1/11/2024 Linear Regression

linear function of input — Predicting a real number

target (y)

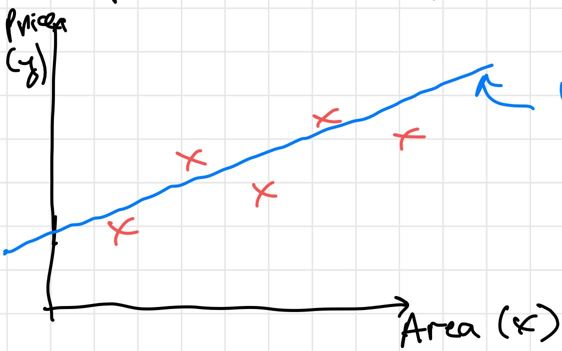
Features (d total), each house is feature vector $x \in \mathbb{R}^d$

Sale price	Area	#bedrooms	has garage?	...	constant feature
\$500k $= y^{(1)}$	1200	2	0 $= x^{(1)}$		1
\$700k	2000	3	1		1
	⋮				1

constant feature → makes b param optional

Dataset $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$ of size n

Simplest case: $d = 1$



$$y = wx + b$$

slope (w), intercept (b)

parameters (to be learned)

When $d > 1$:

$$y = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b$$

$$= w^T x + b$$

$1 \times d$, $d \times 1$, $= \square$ 1×1

Parameters:

$w \in \mathbb{R}^d$

$b \in \mathbb{R}$

Q: How to choose good w & b?

A: Define a loss function

$$L(w,b) = \left[\text{How bad do } w \,\&\, b \text{ fit our observed data?}\right]$$

$$= \frac{1}{n}\sum_{i=1}^{n}\Big(\underbrace{w^T x^{(i)} + b}_{\text{model prediction}} - \underbrace{y^{(i)}}_{\text{true output}}\Big)^2$$

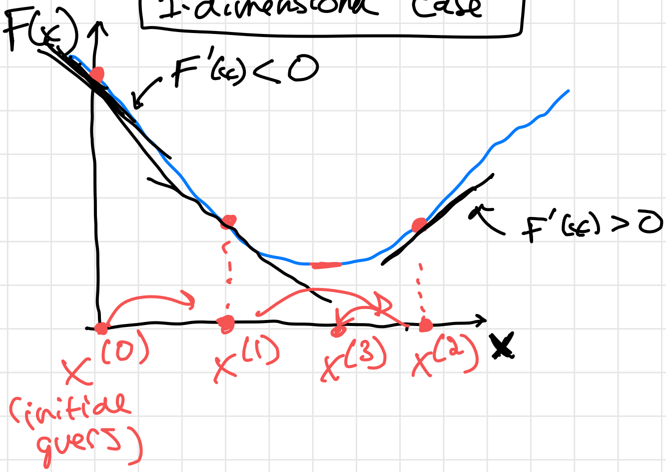
Goal: Find w & b that minimize $L(w,b)$
<u>optimization problem!</u>

$\boxed{\text{Gradient Descent}}$ General method for minimizing function

Given: Function F from $\mathbb{R}^d \to \mathbb{R}$, differentiable

Gradient will try to find x that minimizes $F(x)$

$\boxed{\text{1-dimensional case}}$



Have current guess $x^{(t)}$

If $F'(x^{(t)}) < 0$, increase $x^{(t)}$ to yield $x^{(t+1)}$

If $F'(x^{(t)}) > 0$ decrease $x^{(t)}$ to yield $x^{(t+1)}$

d-dimensional case optimizing w.r.t. $x \in \mathbb{R}^d$

Partial Derivative: $\frac{\partial F}{\partial x_i}$ ← Take derivative wrt x_i holding all other x_j's constant

New G.D. Rule:

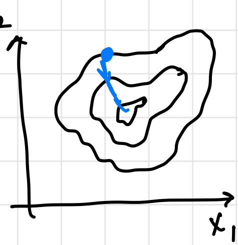
For each $i = 1, \dots, d$:

If $\left.\frac{\partial F}{\partial x_i}\right|_{x = x^{(t)}} < 0$, increase $x_i^{(t)}$

If $\left.\frac{\partial F}{\partial x_i}\right|_{x = x^{(t)}} > 0$, decrease $x_i^{(t)}$

Gradient $\nabla_x F(x) = \left[\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \dots \frac{\partial F}{\partial x_d}\right]$

Starting at $x^{(t)}$, best direction to go (to minimize F) is direction of negative gradient



Fact: Negative gradient is direction of steepest descent

Gradient Descent Algorithm:

$x^{(0)} \leftarrow [0, 0, \dots, 0] \in \mathbb{R}^d$

for t in $1, \dots, \boxed{T}$: ← total # steps

$$x^{(t)} \leftarrow x^{(t-1)} - \boxed{\eta}\, \nabla_x F(x^{(t-1)})$$

← learning rate (e.g. 0.01)

return $x^{(T)}$

Gradient Descent for Linear Regression

$$L(w) = \frac{1}{n} \sum_{i=1}^{n} \left(w^T x^{(i)} - y^{(i)} \right)^2$$

$$\boxed{\frac{d}{dw} 8w = 8}$$

$$\nabla_w L(w) = \frac{1}{n} \sum_{i=1}^{n} 2 \cdot \underbrace{\left(w^T x^{(i)} - y^{(i)} \right)}_{\text{Scalar}} \cdot \underbrace{x^{(i)}}_{\text{Vector}}$$

GD for Linear Regression:

$w^{(0)} \leftarrow [0, \ldots 0] \in \mathbb{R}^d$

for $t = 1, \ldots, T$:

$$w^{(t)} \leftarrow w^{(t-1)} - \eta \cdot \frac{1}{n} \sum_{i=1}^{n} 2 \cdot \underbrace{\left(w^T x^{(i)} - y^{(i)} \right)}_{\text{Determines if we add or subtract multiple of } x^{(i)}} \cdot x^{(i)}$$

return $w^{(T)}$

If $w^T x^{(i)} - y^{(i)} > 0$: prediction too large
subtract multiple of $x^{(i)}$ from w
$\Rightarrow$ $w^T x^{(i)}$ smaller

If $w^T x^{(i)} - y^{(i)} < 0$: prediction too small,
add multiple of $x^{(i)}$
$\Rightarrow$ $w^T x^{(i)}$ bigger