_1} J(\theta) \\ \vdots \end{bmatrix}$$

$A \in \mathbb{R}^{2\times 2}$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad f : A \to \mathbb{R}$$

then

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial a_{11}} & \frac{\partial f}{\partial a_{12}} \\ \frac{\partial f}{\partial a_{21}} & \frac{\partial f}{\partial a_{22}} \end{bmatrix}$$

Now, we want to find minimum

$$\nabla_\theta J(\theta) = \vec{0} \qquad (\nabla_\theta J(\theta) \in \mathbb{R}^{d+1})$$

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$X = \begin{bmatrix} - x^{(1)} - \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{bmatrix} \in \mathbb{R}^{n\times d}$$ Design matrix

$$X\theta = \begin{bmatrix} - x^{(1)} - \\ \\ \\ \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \vdots \end{bmatrix} = \begin{bmatrix} h_\theta(x^{(1)}) \\ \vdots \\ h_\theta(x^{(n)}) \end{bmatrix}$$

$$y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$ then $J(\theta) = \frac{1}{2}(X\theta - y)^T (X\theta - y)$

$$\nabla_\theta J(\theta) = X^T X\theta - X^T y = 0 \Rightarrow \theta = (X^T X)^{-1} X^T y$$

optimal value.