8/24/2023 **Linear Regression**

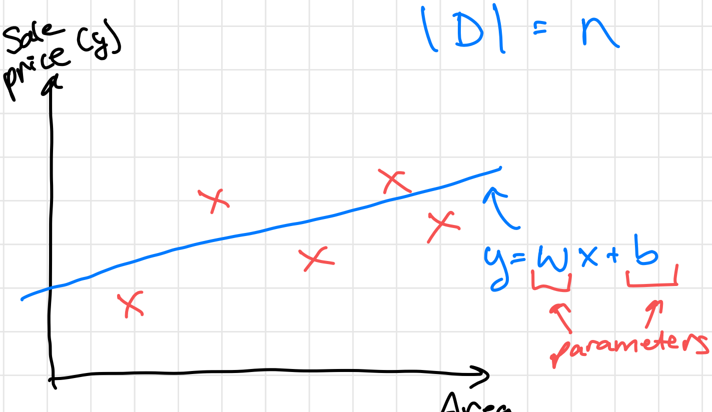
linear function of input — Predicting a real number

target (y) — Features x (d total)

Sale price	Area	# bedrooms	has garage?	constant
$500k ($y^{(1)}$)	1200	2	0 ($x^{(1)} \in \mathbb{R}^d$)	1
$800k	1500	3	1	1
	⋮			⋮

Training dataset $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$

$$|D| = n$$



1-D case

In d dimensions

predicted

$$y = \sum_{i=1}^{d} w_i x_i + b$$

$$= w^T x + b$$

$1 \times d$, $d \times 1$, $= 1 \times 1$

parameters $w \in \mathbb{R}^d$, $b \in \mathbb{R}$

Q: How do we choose good w & b?

A: Define a loss function

$$L(w,b) = [\text{how bad does } w \,\&\, b \text{ fit our observed data}]$$

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

Goal: minimize $L(w,b)$ with respect to w & b

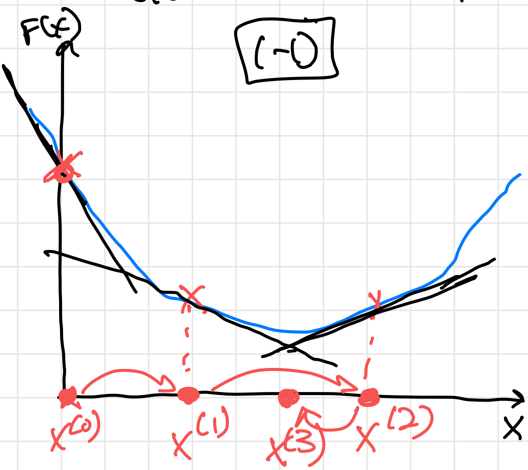
$\boxed{\text{Optimization Problem}}$

Gradient Descent

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

Goal: minimize F

1-D



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

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

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

" $\quad$ " $= 0$, stop

d-dimensional

How should $x_i^{(t)}$ change?

Same except we use $\frac{\partial F}{\partial x_i}$ "partial derivative"

New rule:

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

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

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

Starting at $x^{(t)}$, we should step in direction of the <u>negative gradient</u>

Bonus justification (see notes):

Negative gradient is direction of <u>steepest</u> descent

<u>Gradient descent algorithm</u>:

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

for $t = 1, \dots, \boxed{T}$:

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

return $x^{(T)}$

learning rate (e.g. 0.01)

total steps

Back to linear regression

$$\boxed{\frac{d}{dx} 8x = 8}$$

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

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

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

G.D. for linear regression

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

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

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

return $w^{(T)}$