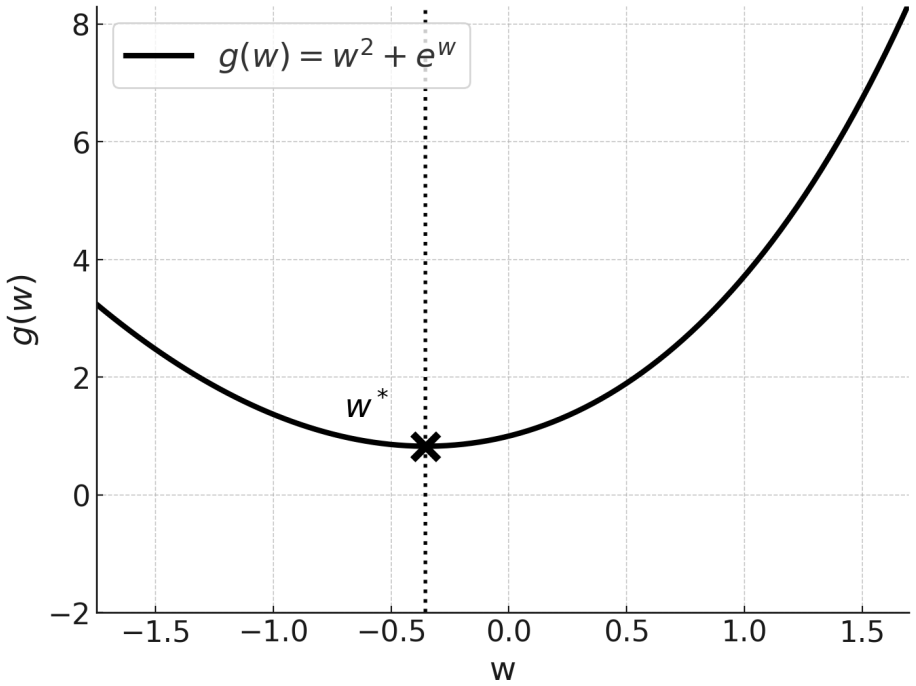
Gradient Descent

Ex $g(w) = w^2 + e^w$

$g'(w) = 2w + e^w$

$g''(w) = 2 + e^w \geq 0$

Convex!



$g'(w) = 2w + e^w = 0$

$e^w = -2w$

$w = \log(-2w) = \log(-2) + \log(w) \Rightarrow \log(-2) = w - \log(w)$

No closed form solution for w^*

calculating $A^{-1} \simeq d^3$ operations on computer which is a lot

Second Order Gradient Descent (Newton's Method)

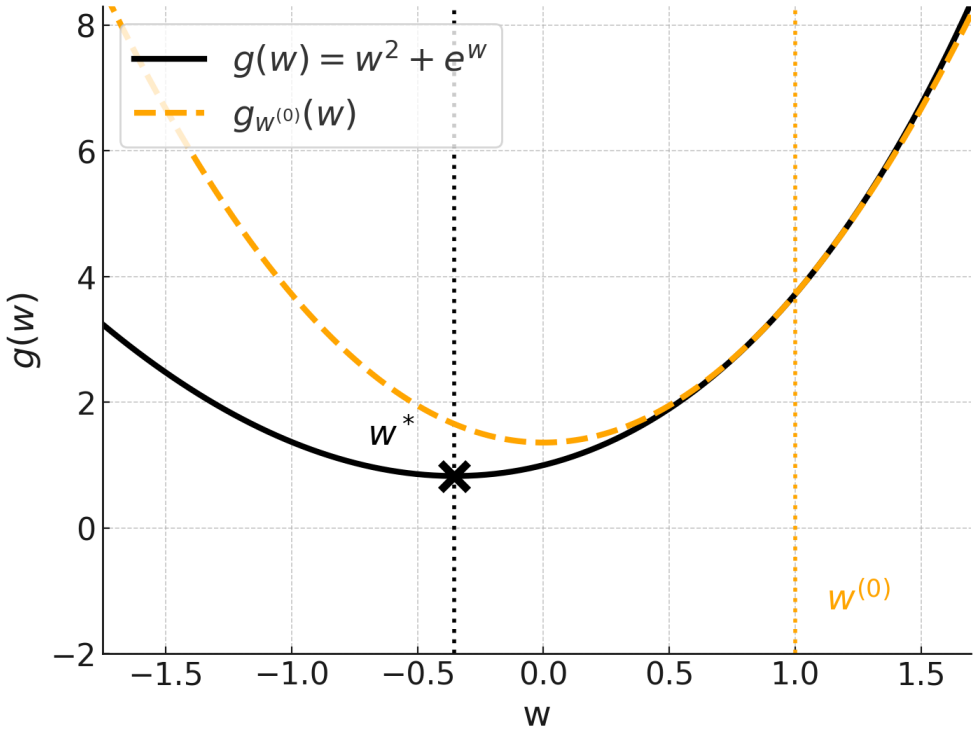
If $g(w)$ is at least a degree 5 polynomial

Then there does not exist a closed form solution for $g(w) = 0$

Our approach: approximate $g(w)$ with a a second order polynomial using Taylor's approximation

$$g(w) \approx g_{w^{(0)}}(w) = g(w^{(0)}) + g'(w^{(0)})(w - w^{(0)}) + \frac{g''(w^{(0)})}{2}(w - w^{(0)})^2$$

$g_{w^{(0)}}(w)$ where $w^{(0)} = 1$



Goal: $\underset{w \in \mathcal{W}}{\operatorname{argmin}}\, g_{w^{(0)}}(w) = \underset{w \in \mathcal{W}}{\operatorname{argmin}}\, g(w^{(0)}) + g'(w^{(0)})(w - w^{(0)}) + \frac{g''(w^{(0)})}{2}(w - w^{(0)})^2$

$g'_{w^{(0)}}(w) = g'(w^{(0)}) + g''(w^{(0)})(w - w^{(0)}) = 0$

$\Rightarrow g''(w^{(0)})\, w = g''(w^{(0)})\, w^{(0)} - g'(w^{(0)})$

$\Rightarrow \underbrace{w}_{w^{(1)}} = w^{(0)} - \frac{g'(w^{(0)})}{g''(w^{(0)})}$

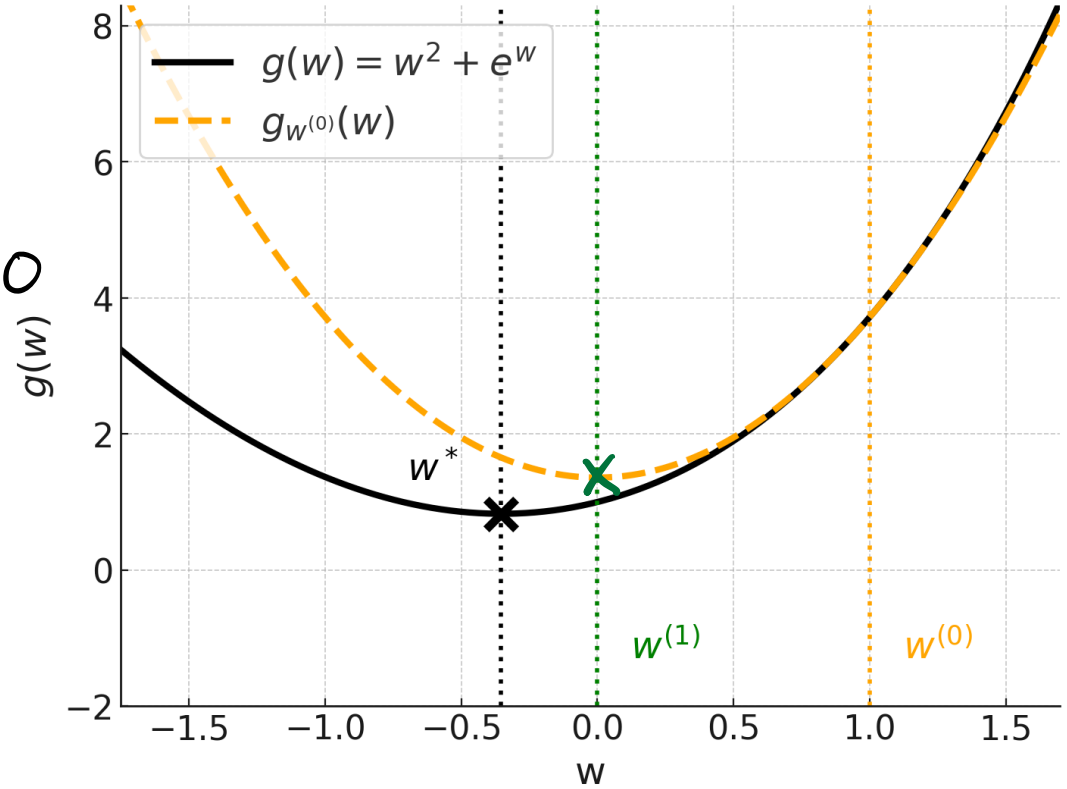
$w^{(0)} = 1$

$g'(w) = 2w + e^w$

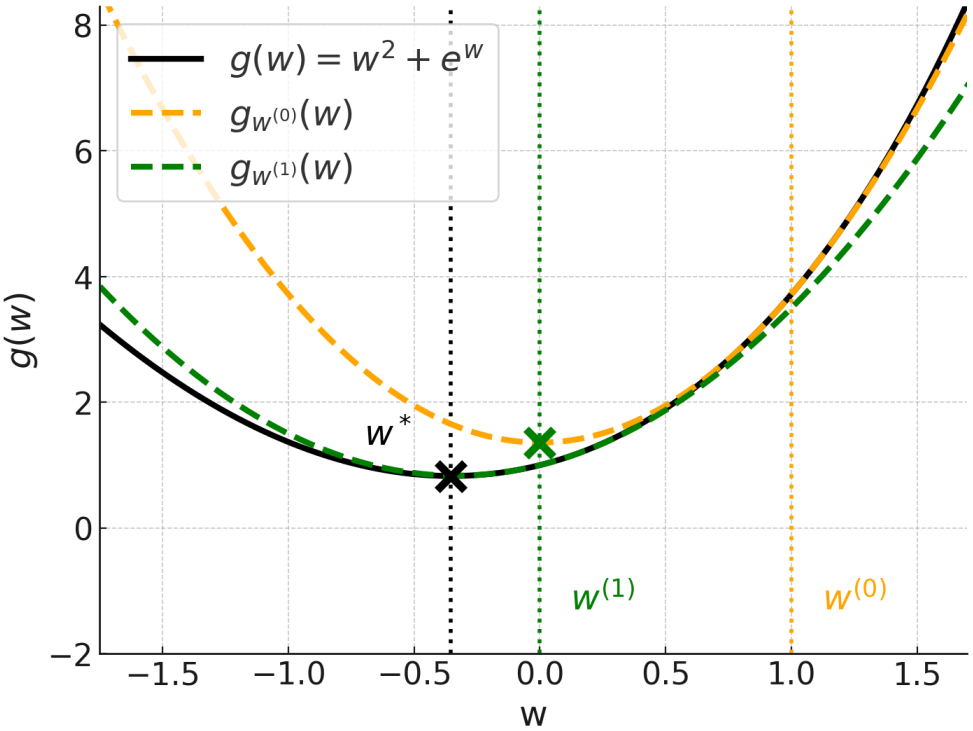
$g''(w) = 2 + e^w \geq 0$

$g'(1) = 2 + e$

$g''(1) = 2 + e$



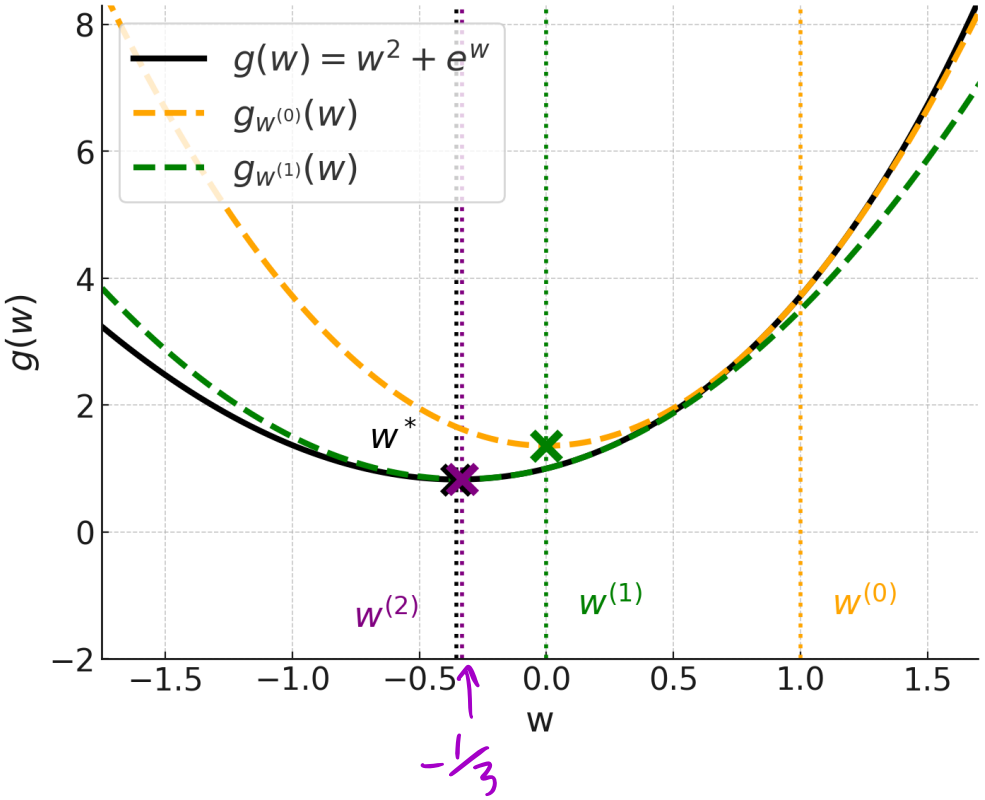
$w^{(1)} = 1 - \frac{2+e}{2+e} = 0$


$g(w) = w^2 + e^w$
$g_{w^{(0)}}(w)$
$g_{w^{(1)}}(w)$
w^*
$w^{(1)}$
$w^{(0)}$
$g(w)$
w
8
6
4
2
0
−2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

$$w^{(2)} = w^{(1)} - \frac{g'(w^{(1)})}{g''(w^{(1)})}$$

$$= 0 - \frac{1}{3}$$

$$= -\frac{1}{3}$$



In general $w^{(t+1)} = w^{(t)} - \frac{g'(w^{(t)})}{g''(w^{(t)})}$

for all $t \in \mathbb{N}$ and $w^{(0)}$ is your initial guess

$w^{(t+1)}$ approaches w^* as $t \to \infty$

(First-order) Gradient Descent

Calculating the second derivative (Hessian) is computationally expensive in higher dimensions

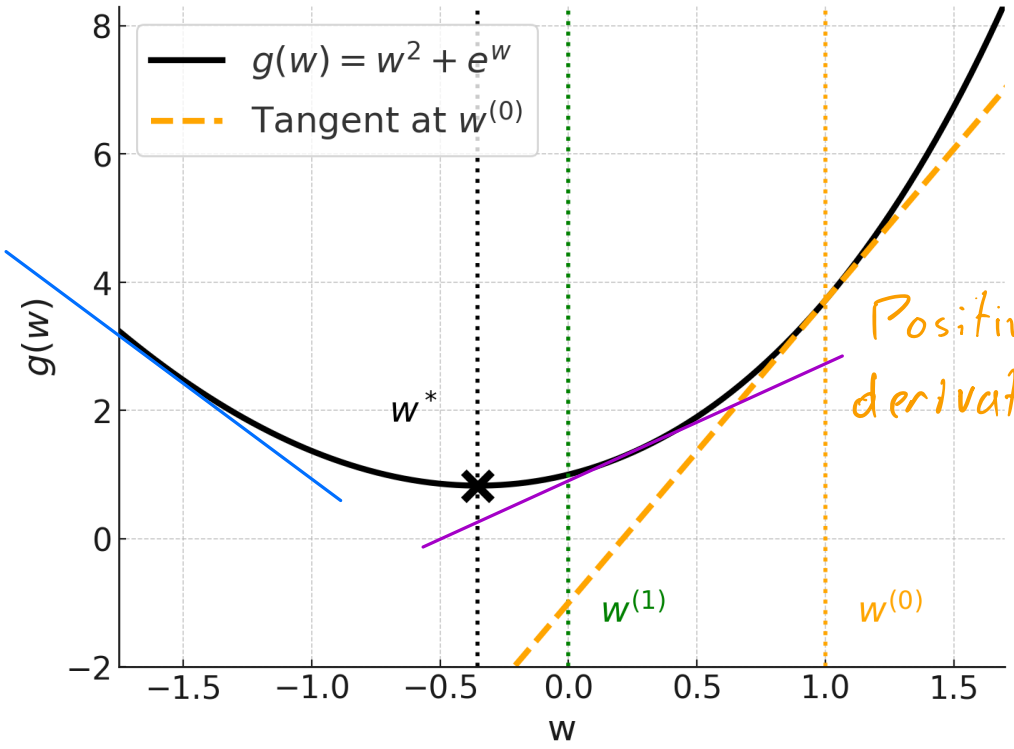
$$w^{(t+1)} = w^{(t)} - \eta^{(t)} g'(w^{(t)})$$

$\eta^{(t)}$ is the "step-size" "Learning rate"

if we knew $g''(w^{(t)})$ we should set

$$\eta^{(t)} = \frac{1}{g''(w^{(t)})}$$

negative derivative

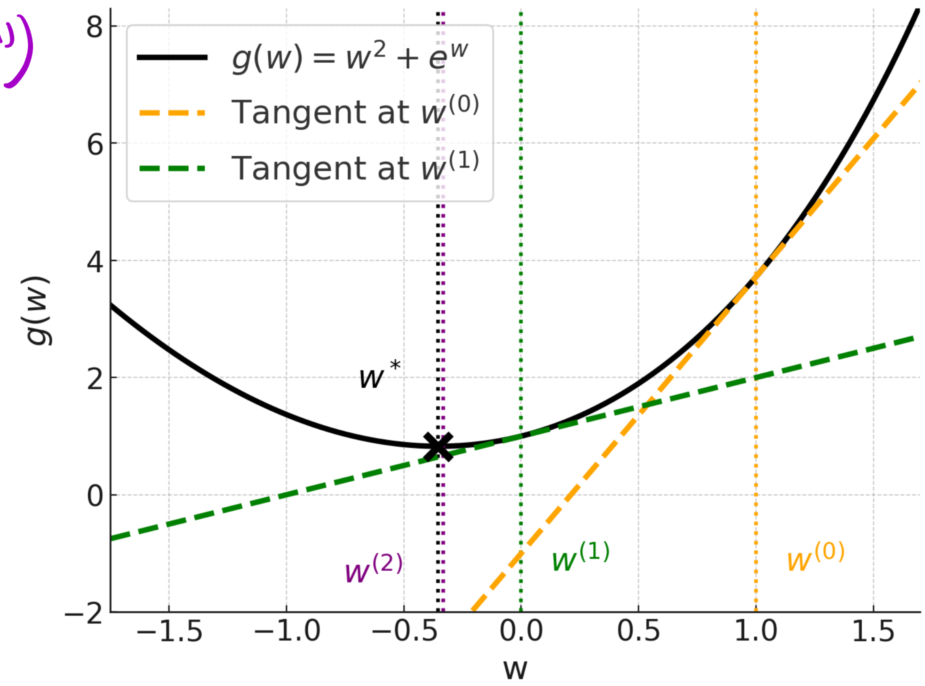


Positive derivative

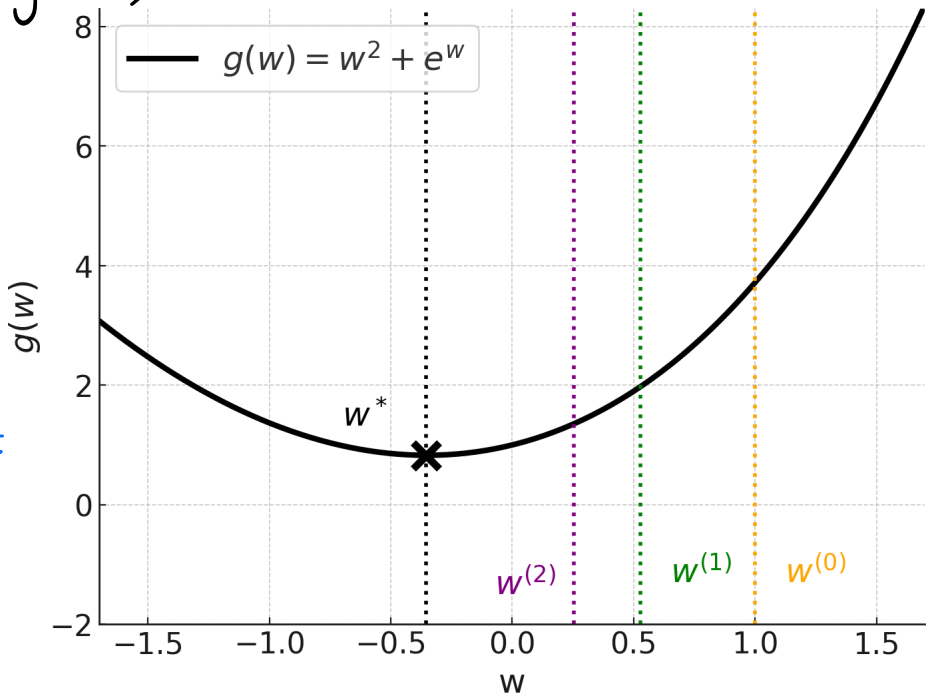
$$w^{(1)} = w^{(0)} - \frac{1}{g''(w^{(0)})} g'(w^{(0)})$$

with $w^{(0)} = 1$

$$w^{(2)} = w^{(1)} - \underbrace{\frac{1}{g''(w^{(1)})}}_{= z^{(1)}} g'(w^{(1)})$$



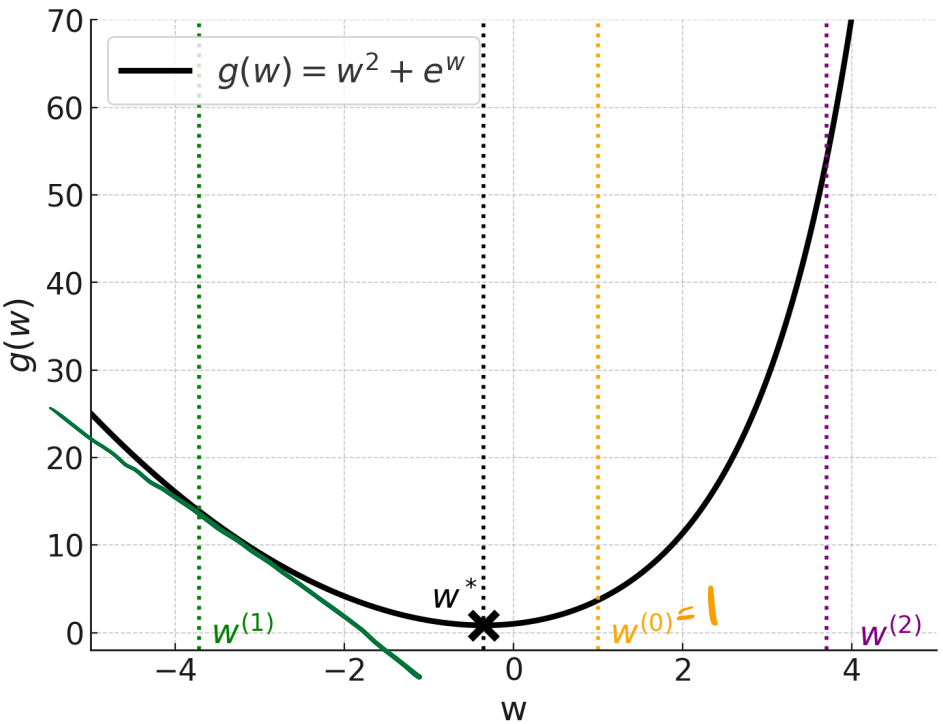
$$\eta^{(t)} = \frac{1}{10} \leq \left(\frac{1}{2+e} = \frac{1}{g''(w^{(0)})}\right)$$



Small $\eta^{(t)}$

Slowly reaches w^*

$$\eta^{(t)} = 1 > \frac{1}{2+e}$$



if $\eta^{(t)}$ is large

then $w^{(t)}$ can diverge from w^* or oscillate

- Small step size is safe (you will get close to w^*); however, it can take long
- Large step size can get you to w^* fast but sometimes it can diverge

---

Multivariate Gradient Descent

$\vec{w} \in \mathcal{W} = \mathbb{R}^d$, $d > 1$ $\quad g: \mathbb{R}^d \to \mathbb{R}$

Obj: $\vec{w}^* = (w_1^*, \ldots, w_d^*)^T = \underset{\vec{w} \in \mathbb{R}^d}{\operatorname{argmin}}\, g(\vec{w})$

gradient descent update rule:

$$\vec{w}^{(t+1)} = \vec{w}^{(t)} - \eta^{(t)} \nabla g(\vec{w}^{(t)}) \quad \in \mathbb{R}^d$$

where $\nabla g(\vec{w}^{(t)}) = \left(\frac{\partial g}{\partial w_1}, \ldots, \frac{\partial g}{\partial w_d} \right)$

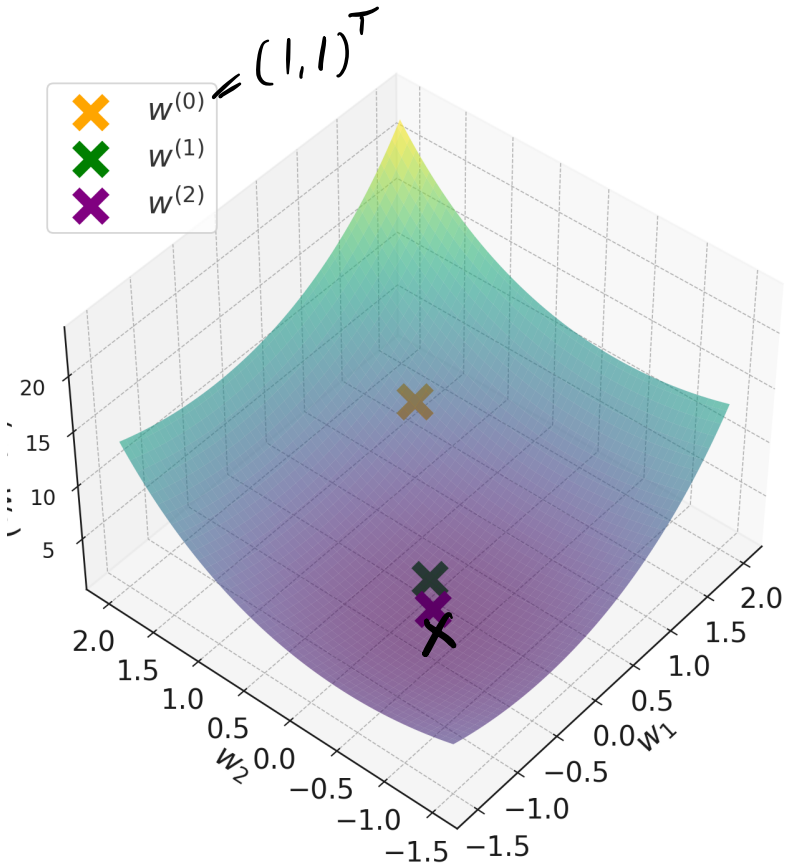
Ex: $g(w_1, w_2) = w_1^2 + e^{w_1} + w_2^2 + e^{w_2}$

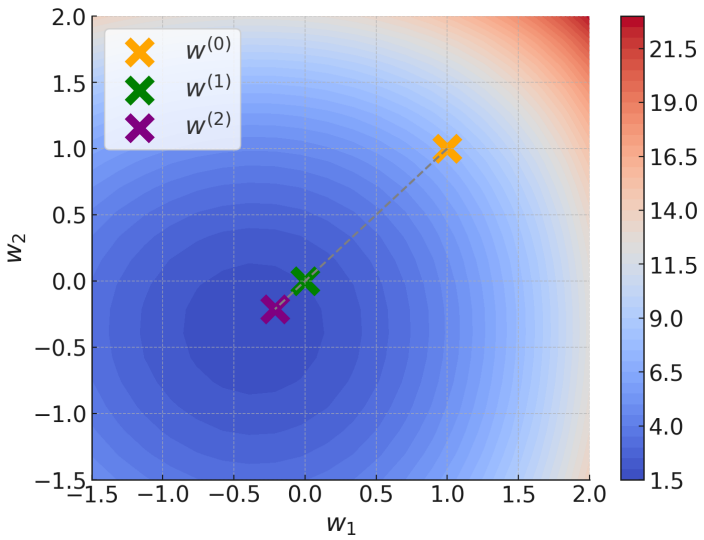
$$\frac{\partial g}{\partial w_1} = 2w_1 + e^{w_1}, \quad \frac{\partial g}{\partial w_2} = 2w_2 + e^{w_2}$$

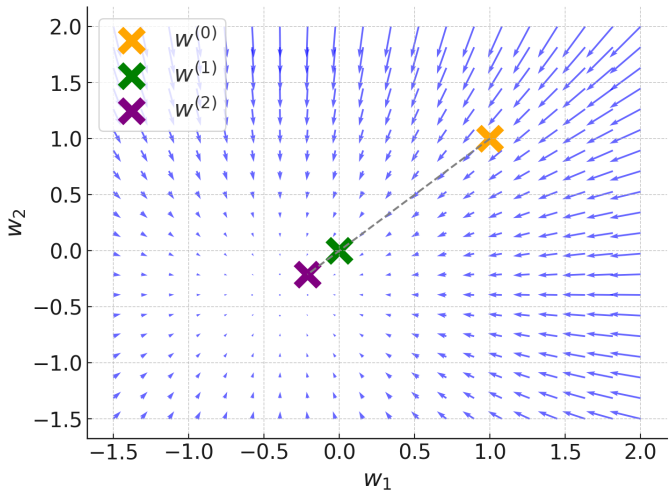
$$\begin{pmatrix} w_1^{(t+1)} \\ w_2^{(t+1)} \end{pmatrix} = \begin{pmatrix} w_1^{(t)} \\ w_2^{(t)} \end{pmatrix} - \eta^{(t)} \begin{pmatrix} 2w_1^{(t)} + e^{w_1^{(t)}} \\ 2w_2^{(t)} + e^{w_2^{(t)}} \end{pmatrix}$$

$$\vec{w}^{(0)} = (1, 1)^T, \quad \eta^{(t)} = \frac{1}{2+e}$$

$$\vec{w}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \frac{1}{2+e} \begin{pmatrix} 2+e \\ 2+e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$


$w^{(0)}$
$w^{(1)}$
$w^{(2)}$
$= (1,1)^T$
20
15
10
5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
w_2
w_1

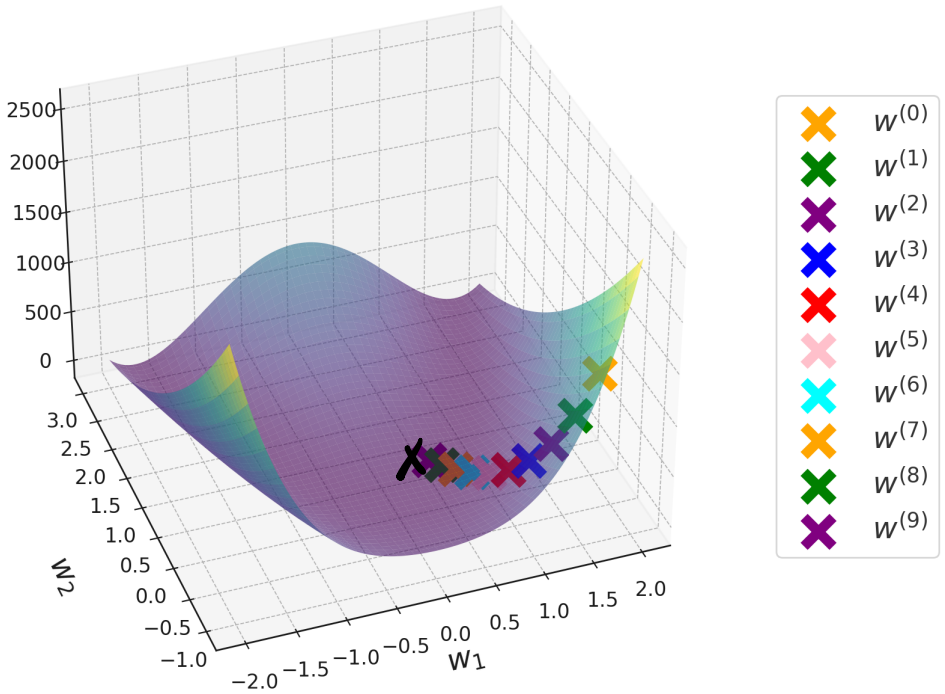

$w^{(0)}$
$w^{(1)}$
$w^{(2)}$
21.5
19.0
16.5
14.0
11.5
9.0
6.5
4.0
1.5
w_1
w_2

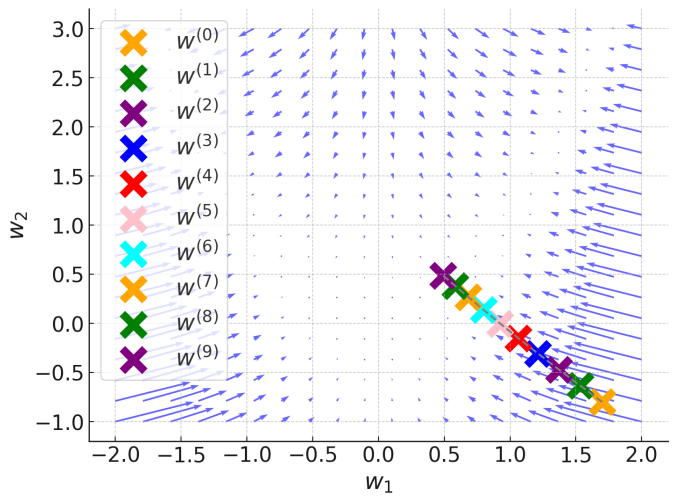

$w^{(0)}$
$w^{(1)}$
$w^{(2)}$
w_1
w_2

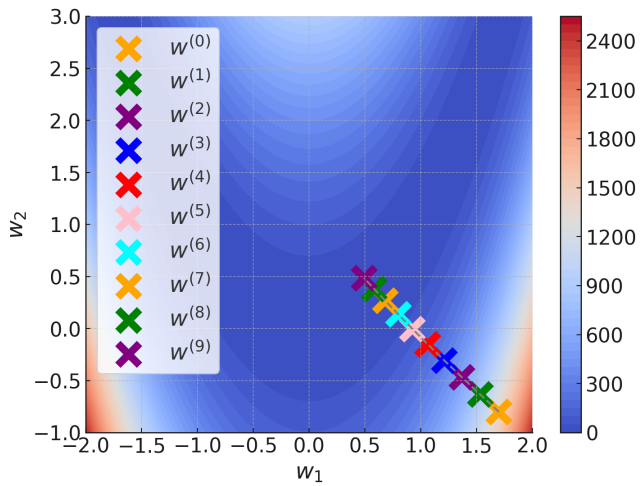
Ex: $g(w_1, w_2) = (1 - w_1)^2 + 100(w_2 - w_1^2)^2$

$= g(\vec{w})$

$\eta^{(t)} = \frac{1}{6}$







Selecting Step Sizes

1. Constant: $\eta^{(t)} = \eta \in (0, \infty)$

2. Inverse decaying: $\eta^{(t)} = \frac{\eta}{1+\lambda t} \propto \frac{1}{t}$ $\quad \eta, \lambda \in (0, \infty)$

3. Exponential decaying: $\eta^{(t)} = \eta \frac{1}{e^{\lambda t}}$ $\quad \eta, \lambda \in (0, \infty)$

4. Normalized gradient: $\eta^{(t)} = \frac{\eta}{\epsilon + \|\nabla g(\vec{w}^{(t)})\|_2}$

size of your gradient

why not $\eta^{(t)} = \lambda t$

Because it increases with time which you do not want

Gradient Descent with $\hat{L}(f)$

$D = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$, $\mathcal{X} = \mathbb{R}^{d+1}$, $\mathcal{Y} = \mathbb{R}$

$\mathcal{F} = \{f \mid f : \mathbb{R}^{d+1} \to \mathbb{R} \text{ s.t. } f(\vec{x}) = \vec{x}^T \vec{w}, \underset{\| }{\vec{w}} \in \mathbb{R}^{d+1}\}$

$(w_0, w_1, \ldots, w_d)^T$

$\ell(\hat{y}, y) = (\hat{y} - y)^2$

Obj: $\hat{\vec{w}} = \underset{\vec{w} \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \hat{L}(\vec{w})$

$= \underset{\vec{w} \in \mathbb{R}^{d+1}}{\operatorname{argmin}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} (\vec{x}_i^T \vec{w} - y_i)^2}_{g(\vec{w})}$

(Batch) Gradient Descent (BGD)

$\vec{w}^{(t+1)} = \vec{w}^{(t)} - \eta^{(t)} \nabla g(\vec{w}^{(t)})$

$\nabla \hat{L}(\vec{w}^{(t)}) = \left(\frac{2\hat{L}}{2w_0}, \ldots, \frac{2\hat{L}}{2w_d} \right)^T \in \mathbb{R}^{d+1}$

$= \left(\frac{2}{n} \sum_{i=1}^{n} (\vec{x}_i^T \vec{w} - y_i) x_{i0}, \ldots, \frac{2}{n} \sum_{i=1}^{n} (\vec{x}_i^T \vec{w} - y_i) x_{id} \right)^T$

$$= \frac{2}{n} \sum_{i=1}^{n} (\vec{x}_i^T \vec{w} - y_i) \left(x_{i0}, \ldots, x_{id} \right)^T$$

$$= \frac{2}{n} \sum_{i=1}^{n} (\vec{x}_i^T \vec{w} - y_i) \vec{x}_i$$

Algorithm: BGD Linear Regression Learner (with a constant size)

---

input: $D = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$, η, T ← number of "epochs"

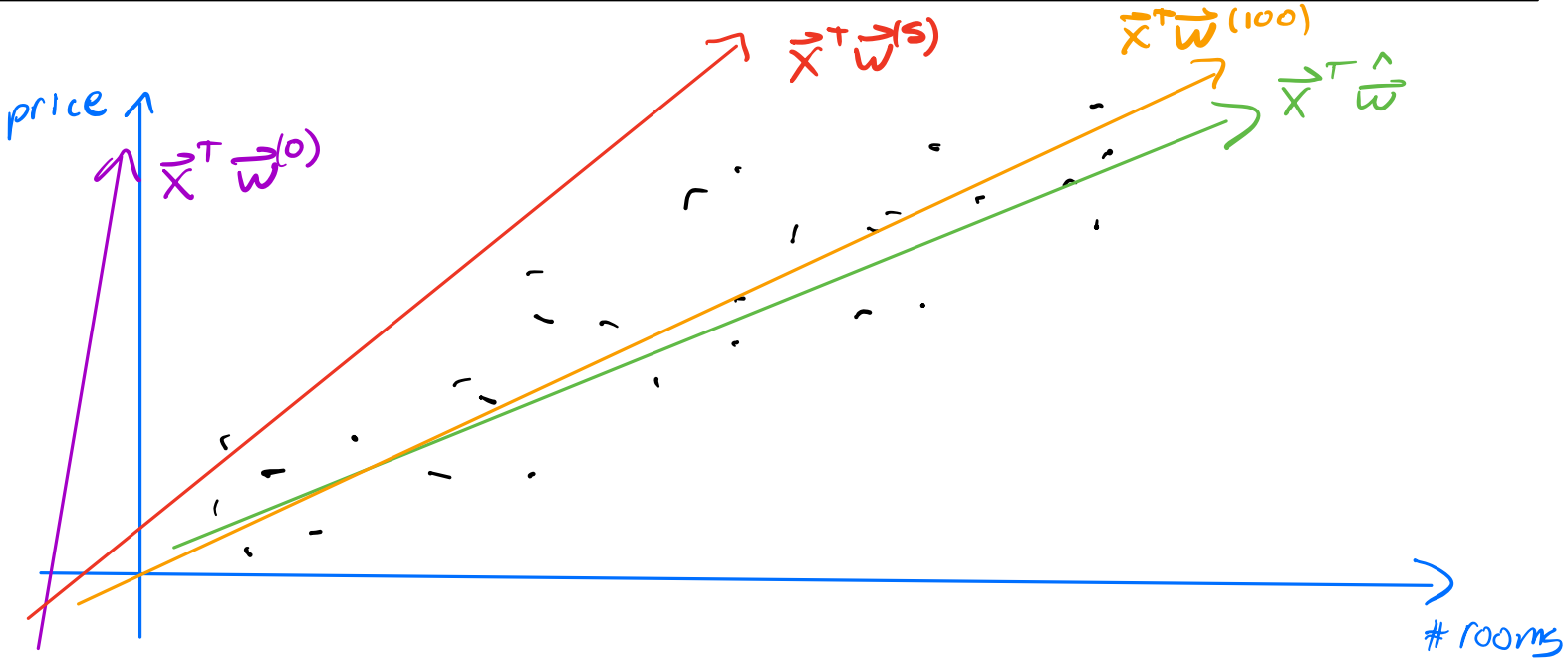
$\vec{w}$ = random vector in $\mathbb{R}^{d+1}$

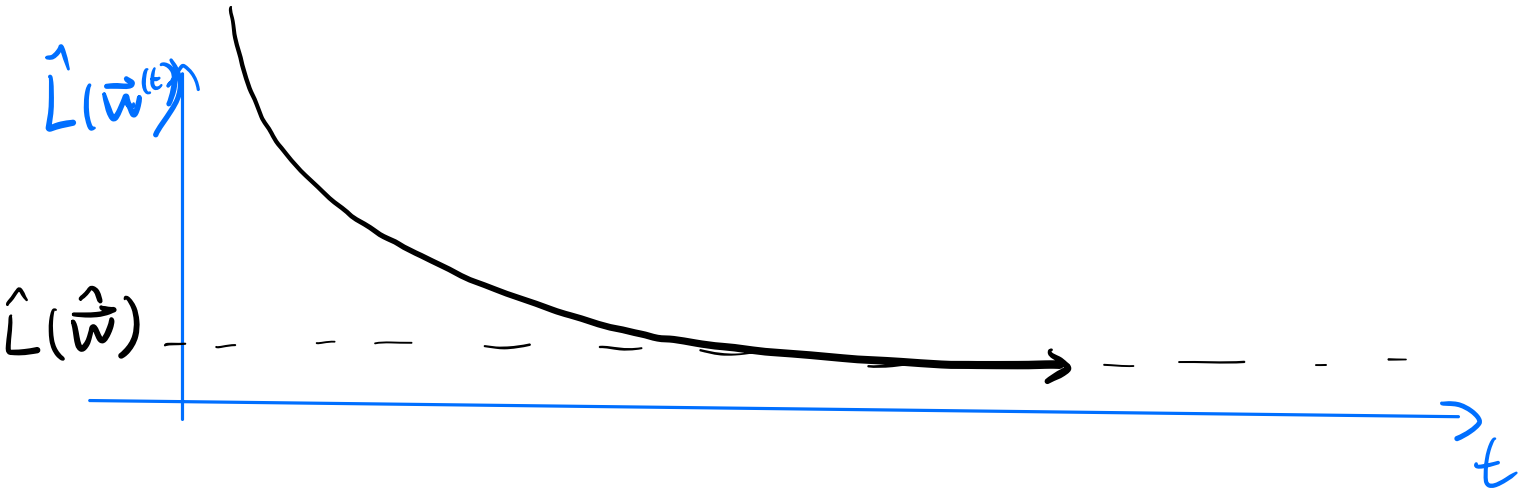
for $t = 1, \ldots, T$

$$\nabla \hat{L}(\vec{w}) = \frac{2}{n} \sum_{i=1}^{n} (\vec{x}_i^T \vec{w} - y_i) \vec{x}_i$$

$$\vec{w} = \vec{w} - \eta \nabla \hat{L}(\vec{w})$$

return $\hat{f} : \mathbb{R}^{d+1} \to \mathbb{R}$ where $\hat{f}(\vec{x}) = \vec{x}^T \vec{w}$





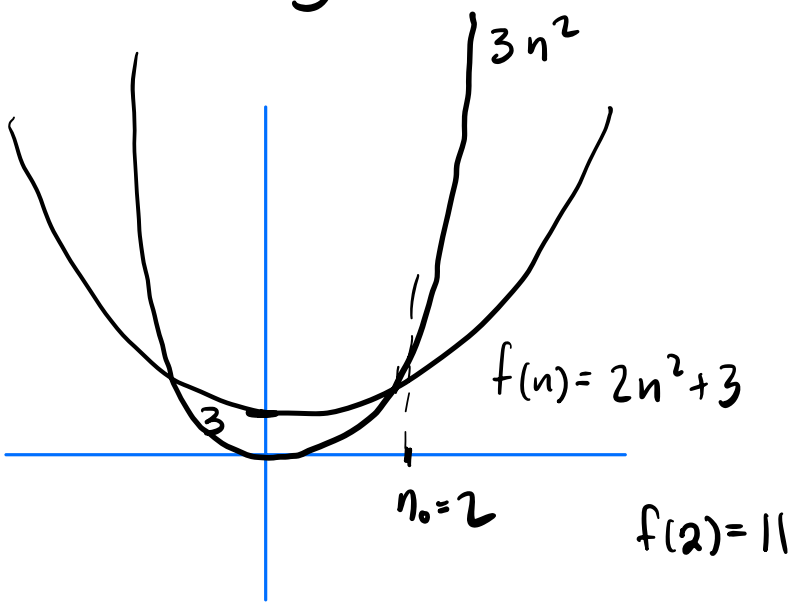
Computation of Closed form soln. vs. BGD

Number of computational steps $\approx$ number of elementary operations (add, subtracted, multiplied, etc.)

Big "O" notation

$f(n) \in O(g(n))$ $\quad f(n) \leq C\, g(n)$ for $n \geq n_0$ ($n \in \mathbb{N}$)

$C \in (0, \infty)$



$f(2) = 11$

Ex: $2n^2 + 3 = O(n^2)$

if $n_0 = 2$ and $C = 3$

then $2(2)^2 + 3 \leq 3(2)^2$

$11 \quad \leq 12$

Closed form: $\hat{\vec{w}} = A^{-1}\vec{b}$ where $A = \sum_{i=1}^{n} \vec{x}_i \vec{x}_i^T$

$\vec{b} = \sum_{i=1}^{n} \vec{x}_i y_i$

Computing $A: O(nd^2)$

$\vec{b}: O(nd)$

$A^{-1}: O(d^3)$

$A^{-1}\vec{b}: O(d^2)$

Total compute: $O(nd^2 + nd + d^3 + d^2)$

$= O(nd^2 + d^3)$

BGD:

computing: $\nabla \hat{L}(\vec{w}): O(nd)$

$\vec{w}$ update: $O(d)$

Total compute: $O(Tnd)$

Comparing $O(Tnd)$ vs. $O(nd^2 + d^3)$

if $T < d$: $O(Tnd) < O(nd^2) \leq O(nd^2 + d^3)$

Mini-Batch Gradient Descent

$$D = \left((\vec{x}_1, y_1), \dots, (\vec{x}_b, y_b), \right.$$
$$(\vec{x}_{b+1}, y_{b+1}), \dots, (\vec{x}_{2b}, y_{2b}),$$
$$\vdots$$
$$\left. (\vec{x}_{(M-1)b+1}, y_{(M-1)b+1}), \dots, (\vec{x}_{Mb}, y_{Mb}), \right)$$

$Mb \leq n$

$b \in \mathbb{N}$: Mini-batch size

$M = \left\lfloor \frac{n}{b} \right\rfloor$: # of mini batches

Ex: $n = 6$, $b = 2$, $M = \frac{6}{2} = 3$

$$D = \left(\begin{array}{ll} (\vec{x}_1, y_1), (\vec{x}_2, y_2), & \text{MB 1} \\ (\vec{x}_3, y_3), (\vec{x}_4, y_4), & \text{MB 2} \\ (\vec{x}_5, y_5), (\vec{x}_6, y_6) & \text{MB 3} \end{array} \right)$$

$$\hat{L}_m(\vec{w}) = \frac{1}{b} \sum_{i=(m-1)b+1}^{mb} (\vec{x}_i^T \vec{w} - y_i)^2 \qquad m \in \{1, \dots, M\}$$

if $b \leq n$ then $\hat{L}_m(\vec{w})$ is a worse estimate than $\hat{L}(\vec{w})$

Algorithm: MBGD Linear Regression Learner (with a constant size)

---

input: $D = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$, η, T, b

$\vec{w}$ = random vector in $\mathbb{R}^{d+1}$

$M = \text{floor}\left(\frac{n}{b}\right)$

for $t = 1, \ldots, T$

Randomly shuffle D

for $m = 1, \ldots, M$

$$\nabla \hat{L}(\vec{w}) = \frac{2}{b} \sum_{i=(m-1)b+1}^{mb} (\vec{x}_i^T \vec{w} - y_i) \vec{x}_i$$

$$\vec{w} = \vec{w} - \eta \nabla \hat{L}(\vec{w})$$

return $\hat{f}: \mathbb{R}^{d+1} \to \mathbb{R}$ where $\hat{f}(\vec{x}) = \vec{x}^T \vec{w}$

---

if $b = n \Rightarrow M = 1 \Rightarrow$ BGD

if $b = 1 \Rightarrow M = n \Rightarrow$ SGD "stochastic gradient descent"

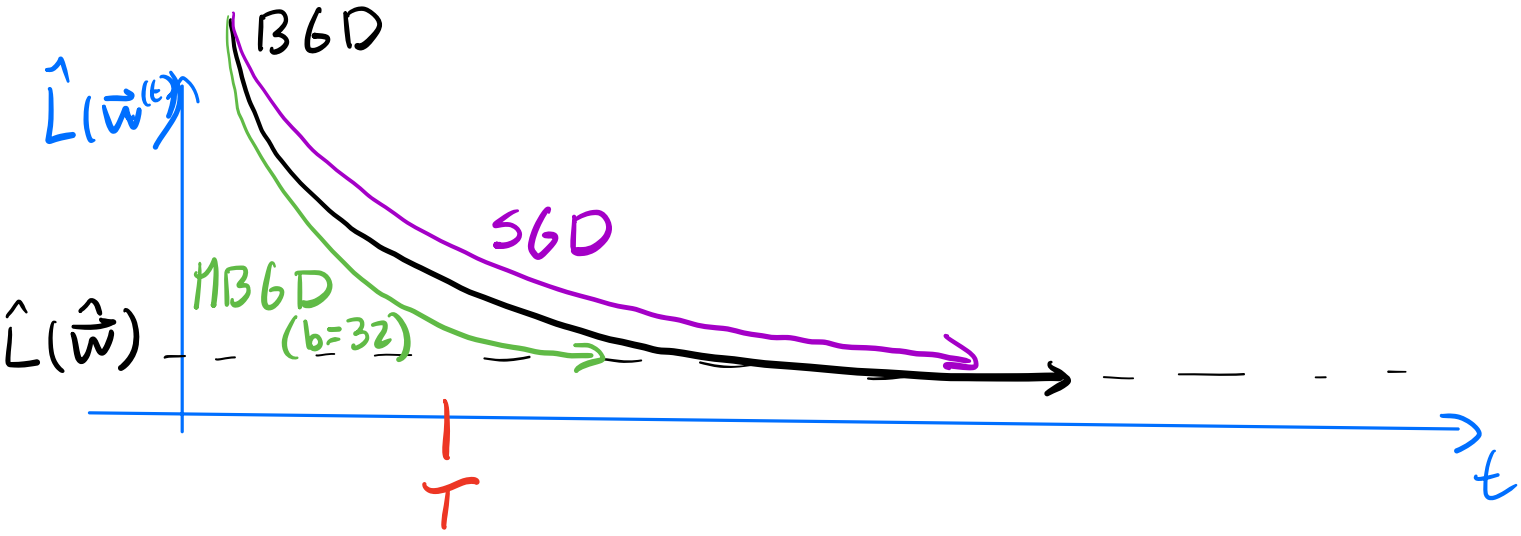
Computation

MBGD : $O(TMbd)$

BGD : $O(Tnd)$

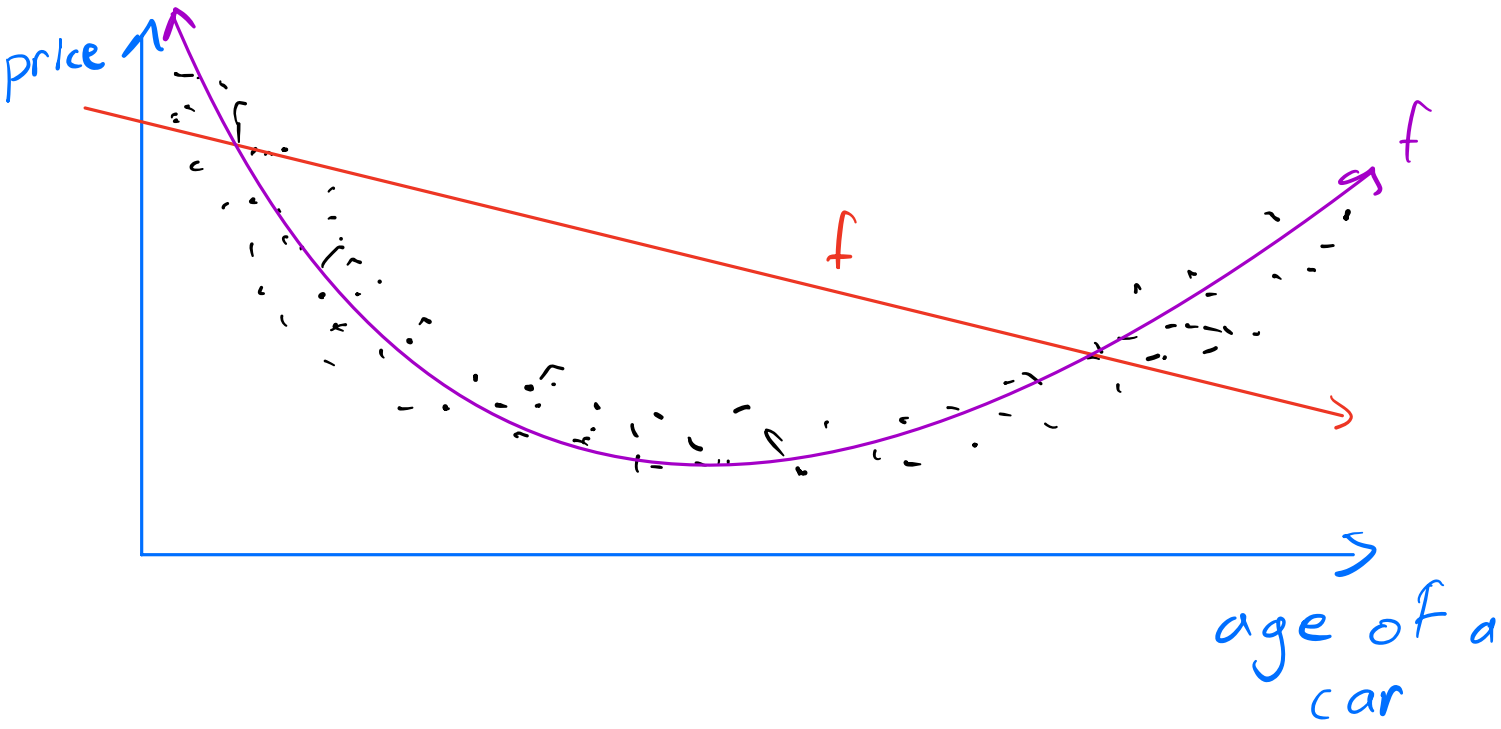
In practice for the same T

MBGD usually finds a better $\vec{w}$ (case b<n)



Polynomial Regression

$\hat{L}(f) < \hat{L}(f)$



Ex: $d = 1$, $\mathcal{X} = \mathbb{R}^{d+1} = \mathbb{R}^2$, $\mathcal{Y} = \mathbb{R}$

linear function: $\vec{x}^T \vec{w} = (1, x_1)(w_0, w_1)^T = w_0 + w_1 x_1$

$$F_1 = \{f \mid f : \mathbb{R}^2 \to \mathbb{R} \text{ s.t } f(\vec{x}) = \underbrace{\vec{x}^T \vec{w}}_{\phi_1(\vec{x})}, \vec{w} \in \mathbb{R}^2\}$$

degree 2 polynomial: $w_0 + w_1 x_1 + w_2 x_1^2$

$= (1, x_1, x_1^2)(w_0, w_1, w_2)^T$

feature map $\rightarrow$ $= \phi_2(\vec{x})^T (w_0, w_1, w_2)^T$

$$F_2 = \{f \mid f : \mathbb{R}^2 \to \mathbb{R} \text{ s.t } f(\vec{x}) = \phi_2(\vec{x})^T \vec{w}, \vec{w} \in \mathbb{R}^3\}$$

degree 3 polynomial: $w_0 + w_1 x_1 + w_2 x_1^2 + w_3 x_1^3$

$$= (1, x_1, x_1^2, x_1^3)(w_0, w_1, w_2, w_3)^T$$

$$= \phi_3(\vec{x})^T (w_0, w_1, w_2, w_3)^T$$

$$\mathcal{F}_3 = \{f \mid f : \mathbb{R}^2 \to \mathbb{R} \text{ s.t } f(\vec{x}) = \phi_3(\vec{x})^T \vec{w}, \vec{w} \in \mathbb{R}^4\}$$

Ex: $d = 2$, $\mathcal{X} = \mathbb{R}^{d+1} = \mathbb{R}^3$, $\mathcal{Y} = \mathbb{R}$

linear function: $\vec{x}^T \vec{w} = (1, x_1, x_2)(w_0, w_1, w_2)^T$

$$= w_0 + w_1 x_1 + w_2 x_2$$

$$\mathcal{F}_1 = \{f \mid f : \mathbb{R}^3 \to \mathbb{R} \text{ s.t } f(\vec{x}) = \vec{x}^T \vec{w}, \vec{w} \in \mathbb{R}^3\}$$

degree 2 polynomial:

$$w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 + w_5 x_1 x_2$$

$$= (1, x_1, x_2, x_1^2, x_2^2, x_1 x_2)(w_0, w_1, w_2, w_3, w_4, w_5)^T$$

$$= \phi_2(\vec{x})^T (w_0, w_1, w_2, w_3, w_4, w_5)^T$$

$$\mathcal{F}_2 = \{f \mid f : \mathbb{R}^3 \to \mathbb{R} \text{ s.t } f(\vec{x}) = \phi_2(\vec{x})^T \vec{w}, \vec{w} \in \mathbb{R}^6\}$$

$$\mathcal{F}_1 \subsetneq \mathcal{F}_2 \subsetneq \mathcal{F}_3 \cdots$$

In general $\phi_p : \mathcal{X} \to \mathcal{Z}$ $\quad p \in \mathbb{N}$

if $\mathcal{X} = \mathbb{R}^{d+1}$, $\mathcal{Z} = \mathbb{R}^{\bar{p}}$ where $\bar{p} = \binom{d+p}{p}$

Ex: $d = 2$, $p = 2 \Rightarrow \bar{p} = \binom{2+2}{2} = \frac{4 \cdot 3}{2} = 6$

$$\mathcal{F}_p = \{ f \mid f : \mathbb{R}^{d+1} \to \mathbb{R} \text{ s.t } f(\vec{x}) = \phi_p(\vec{x})^T \vec{w}, \vec{w} \in \mathbb{R}^{\bar{p}} \}$$

$$\mathcal{A}_p(D) = \hat{f}_p = \underset{f \in \mathcal{F}_p}{\operatorname{argmin}} \hat{L}(\hat{f}_p)$$

$$\min_{f \in \mathcal{F}_p} \hat{L}(\hat{f}_p) = \min_{\vec{w} \in \mathbb{R}^{\bar{p}}} \frac{1}{n} \sum_{i=1}^{n} \ell(\phi_p(\vec{x}_i)^T \vec{w}, y_i)$$

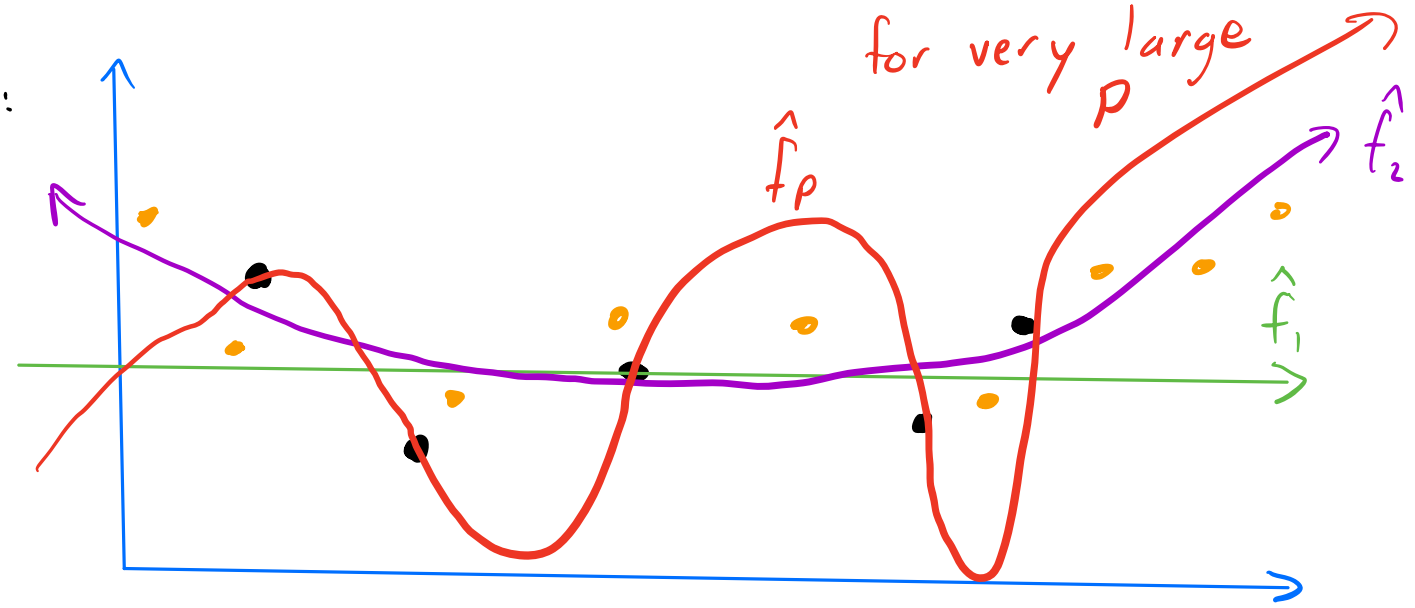
$$\Rightarrow \hat{f}_p(\vec{x}) = \phi_p(\vec{x})^T \hat{\vec{w}} \quad \text{where}$$

$$\hat{\vec{w}} = \arg \min_{\vec{w} \in \mathbb{R}^{\bar{p}}} \frac{1}{n} \sum_{i=1}^{n} \ell(\phi_p(\vec{x}_i)^T \vec{w}, y_i)$$

Solve for $\hat{\vec{w}}$ using Closed form Soln
or BGD, MBGD, SGD

as p increase computation increases

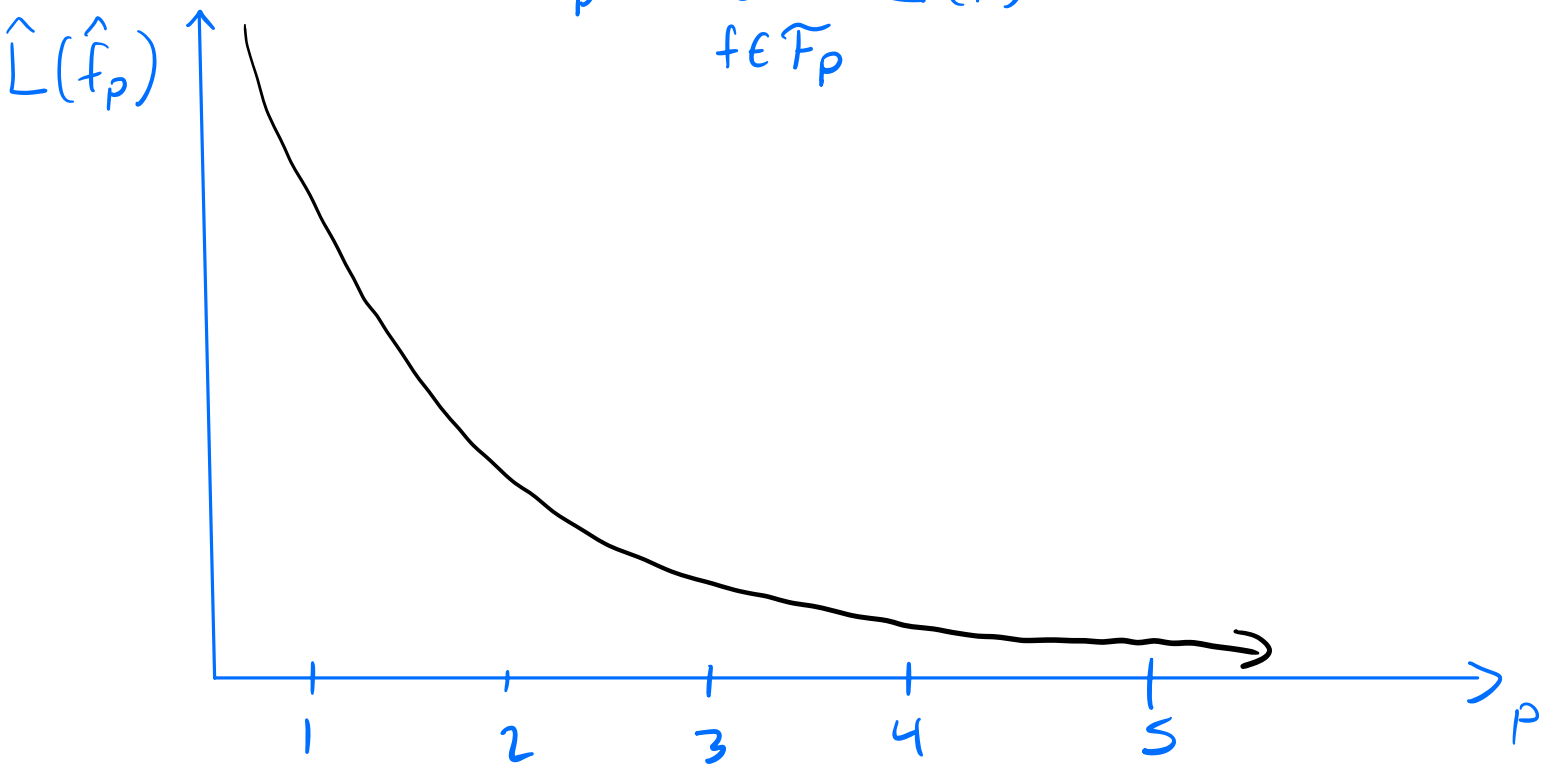
Ex:



$$\hat{L}(\hat{f}_1) \geq \hat{L}(\hat{f}_2) \geq \cdots \geq \hat{L}(\hat{f}_p) \approx 0$$

$$\tilde{F}_1 \subsetneq \tilde{F}_2 \subsetneq \tilde{F}_3 \subsetneq \cdots$$

$$\hat{f}_p = \underset{f \in \tilde{F}_p}{\operatorname{argmin}} \hat{L}(f)$$



$X_1 \in \{1, \ldots 3\} \subseteq \mathbb{R}$ # of rooms

$X_2 \in [0, 30] \subseteq \mathbb{R}$ distance to downtown

$$200 X_1 + 300 X_1 \left(\frac{30 - X_2}{30}\right)$$

if $X_1 = 1$, $X_2 = 30$: 200

$X_2 = 15$: 350

$X_2 = 0$: 500

$X_1 = 5$, $X_2 = 30$: 1000

$X_2 = 15$: 1750

$X_2 = 0$: 2500