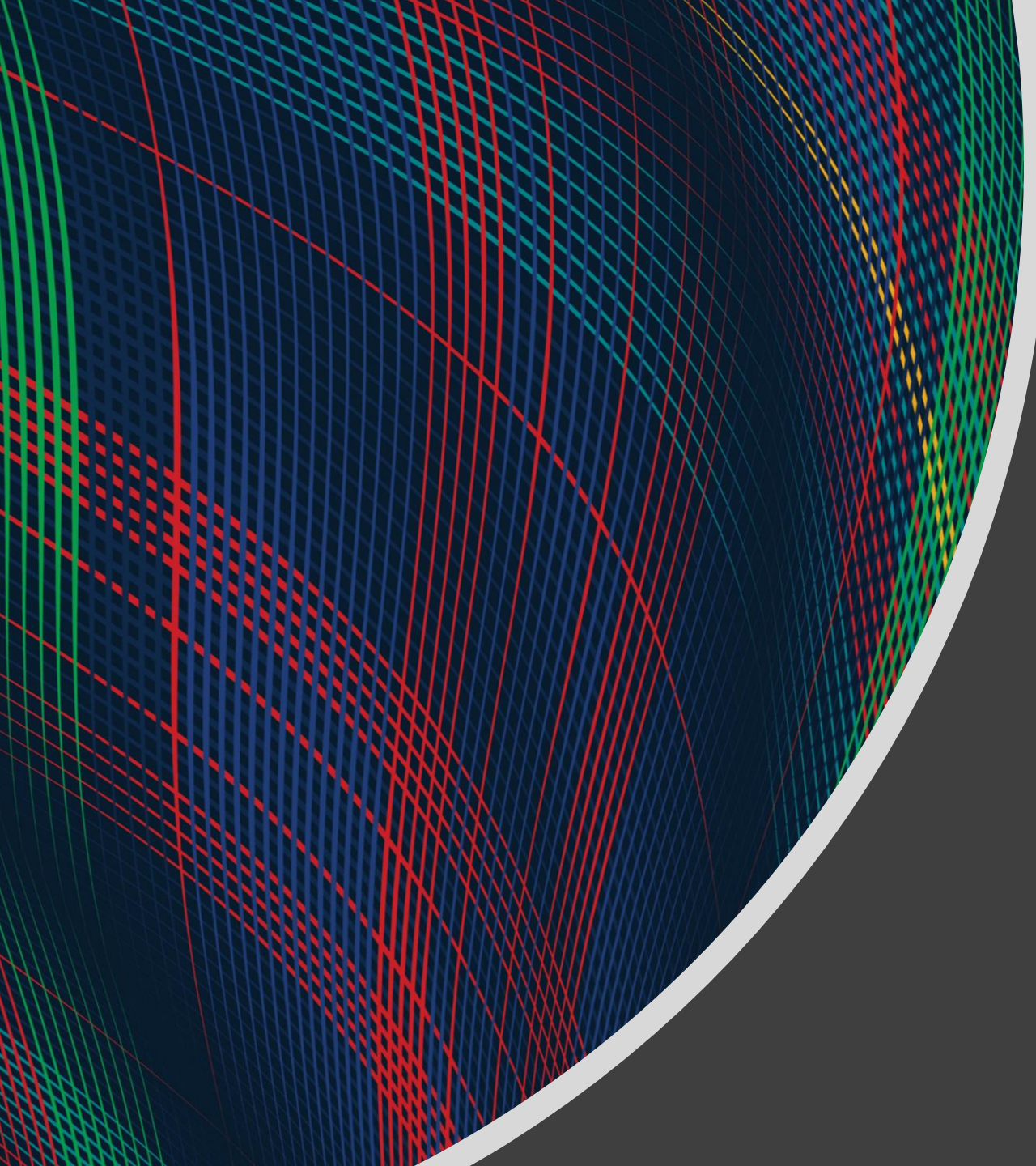


An abstract graphic on the left side of the slide, featuring a sphere-like shape composed of a dense grid of intersecting red, green, and blue lines. The lines are curved and follow the contour of the sphere, creating a complex, woven pattern. The sphere is set against a dark gray background.

10-315
Introduction to ML

Optimization and
Linear Regression

Instructor: Pat Virtue

Pre-reading: Optimization Formulation

Find the best $h(x) \rightarrow \hat{y}$ by searching in the space of hypothesis functions $h \in \mathcal{H}$.

Loss: $\ell(y, \hat{y})$ is the loss or cost of predicting $\hat{y}$ when the true value is y .

Risk: Risk for a hypothesis function is the expected loss over the data distribution:

$$R(h) = \mathbb{E}_{XY}[\ell(Y, h(X))]$$

Minimizing Risk

Specifically: find a prediction function $h^*: \mathcal{X} \rightarrow \mathcal{Y}$ that minimizes the expected loss for randomly drawn test data (X, Y)

$$h^* = \underset{h}{\operatorname{argmin}} \mathbb{E}_{XY}[\ell(Y, h(X))]$$

Pre-reading: Empirical Risk Minimization

Risk

$$R(h) = \mathbb{E}_{X,Y} [\ell(Y, h(X))]$$

Empirical Risk

$$\hat{R}(h) = \frac{1}{N} \sum_{i=1}^N \ell \left(y^{(i)}, h \left(x^{(i)} \right) \right) \quad \mathcal{D} = \{ (x^{(i)}, y^{(i)}) \}_{i=1}^N$$

Empirical Risk Minimization

$$h^* = \operatorname{argmin}_{h \in \mathcal{H}} \hat{R}(h)$$

Pre-reading: Loss Functions

$$h^* = \operatorname{argmin}_h \mathbb{E}_{XY} [L(Y, h(X))]$$

Loss function:

$$L: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$$

Classification:

- Two-class, 0,1 loss
- Two-class, arbitrary loss

False positives and false negatives:

Poll (unused)

For a healthcare classification problem where:

$Y=0$ represents a negative test for cancer (healthy)

$Y=1$ represents a positive test for cancer (sick)

Which two-class loss value should be highest?

- A. True negative
- B. False negative
- C. False positive
- D. True positive

Poll (unused)

Does our decision tree algorithm (recursively finding the best attribute to split on at each node) find a hypothesis that minimizes empirical risk for zero-one loss?

- A. Yes, always
- B. Yes, but only when splitting with error rate
- C. Yes, but only when splitting with mutual information
- D. No

$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$$

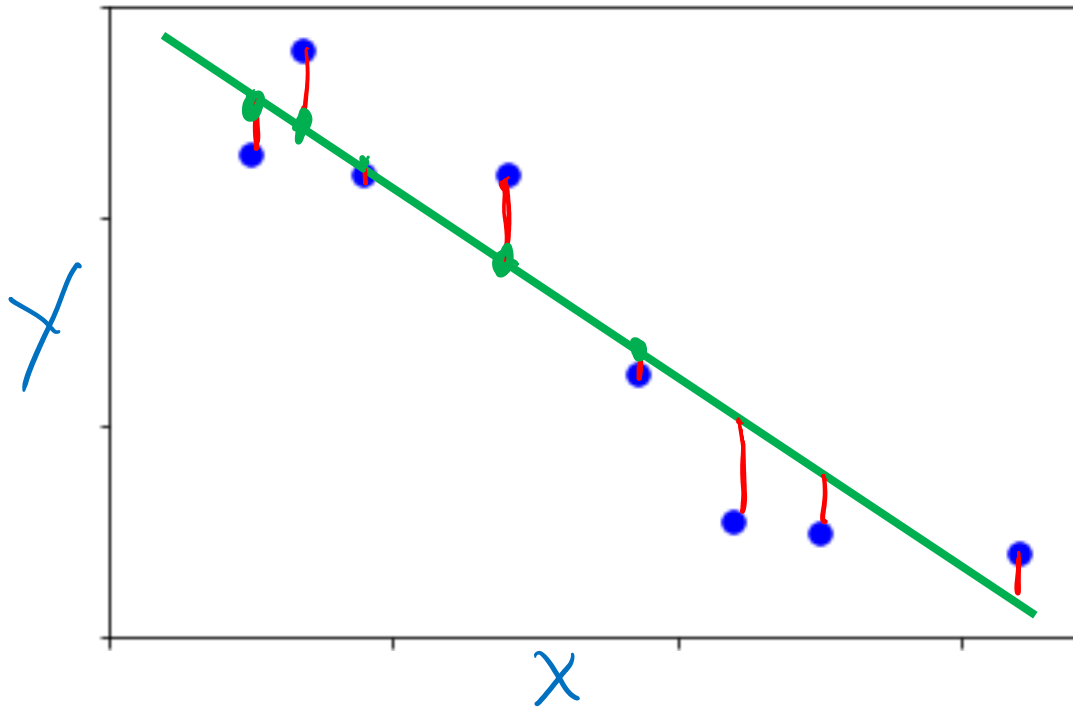
$$\hat{R}(h) = \frac{1}{N} \sum_{i=1}^N \ell(y^{(i)}, h(x^{(i)}))$$

$$h^* = \operatorname{argmin}_{h \in \mathcal{H}} \hat{R}(h)$$

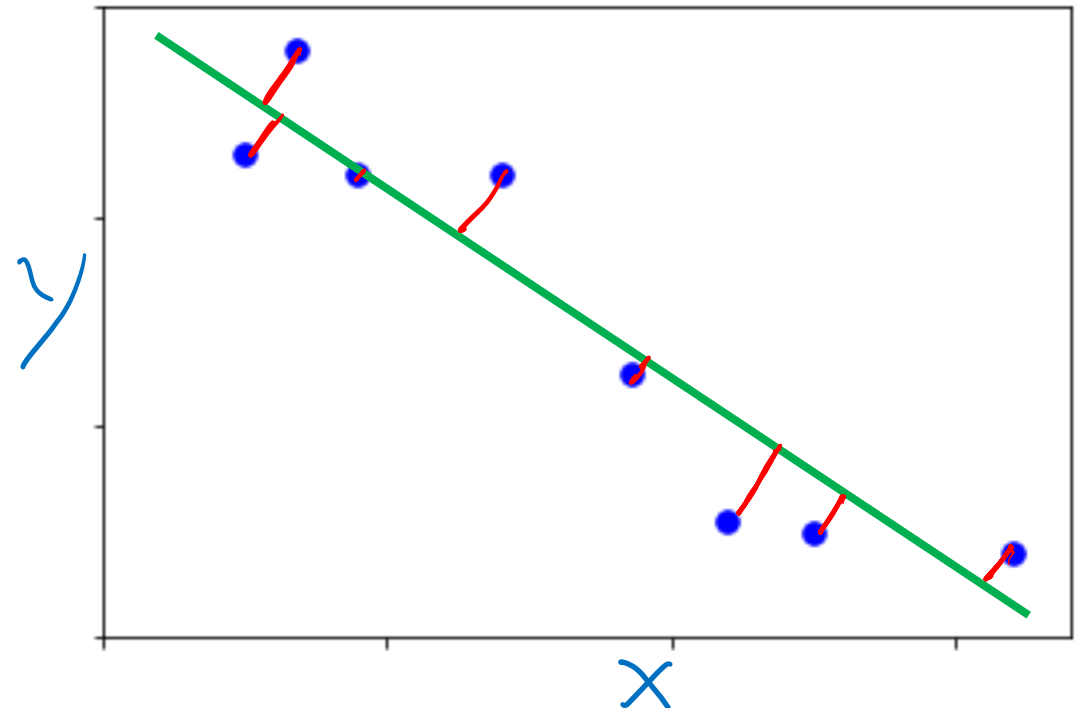
Poll 1

Which is a correct depiction of linear regression error lines for $x \in \mathbb{R}$,
 $\hat{y} = wx + b$

A



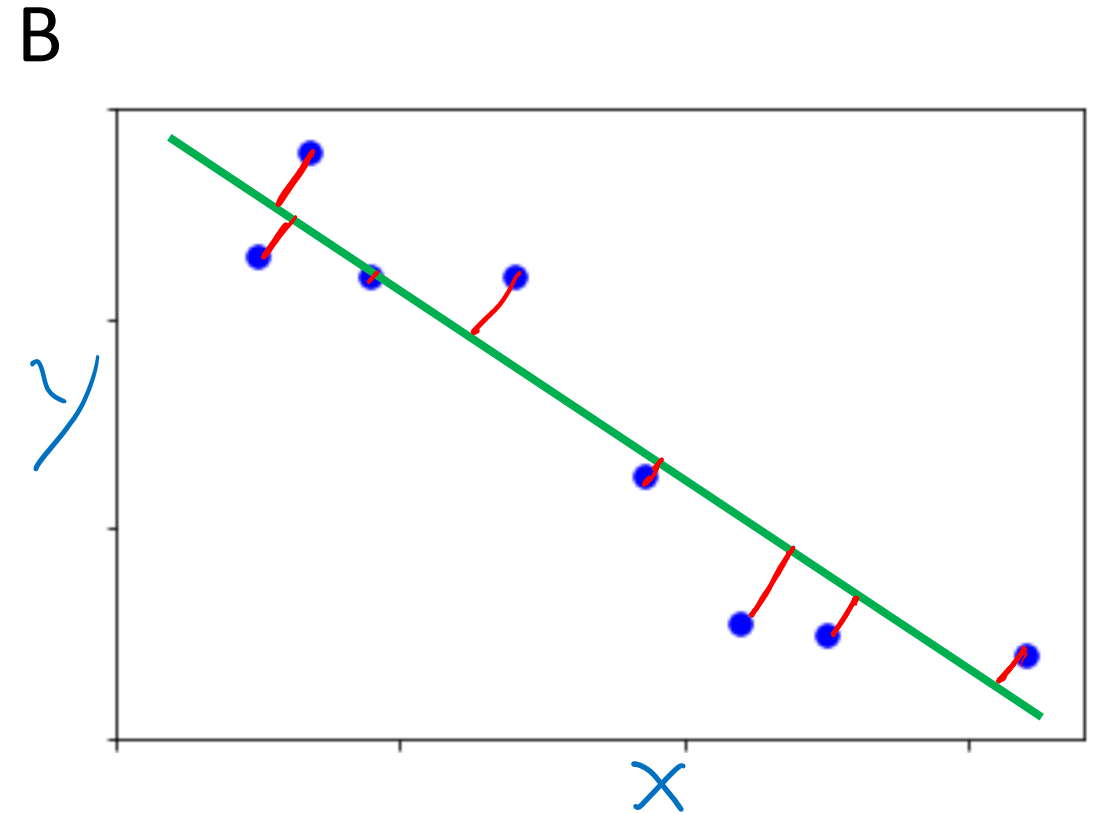
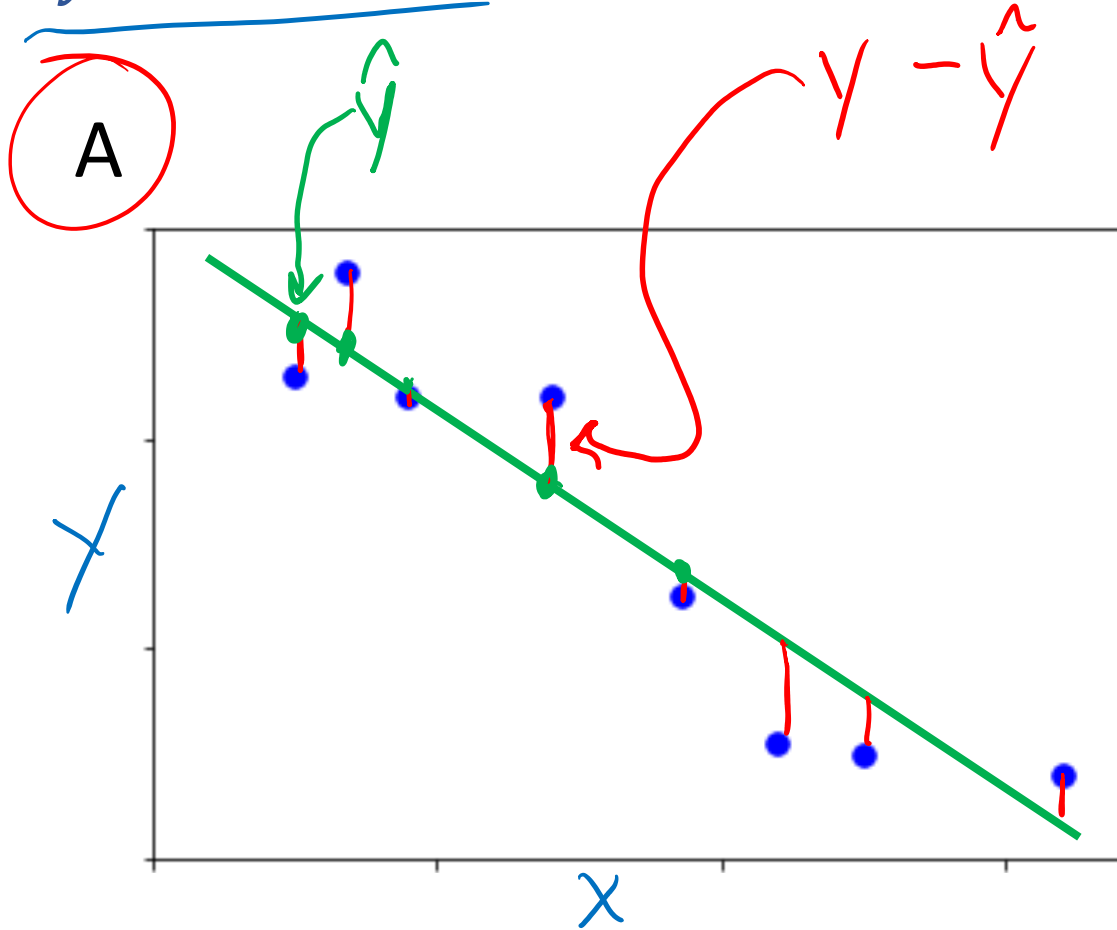
B



Poll 1

Which is a correct depiction of linear regression error lines for $x \in \mathbb{R}$,

$\hat{y} = wx + b$



Poll 2

Given $\theta = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}$, what goes in the top entry for a linear regression

vector with two features, $\mathbf{x} = \begin{bmatrix} ? \\ x_1 \\ x_2 \end{bmatrix}$?

Poll 2

Given $\theta = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}$, what goes in the top entry for a linear regression

vector with two features, $\mathbf{x} = \begin{bmatrix} ? \\ x_1 \\ x_2 \end{bmatrix}$?

$$y = mx + b$$

↑ ↑
slope intercept

bias term
(intercept)
offset

$$b + \vec{w}^T \vec{x}$$

weights
(slope)

$$\hat{y} = h(\vec{x})$$

$$= b + x_1 w_1 + x_2 w_2$$

$$= \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}^T \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} = \vec{\theta}^T \vec{x}$$

$$= b + \sum_{j=1}^n x_j w_j$$

Poll 3: Two questions (unused)

Fill in the blanks: Linear regression optimization

1) min/max/argmin/argmax?

2) ?

$$\frac{1}{N} \sum_{i=1}^N (y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)})^2$$

1) What type of optimization?

- A. min
- B. max
- C. argmin
- D. argmax

2) What are we optimizing over?

- A. x
- B. y
- C. θ
- D. x, θ
- E. x, y
- F. x, y, θ

Poll 3: Two questions (unused)

Fill in the blanks: Linear regression optimization

1) min/max/argmin/argmax?

2) ?

$$\frac{1}{N} \sum_{i=1}^N (y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)})^2$$

1) What type of optimization?

- A. min
- B. max
- ☒ C. argmin
- D. argmax

2) What are we optimizing over?

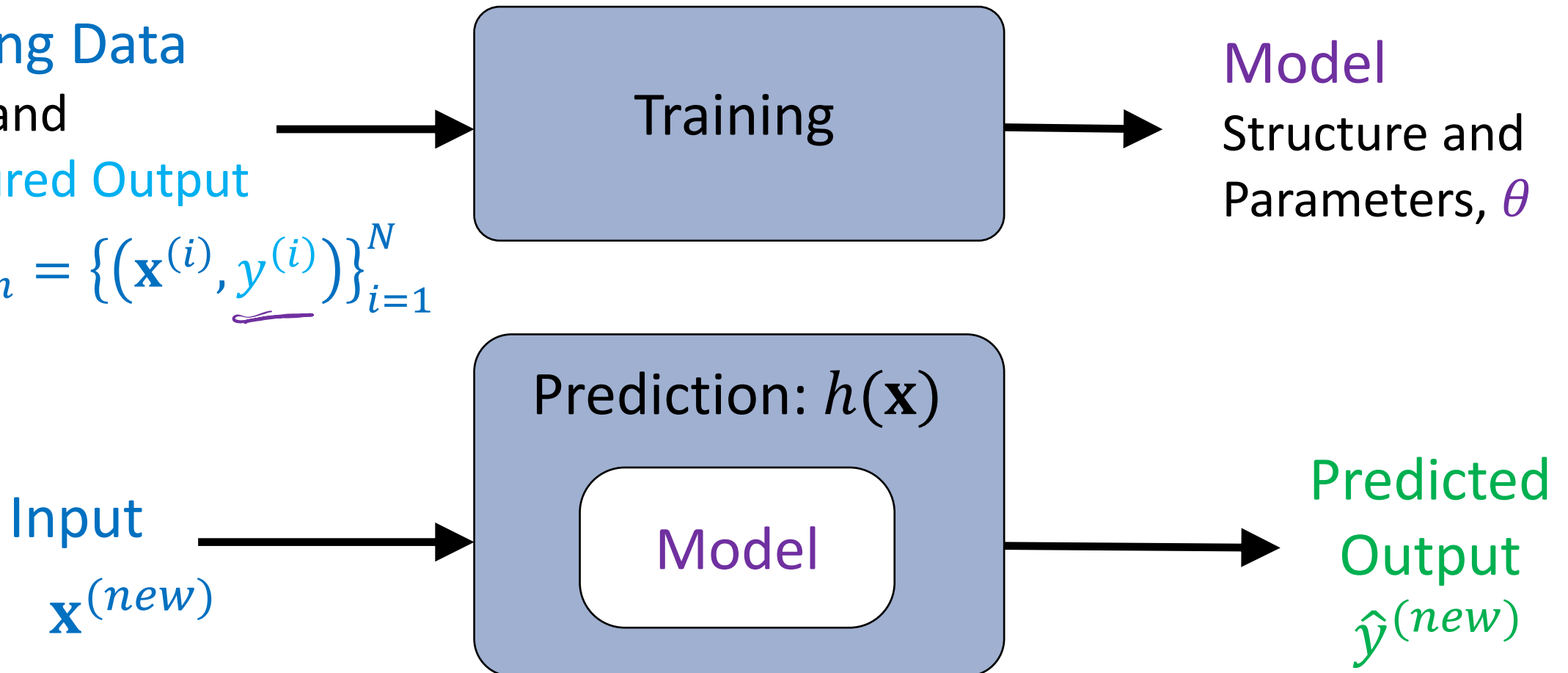
- A. x
- B. y
- ☒ C. θ
- D. x, θ
- E. x, y
- F. x, y, θ

Machine Learning

Using (training) data to learn a model that we'll later use for prediction

Training Data
Input and
Measured Output

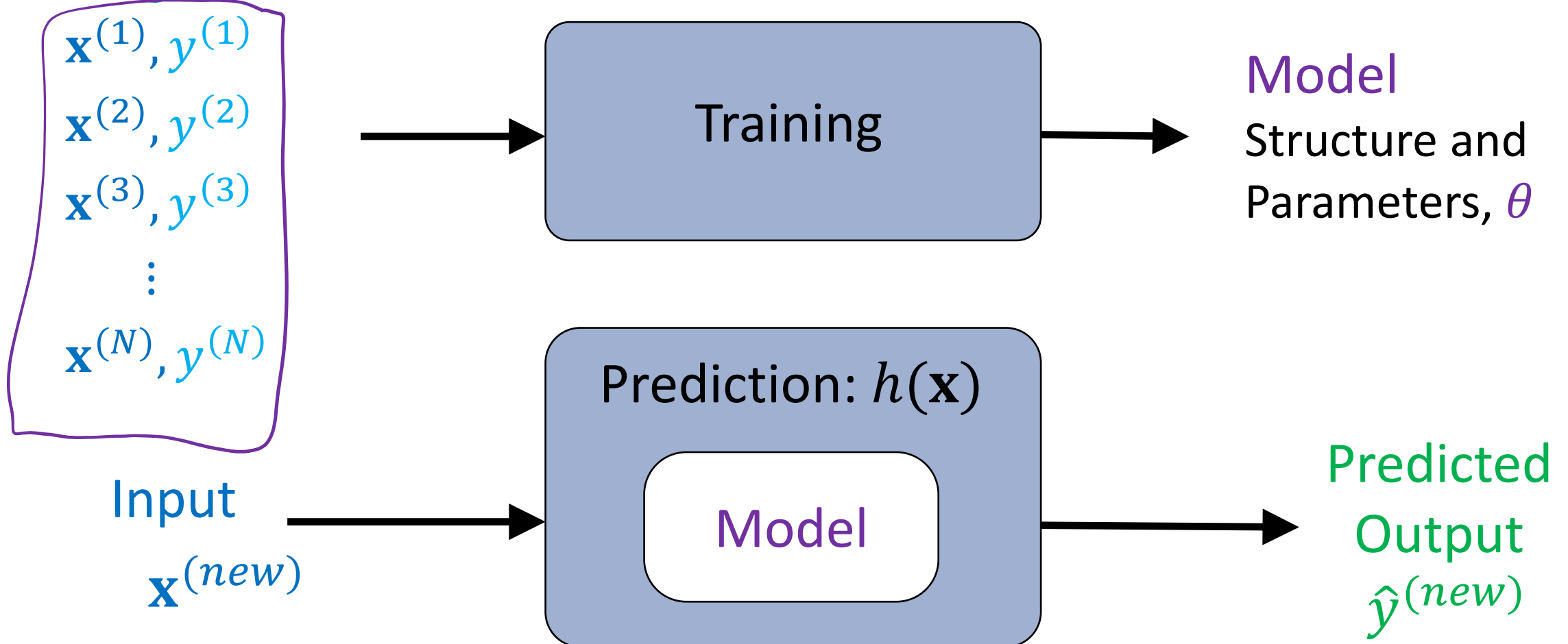
$$\mathcal{D}_{train} = \{(\mathbf{x}^{(i)}, \underline{y}^{(i)})\}_{i=1}^N$$



Machine Learning

Using (training) data to learn a model that we'll later use for prediction

Training Data



Machine Learning

Using (training) data to learn a model that we'll later use for prediction

Training Data

$$\mathbf{x}^{(1)}, y^{(1)}$$

$$\mathbf{x}^{(2)}, y^{(2)}$$

$$\mathbf{x}^{(3)}, y^{(3)}$$

$\vdots$

$$\mathbf{x}^{(N)}, y^{(N)}$$



Model

Structure:

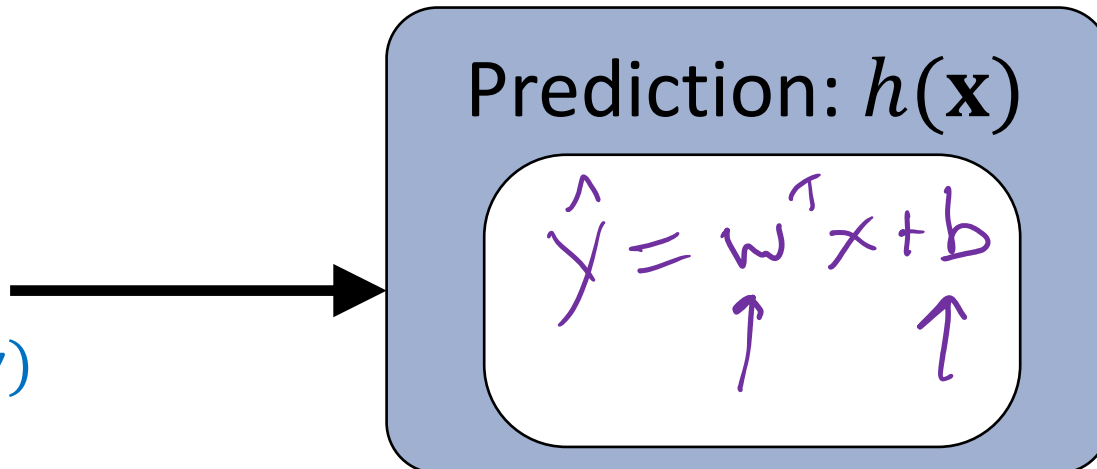
✓ $y = \vec{w}^T \vec{x} + b$

Parameters, θ :

$\vec{w}, b$

Input

$$\mathbf{x}^{(new)}$$



Predicted

Output

$$\hat{y}^{(new)}$$

Machine Learning

Using (training) data to learn a model that we'll later use for prediction

Training Data

$\mathbf{x}^{(1)}, y^{(1)}$

$\mathbf{x}^{(2)}, y^{(2)}$

$\mathbf{x}^{(3)}, y^{(3)}$

$\vdots$

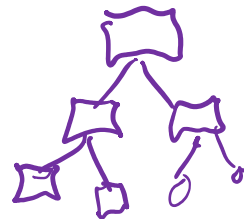
$\mathbf{x}^{(N)}, y^{(N)}$

Input

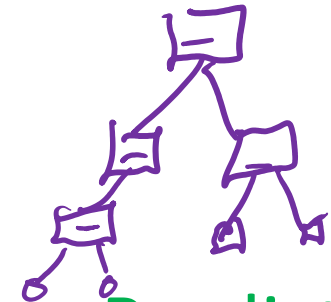
$\mathbf{x}^{(new)}$

Training

Prediction: $h(\mathbf{x})$



Model
Structure and
Parameters, θ



Predicted
Output
 $\hat{y}^{(new)}$

DT

Machine Learning

Using (training) data to learn a model that we'll later use for prediction

Training Data

$\mathbf{x}^{(1)}, y^{(1)}$

$\mathbf{x}^{(2)}, y^{(2)}$

$\mathbf{x}^{(3)}, y^{(3)}$

$\vdots$

$\mathbf{x}^{(N)}, y^{(N)}$

Input

$\mathbf{x}^{(new)}$

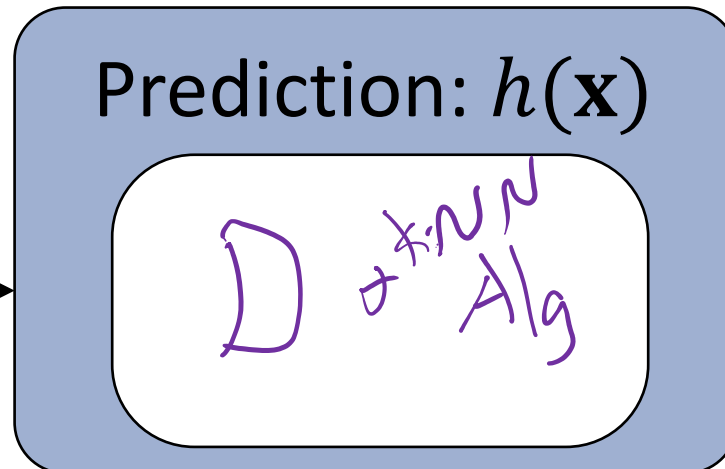


Nearest Neighbor

Model

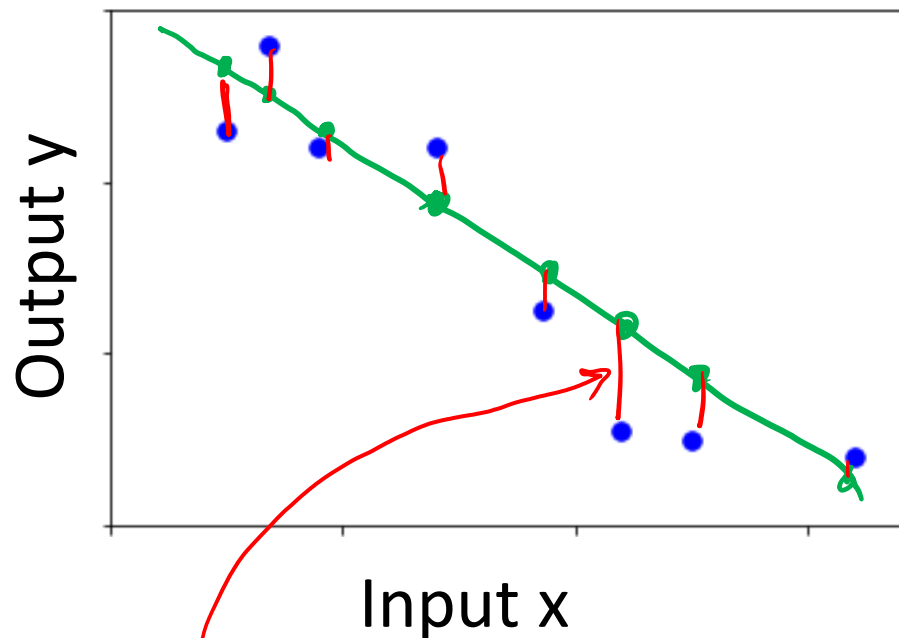
Structure and
Parameters, θ

D



Predicted
Output
 $\hat{y}^{(new)}$

Linear Regression



Regression

$$x \in \mathbb{R}$$

$$y \in \mathbb{R}$$

$$\hat{y} = h(x) = \underline{w}x + \underline{b}$$

$$\hat{y}^{(i)} = h(x^{(i)})$$

Error $y - \hat{y}$

$$\frac{1}{N} \sum (y^{(i)} - \hat{y}^{(i)})^2$$

mean sq error

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

Squared error loss

$$\hat{R}(h) = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - \hat{y}^{(i)})^2$$

Risk in Regression

Squared error loss $\ell(y, h(x)) = (y - h(x))^2$

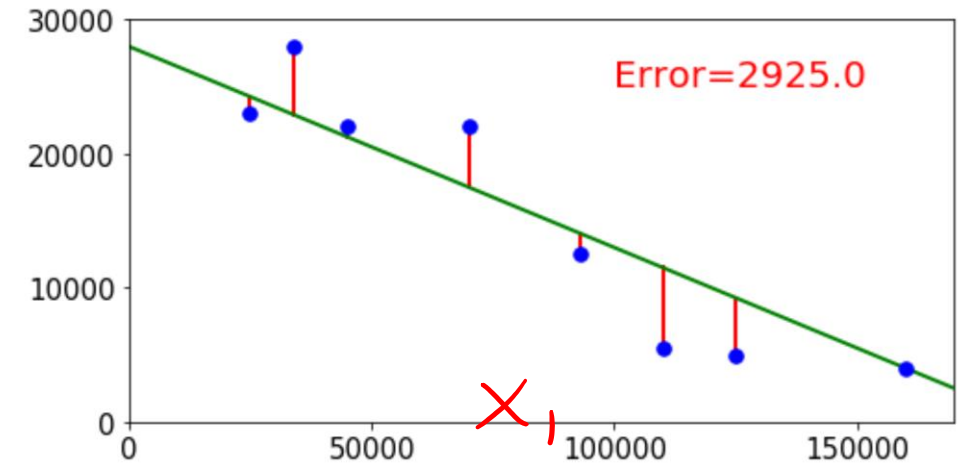
$$h^* = \operatorname{argmin}_h \mathbb{E}_{XY}[\ell(Y, h(X))]$$

$$\begin{aligned}\hat{R}(h) &= \frac{1}{N} \sum_{i=1}^N \ell\left(y^{(i)}, h\left(x^{(i)}\right)\right) \\ &= \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - h\left(x^{(i)}\right)\right)^2 \quad \leftarrow \text{Mean squared error!} \\ &= \frac{1}{N} \sum_{i=1}^N \overbrace{\left(y^{(i)} - \underbrace{\left(wx^{(i)} + b\right)}_{\hat{y}^{(i)}}\right)}^{\text{loss}}^2\end{aligned}$$

Linear Regression Optimization

Finding the parameter values that minimize the mean squared error objective function

$$x \in \mathbb{R}$$
$$J(w, b) = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - \hat{y}^{(i)})^2$$
$$\hat{y}^{(i)} = wx^{(i)} + b$$



$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$$

$$h^* = \operatorname{argmin}_{h \in \mathcal{H}} \hat{R}(h) \quad \checkmark \quad J(w, b)$$

$$\begin{aligned} \hat{R}(h) &= \frac{1}{N} \sum_{i=1}^N \ell(y^{(i)}, h(x^{(i)})) \\ &= \frac{1}{N} \sum_{i=1}^N (y^{(i)} - h(x^{(i)}))^2 \\ &= \frac{1}{N} \sum_{i=1}^N (y^{(i)} - (wx^{(i)} + b))^2 \end{aligned}$$

Linear in Higher Dimensions

1-D $y = wx + b$
2-D $y = w_1 x_1 + w_2 x_2 + b$

What are these linear shapes called for 1-D, 2-D, 3-D, M-D input?

	$x \in \mathbb{R}$	$x \in \mathbb{R}^2$	$x \in \mathbb{R}^3$	$x \in \mathbb{R}^M$
$y = \mathbf{w}^T \mathbf{x} + b$	line	plane	hyperplane	hyperplane

$\mathbf{w}^T \mathbf{x} + b = 0$	point	line	plane	hyperplane
-----------------------------------	-------	------	-------	------------

$\mathbf{w}^T \mathbf{x} + b \geq 0$	halfline	halfplane	halfspace	halfspace
--------------------------------------	----------	-----------	-----------	-----------

Linear Regression Optimzation

Finding the parameter values that minimize the mean squared error objective function

$x \in \mathbb{R}$

$$J(w, b) = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - \hat{y}^{(i)})^2$$

$$\hat{y}^{(i)} = wx^{(i)} + b$$

$\vec{x} \in \mathbb{R}^2$

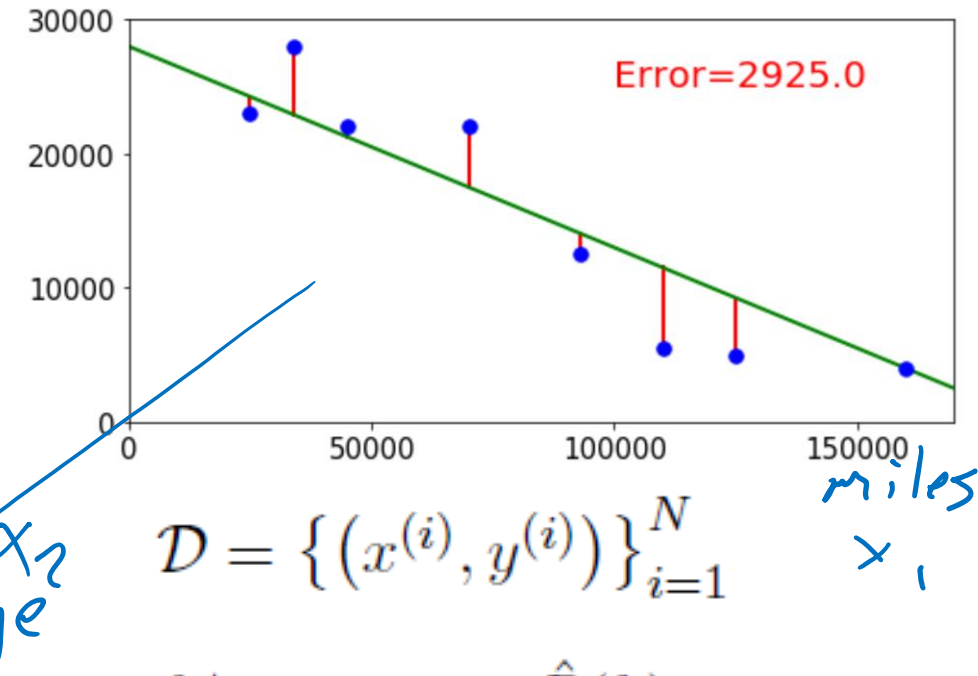
$$J(w_1, w_2, b) = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - \hat{y}^{(i)})^2 \leftarrow$$

$$\hat{y}^{(i)} = w_1 x_1^{(i)} + w_2 x_2^{(i)} + b$$

$\vec{x} \in \mathbb{R}^M$

$$J(w_1, \dots, w_M, b) =$$

$$\hat{y}^{(i)} = b + \sum_{j=1}^M w_j x_j^{(i)} = \vec{\theta}^T \vec{x}$$



$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$$

$$h^* = \operatorname{argmin}_{h \in \mathcal{H}} \hat{R}(h)$$

$$\hat{R}(h) = \frac{1}{N} \sum_{i=1}^N \ell(y^{(i)}, h(x^{(i)}))$$

$$= \frac{1}{N} \sum_{i=1}^N (y^{(i)} - h(x^{(i)}))^2$$

$$= \frac{1}{N} \sum_{i=1}^N (y^{(i)} - (wx^{(i)} + b))^2$$

Constructing the Design Matrix

Linear Algebra Formulation

Constructing the design matrix

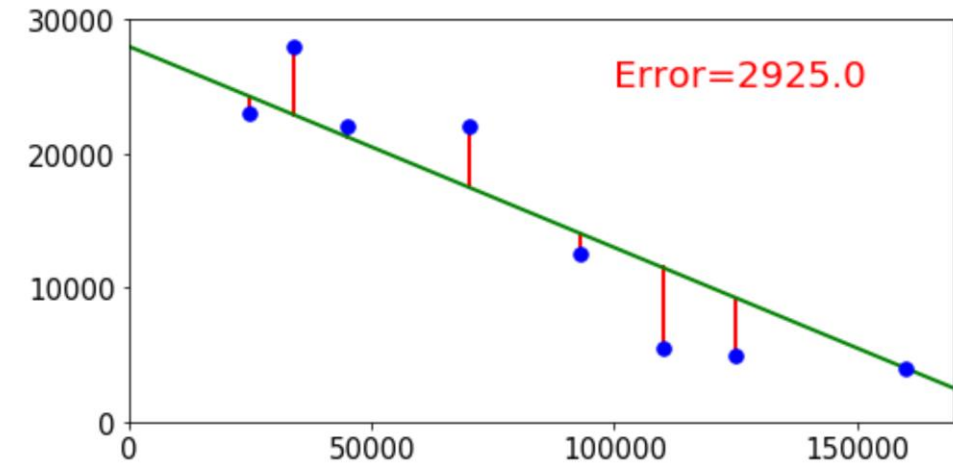
Goal:
Replace summations w/ lin. alg.

$$J(w, b) = \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - \left(b + \sum_{j=1}^M w_j x_j^{(i)} \right) \right)^2$$

$$J(\vec{\theta}) = J(w, b) = \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - \vec{\theta}^T \vec{x}^{(i)} \right)^2$$

$$J(\vec{\theta}) = J(w, b) = \frac{1}{N} \| \vec{y} - X \vec{\theta} \|_2^2$$

vector of y data $\nearrow$ $\nwarrow$ Design matrix



$$\theta = \begin{bmatrix} b \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

$$\vec{X} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$M=5$

Linear Algebra Formulation

Constructing the design matrix

Goal:
Replace summations w/ lin. alg.

$$J(w, b) = \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - \left(b + \sum_{j=1}^M w_j x_j^{(i)} \right) \right)^2$$

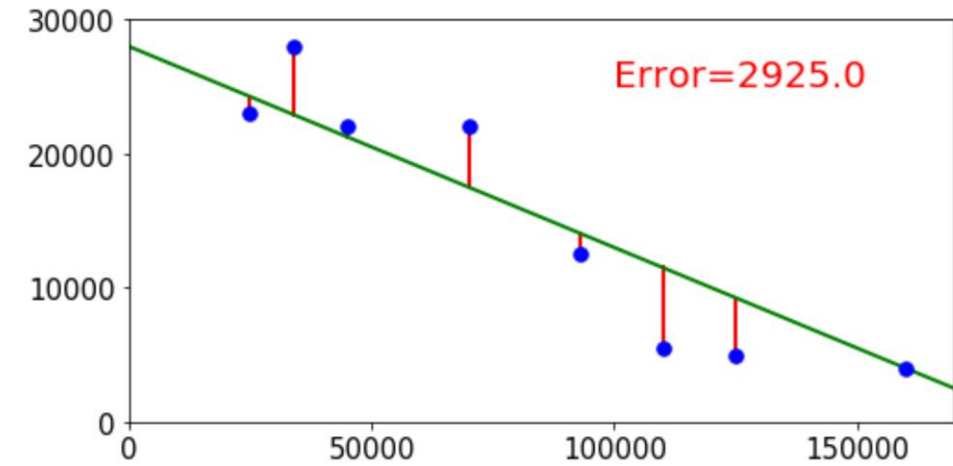
$$J(\vec{\theta}) = J(w, b) = \frac{1}{N} \sum_{i=1}^N \left(y^{(i)} - \vec{\theta}^T \vec{x}^{(i)} \right)^2$$

$$J(\vec{\theta}) = J(w, b) = \frac{1}{N} \| \vec{y} - X \vec{\theta} \|_2^2$$

vector of y data $\nearrow$ $\nwarrow$ Design matrix

FYI

$$\vec{\theta}^T \vec{x}^{(i)} = \vec{x}^{(i)T} \vec{\theta}$$



$$\theta = \begin{bmatrix} b \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

$$\vec{X} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$M=5$

Linear Algebra Formulation

Constructing the design matrix

Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
2	5.9	3.0	5.1	1.8

$\vec{y}$

$X_{orig} \in \mathbb{R}^{N \times M}$

Linear Algebra Formulation

Constructing the design matrix

Species
0
0
0
1
1
1
2

$\vec{y}$

Sepal Length	Sepal Width	Petal Length	Petal Width
4.3	3.0	1.1	0.1
4.9	3.6	1.4	0.1
5.3	3.7	1.5	0.2
4.9	2.4	3.3	1.0
5.7	2.8	4.1	1.3
6.3	3.3	4.7	1.6
5.9	3.0	5.1	1.8

$X_{orig} \in \mathbb{R}^{N \times M}$

$\vec{X}_{orig} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$
 $m=4$

$\vec{\theta} = \begin{bmatrix} b \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$
 $\vec{X} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$
 $M=5$

Linear Algebra Formulation

Constructing the design matrix

Species	(account for bias)	Sepal Length	Sepal Width	Petal Length	Petal Width
0	1	4.3	3.0	1.1	0.1
0	1	4.9	3.6	1.4	0.1
0	1	5.3	3.7	1.5	0.2
1	1	4.9	2.4	3.3	1.0
1	1	5.7	2.8	4.1	1.3
1	1	6.3	3.3	4.7	1.6
2	1	5.9	3.0	5.1	1.8

$\vec{y}$

Design matrix

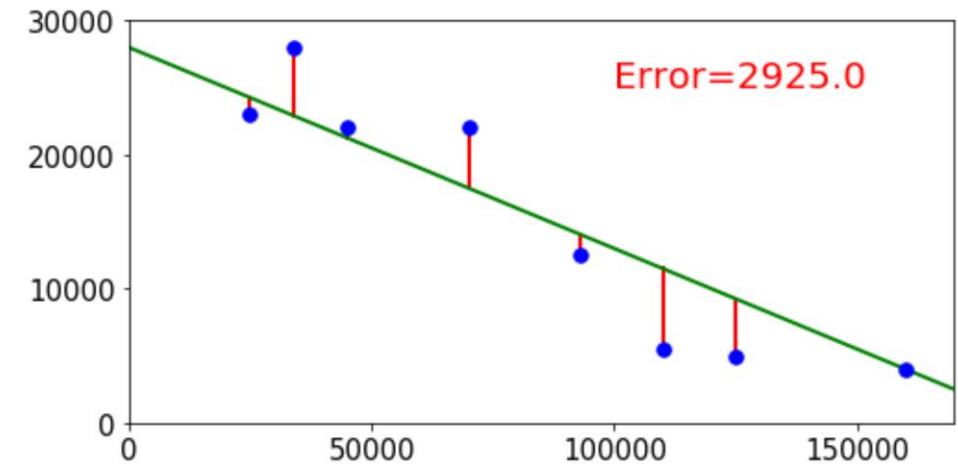
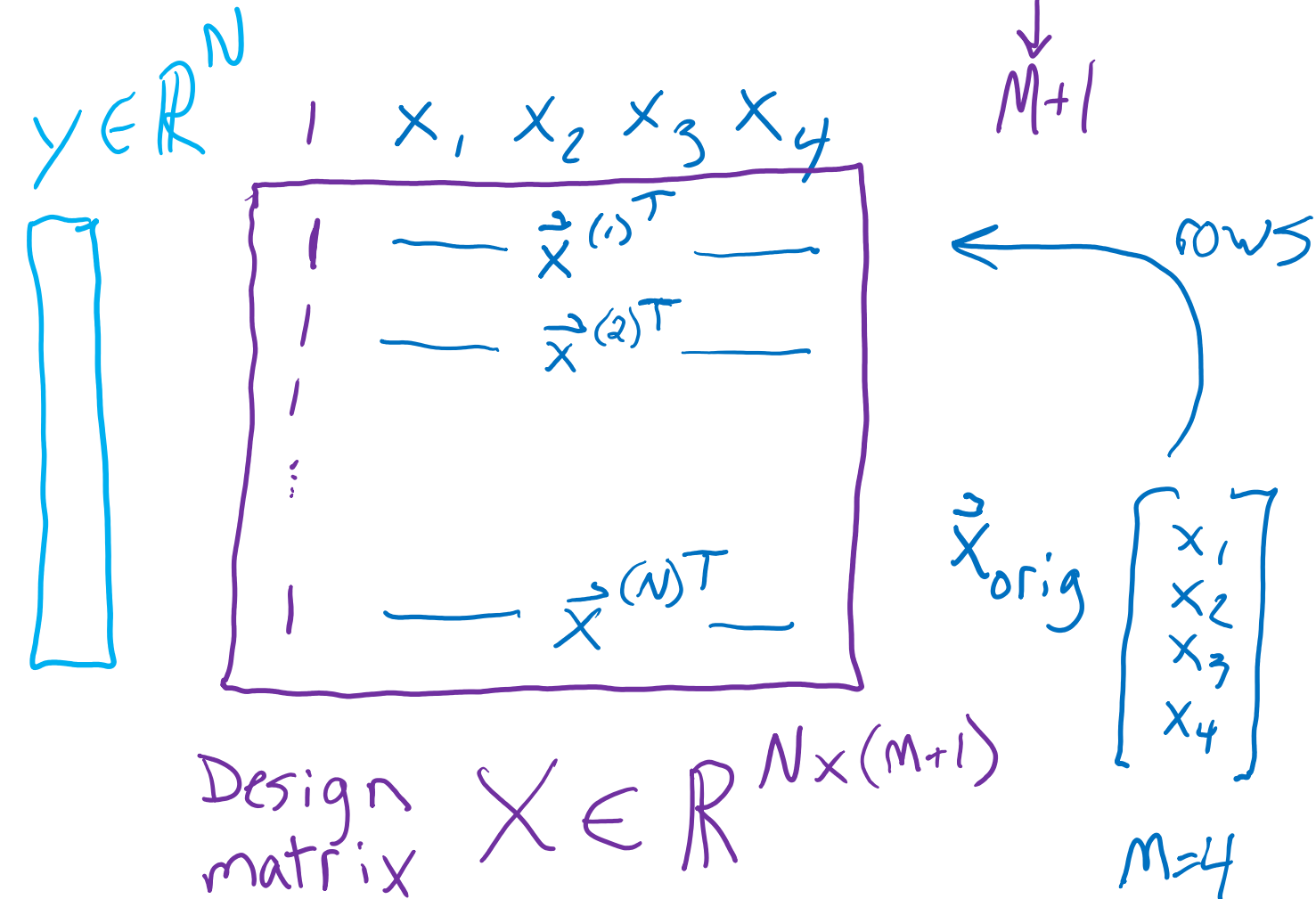
$$X \in \mathbb{R}^{N \times (m+1)}$$

$\vec{x}_{orig} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$
 $m=4$

$\vec{\theta} = \begin{bmatrix} b \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$
 $\vec{X} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$
 $M=5$

Linear Algebra Formulation

Constructing the design matrix M features

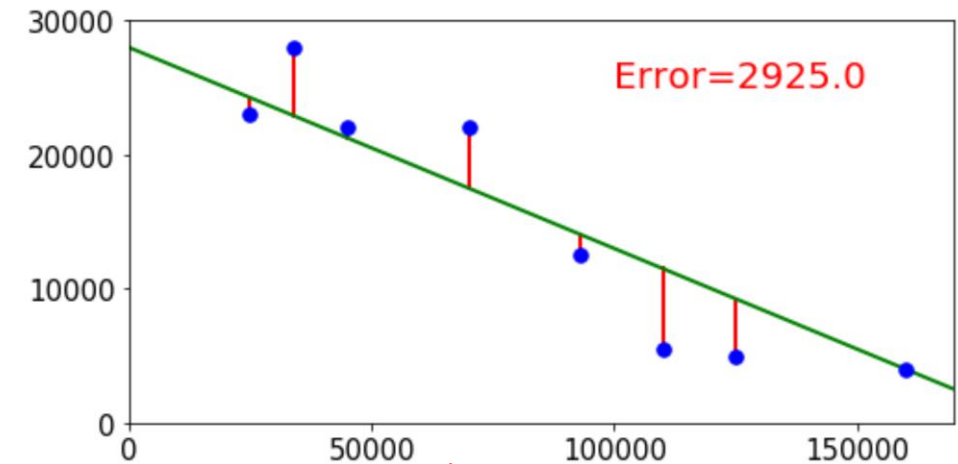


$$\Theta = \begin{bmatrix} b \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

$$\vec{X} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad M=5$$

Linear Algebra Formulation

Constructing the design matrix M features



$$y \in \mathbb{R}^N$$

	1	x_1	x_2	x_3	x_4
$\vec{x}^{(1)T}$					
$\vec{x}^{(2)T}$					
$\vdots$					
$\vec{x}^{(N)T}$					

X design matrix

$$\vec{x}_{\text{orig}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad M=4$$

$$\vec{\tilde{x}} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad M=5$$

$$J(\vec{\theta}) = J(w, b) = \frac{1}{N} \sum_{i=1}^N (y^{(i)} - \vec{\theta}^T \vec{x}^{(i)})^2$$

$$J(\vec{\theta}) = J(w, b) = \frac{1}{N} \|\vec{y} - X \vec{\theta}\|_2^2$$

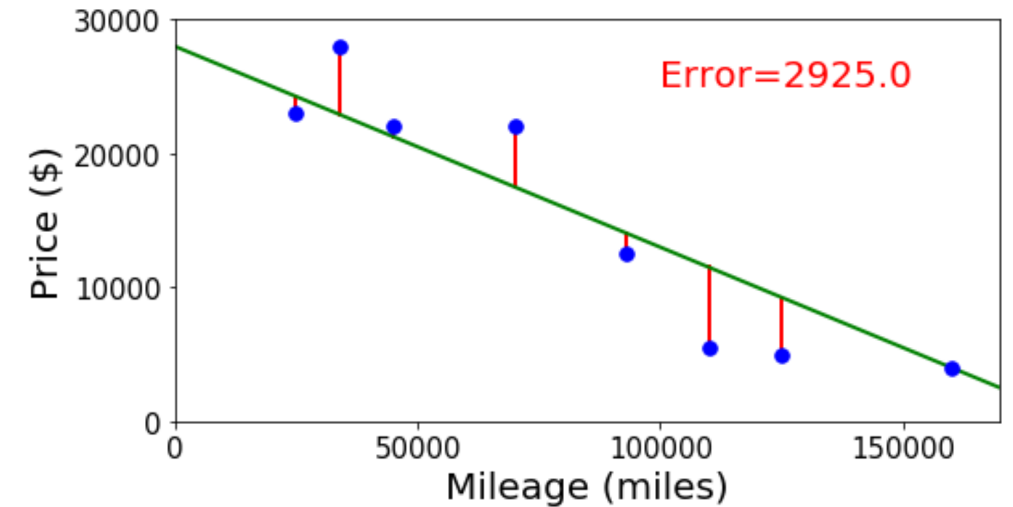
Linear Regression

System of Equations

Poll 4 (unused)

Now that we have $\mathbf{y} = X\boldsymbol{\theta}$, we can solve this system of equations to get the best $\boldsymbol{\theta}$?

- A. Yes
- B. No
- C. I have no idea



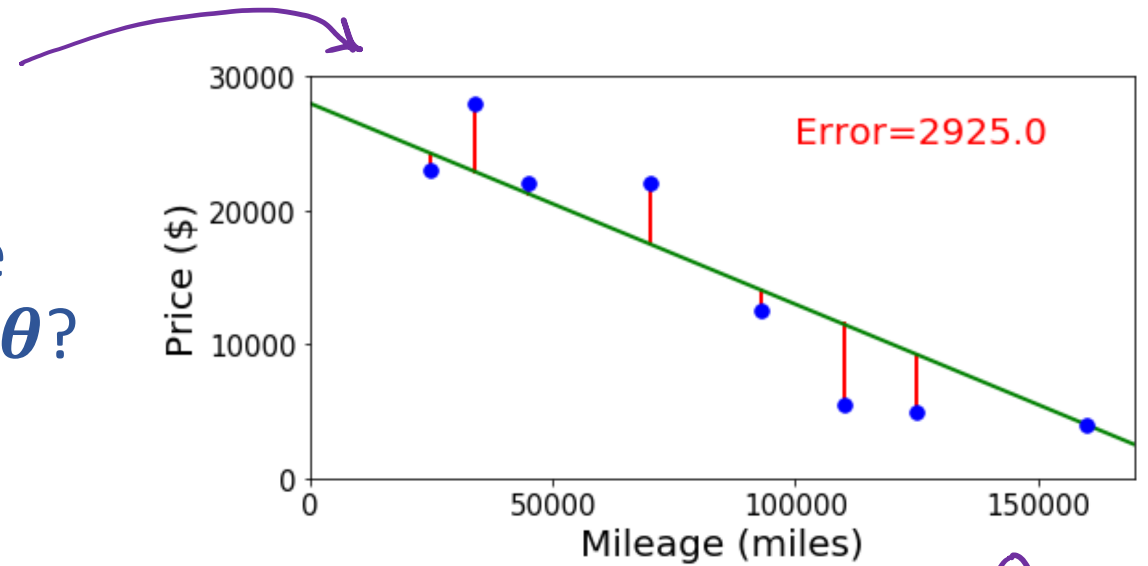
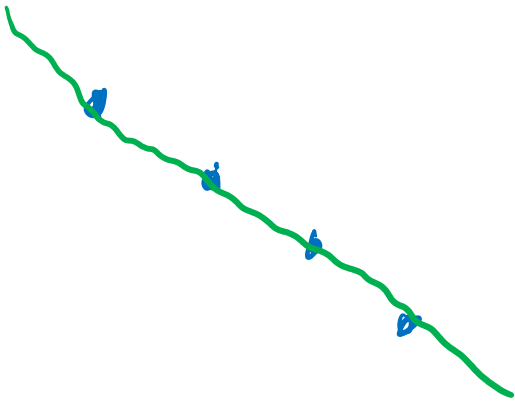
Poll 4 (unused)

Now that we have $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}$, we can solve this system of equations to get the best $\boldsymbol{\theta}$?

A. Yes

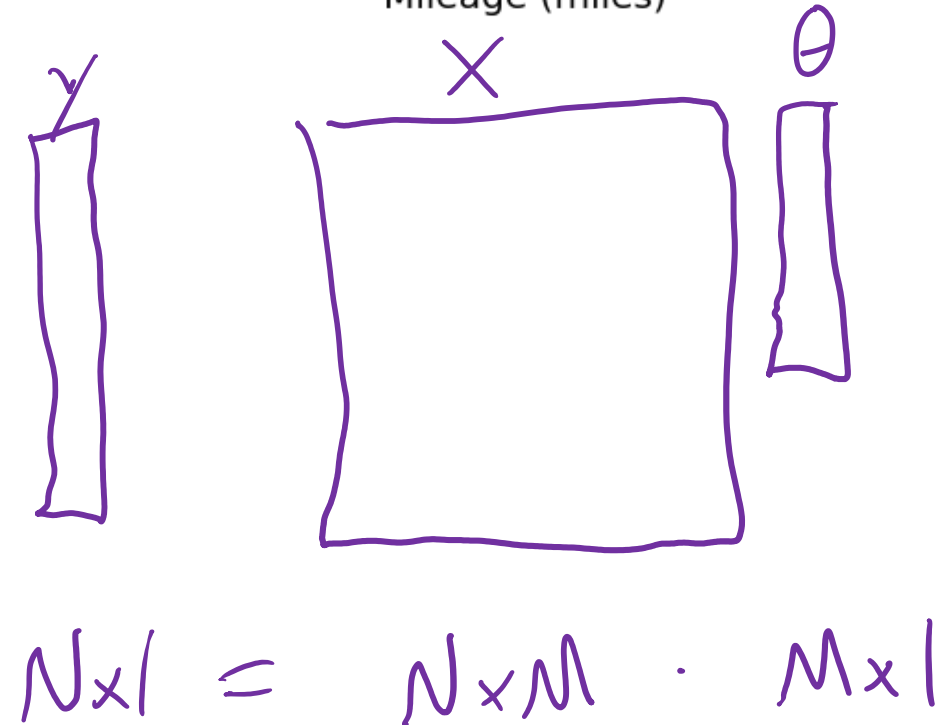
☒ B. No

C. I have no idea



solve?

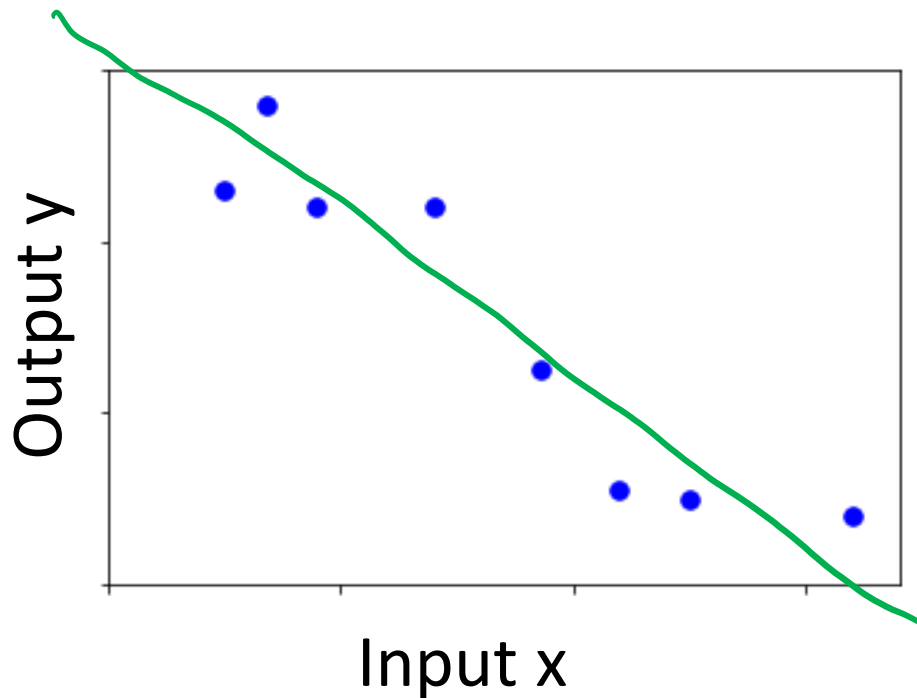
$$\mathbf{X}^{-1} \mathbf{y} = \hat{\boldsymbol{\theta}}$$


$$N \times 1 = N \times M \cdot M \times 1$$

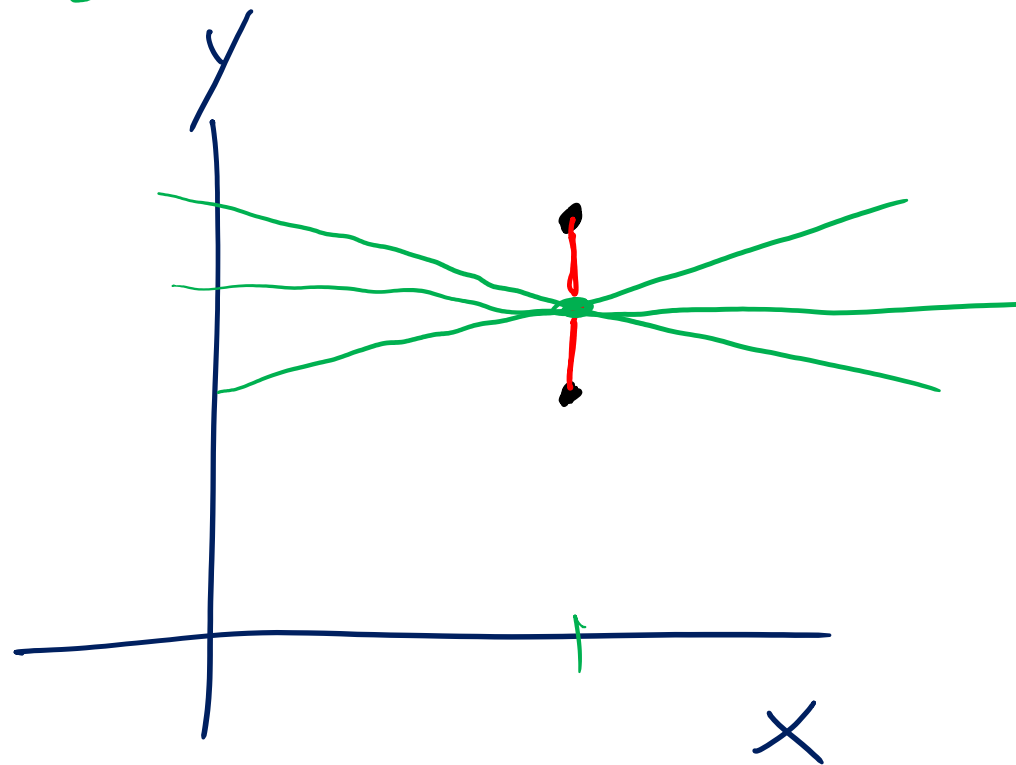
Linear Regression

Number of solutions?

$$J(\theta) = \frac{1}{N} \|\vec{y} - X\theta\|_2^2 \neq 0$$

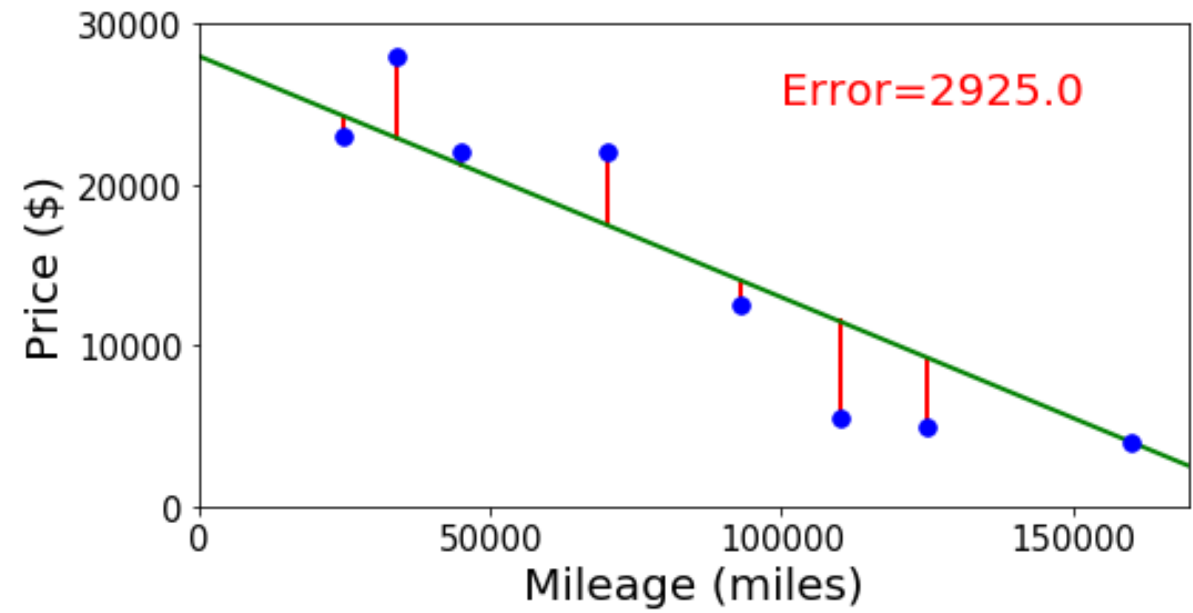


1 sol



Linear Regression

Linear algebra formulation



Linear Regression

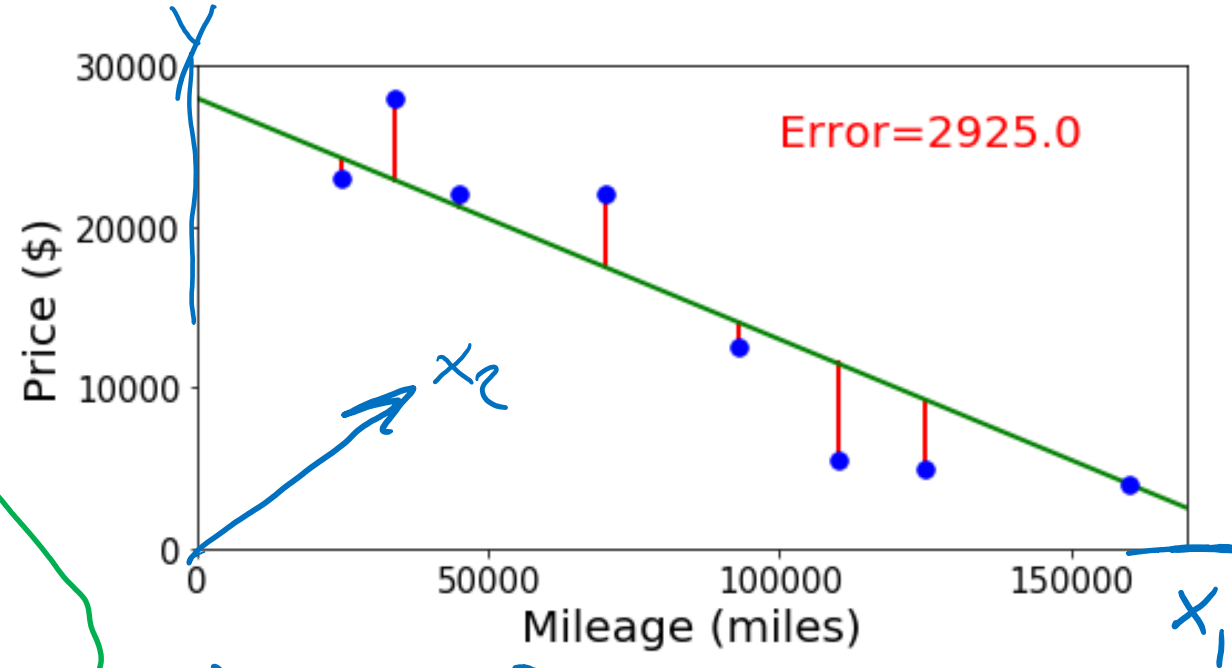
Linear algebra formulation

$$\vec{\theta} = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix} \quad \vec{X}^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \\ x_2^{(i)} \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ \vdots & \vdots & \vdots \\ 1 & x_1^{(n)} & x_2^{(n)} \end{bmatrix}$$

$$\vec{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

$$J(\vec{\theta}) = \frac{1}{N} \|\vec{y} - X\vec{\theta}\|_2^2$$



$$\begin{aligned} & \text{L}_2 \text{-norm squared} \\ & \|\vec{z}\|_2^2 = \vec{z}^T \vec{z} \\ & \sum_{i=1}^N (z_i)^2 \end{aligned}$$

Linear Regression

Expanding objective before computing gradient

$$\begin{aligned} J(\boldsymbol{\theta}) &= \frac{1}{N} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 \\ &= \frac{1}{N} (\mathbf{y} - X\boldsymbol{\theta})^T (\mathbf{y} - X\boldsymbol{\theta}) \\ &= \frac{1}{N} (\mathbf{y}^T - \boldsymbol{\theta}^T X^T) (\mathbf{y} - X\boldsymbol{\theta}) \\ &= \frac{1}{N} (\mathbf{y}^T \mathbf{y} - \boldsymbol{\theta}^T X^T \mathbf{y} - \mathbf{y}^T X \boldsymbol{\theta} + \boldsymbol{\theta}^T X^T X \boldsymbol{\theta}) \\ &= \frac{1}{N} (\mathbf{y}^T \mathbf{y} - 2\boldsymbol{\theta}^T X^T \mathbf{y} + \boldsymbol{\theta}^T X^T X \boldsymbol{\theta}) \end{aligned}$$

Linear Regression

$$\underline{\|\vec{z}\|_2^2 = \sum_{i=1}^N z_i^2 = \sum_{i=1}^N z_i z_i = \frac{\vec{z}^T \vec{z}}{1}}$$

Expanding objective before computing gradient

$$J(\boldsymbol{\theta}) = \frac{1}{N} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$$

$$= \frac{1}{N} (\mathbf{y} - X\boldsymbol{\theta})^T (\mathbf{y} - X\boldsymbol{\theta})$$

$$= \frac{1}{N} (\mathbf{y}^T - \boldsymbol{\theta}^T X^T) (\mathbf{y} - X\boldsymbol{\theta})$$

$$= \frac{1}{N} (\mathbf{y}^T \mathbf{y} - \boldsymbol{\theta}^T X^T \mathbf{y} - \mathbf{y}^T X \boldsymbol{\theta} + \boldsymbol{\theta}^T X^T X \boldsymbol{\theta})$$

$$= \frac{1}{N} (\mathbf{y}^T \mathbf{y} - 2\boldsymbol{\theta}^T X^T \mathbf{y} + \boldsymbol{\theta}^T X^T X \boldsymbol{\theta})$$

distrib T

distr. mul. + add.

$$\boldsymbol{\theta}^T X^T \mathbf{y} = \mathbf{y}^T X \boldsymbol{\theta}$$

both scalars

$$[1 \times M][M \times N][N \times 1] = [1 \times N][N \times M][M \times 1]$$

Linear Regression

Optimizing the objective

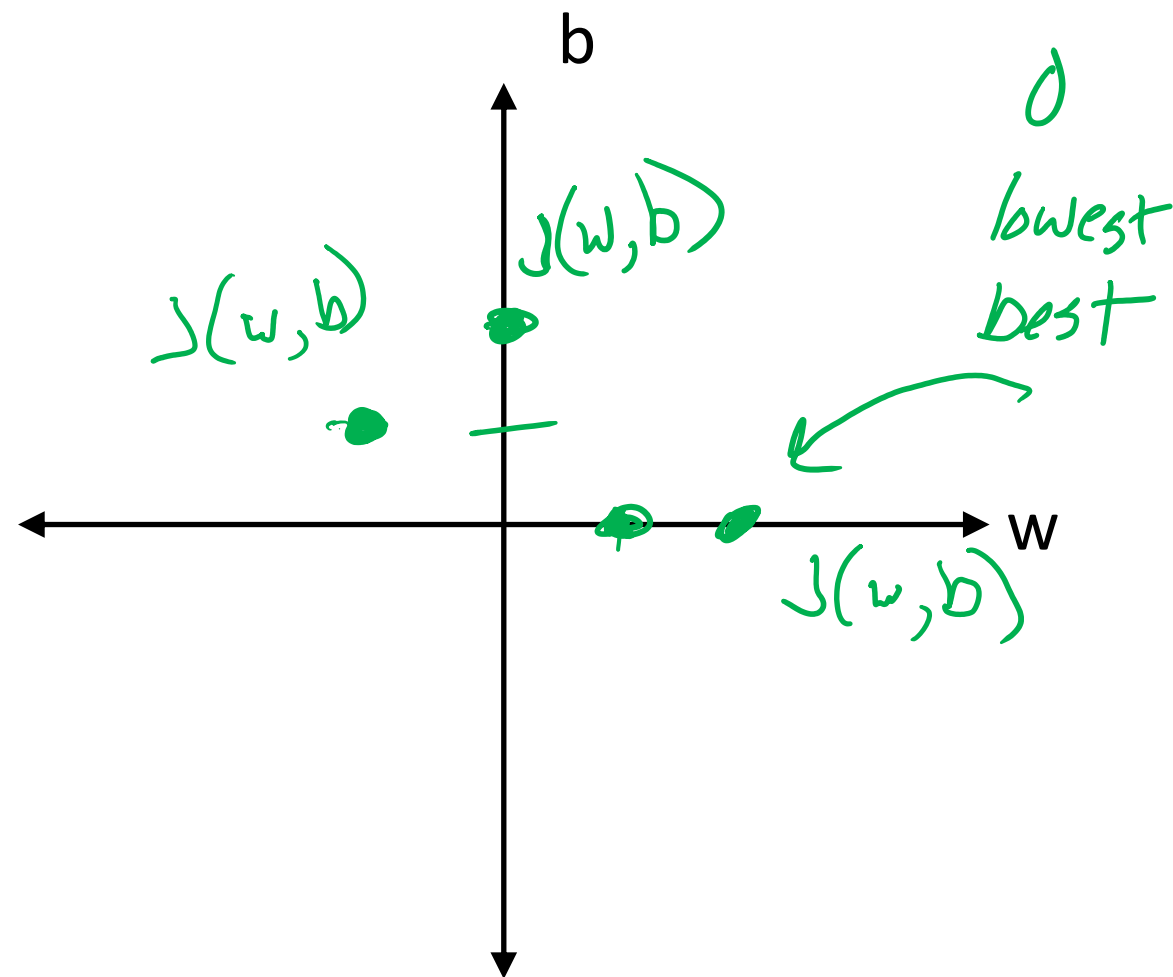
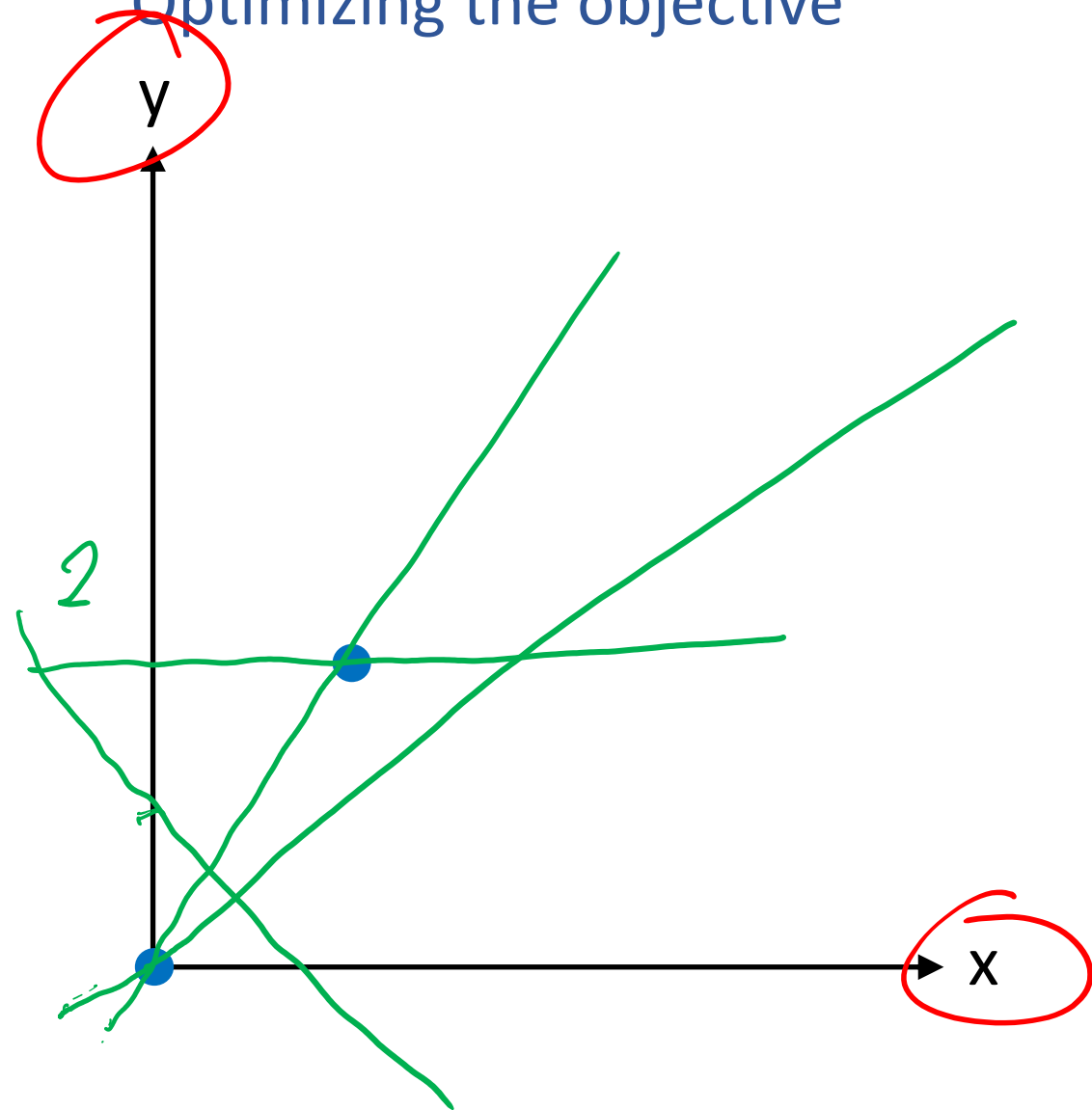
From course website under lectures:

- Demo: `interactive_regression.ipynb`
- Exercise: `interactive_blind_regression.ipynb`

Linear Regression

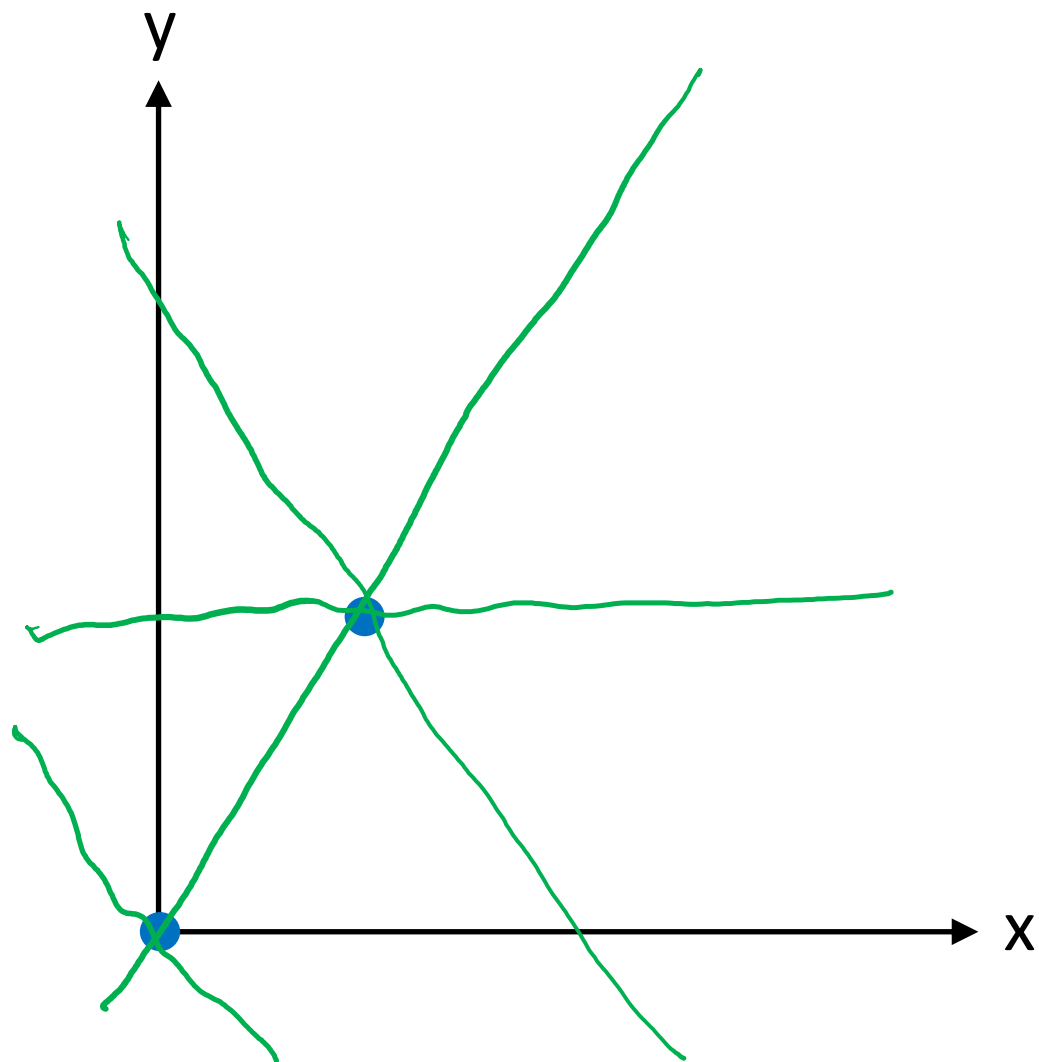
$$\underline{J(w, b)} = \frac{1}{2} \left[\left(y^{(1)} - (wx^{(1)} + b) \right)^2 + \left(y^{(2)} - (wx^{(2)} + b) \right)^2 \right]$$

Optimizing the objective

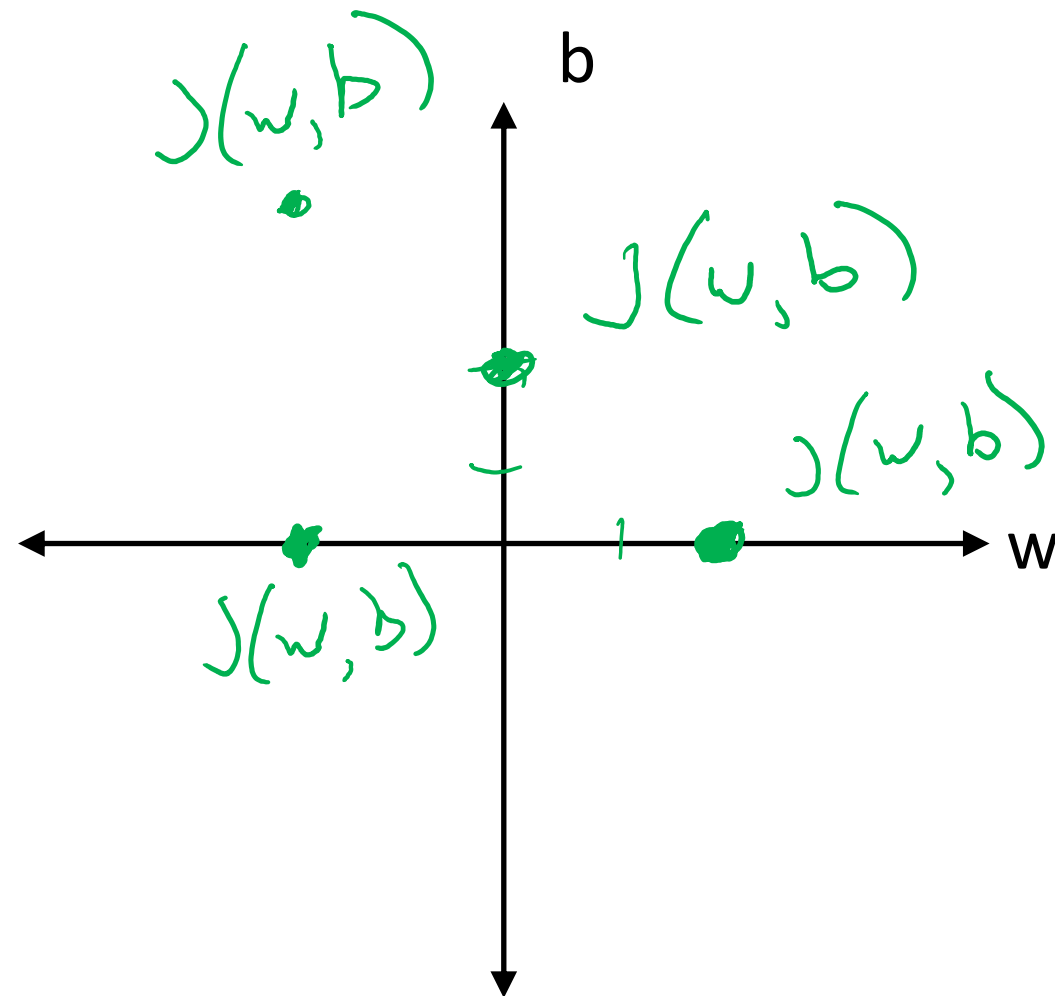


Linear Regression

Optimizing the objective



$$J(w, b) = \frac{1}{2} \left[\left(y^{(1)} - (wx^{(1)} + b) \right)^2 + \left(y^{(2)} - (wx^{(2)} + b) \right)^2 \right]$$

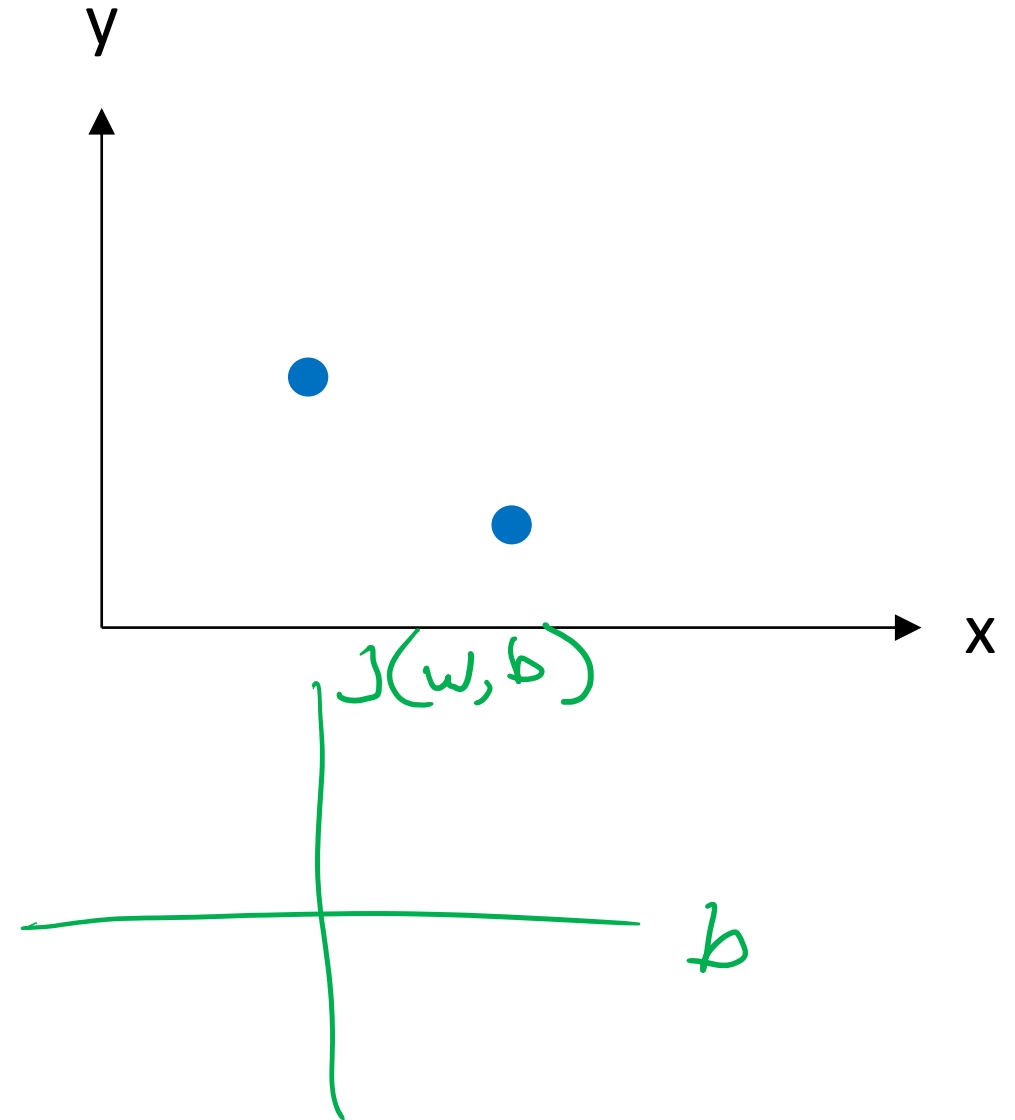


Poll 5

$$J(w, b) = \frac{1}{2} \left[\left(\cancel{y^{(1)}} - (\cancel{wx^{(1)}} + \underline{b}) \right)^2 + \left(\cancel{y^{(2)}} - (\cancel{wx^{(2)}} + \underline{b}) \right)^2 \right]$$

For fixed data and fixed slope, w , what shape do we get by plotting MSE objective vs intercept, b ?

- A. Line
- B. Plane
- C. Half-plane
- ☒ D. Convex parabola (U-shape)
- E. Concave parabola (up-side-down U)
- F. None of the above

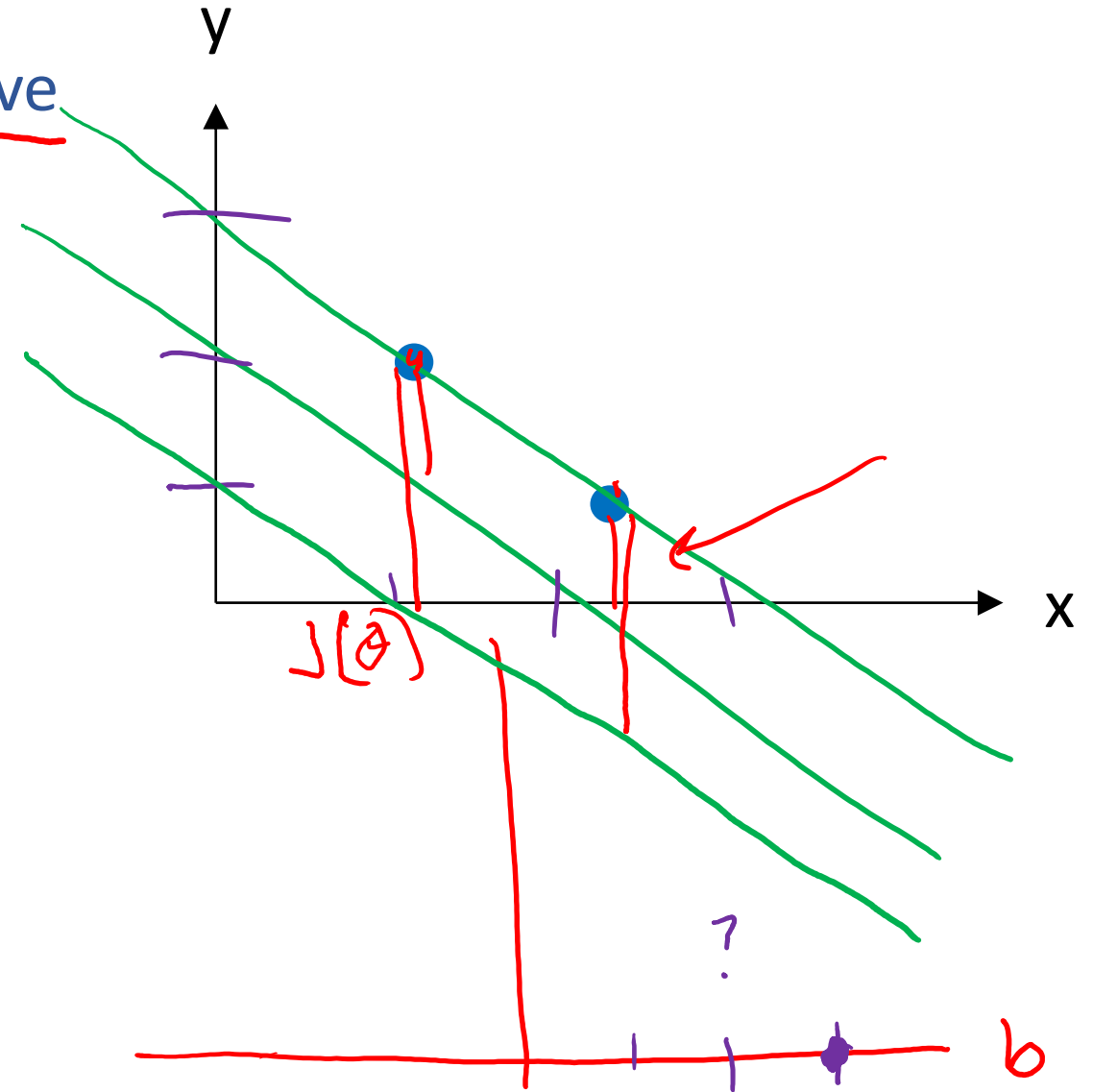


Poll 5

$$J(w, b) = \frac{1}{2} \left[\left(\underline{y^{(1)}} - (\underline{wx^{(1)}} + b) \right)^2 + \left(\underline{y^{(2)}} - (\underline{wx^{(2)}} + b) \right)^2 \right]$$

For fixed data and fixed slope, w , what shape do we get by plotting MSE objective vs intercept, b ?

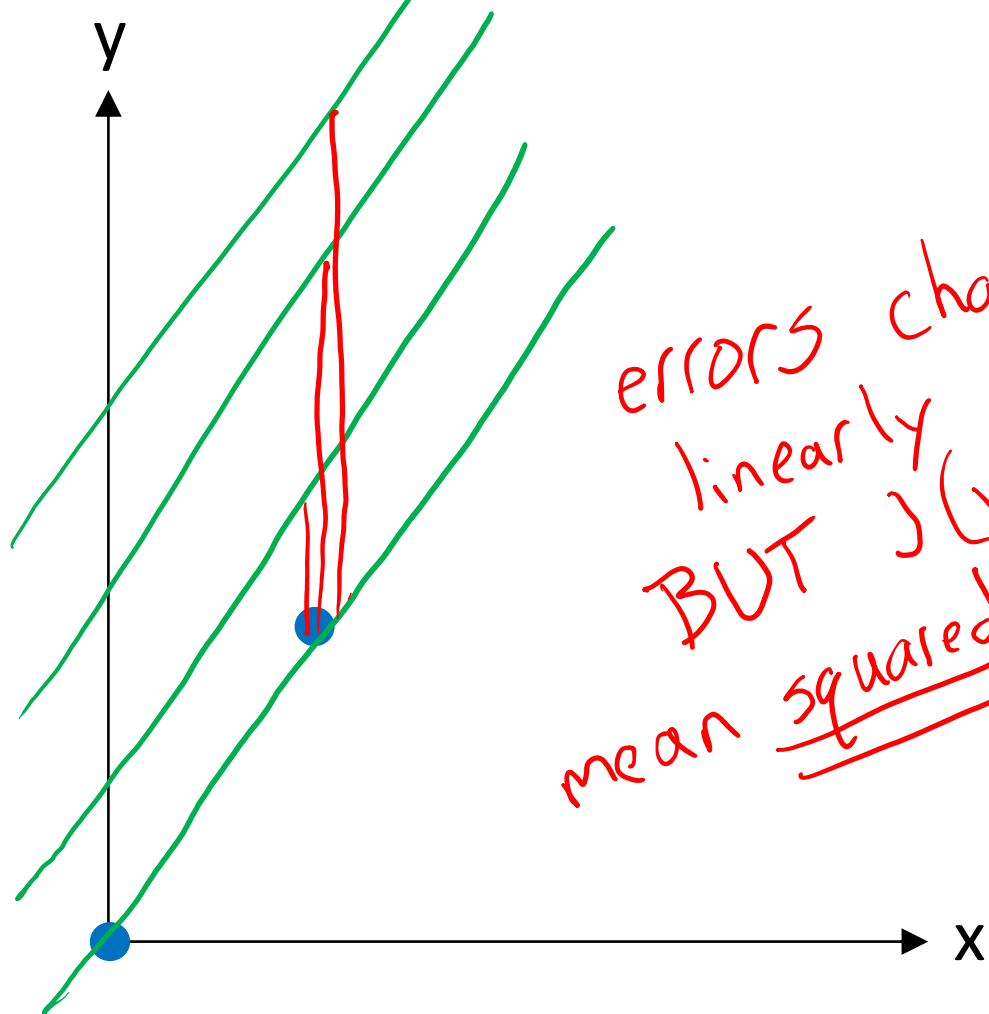
- A. Line
- B. Plane
- C. Half-plane
- ☒ D. Convex parabola (U-shape)
- E. Concave parabola (up-side-down U)
- F. None of the above



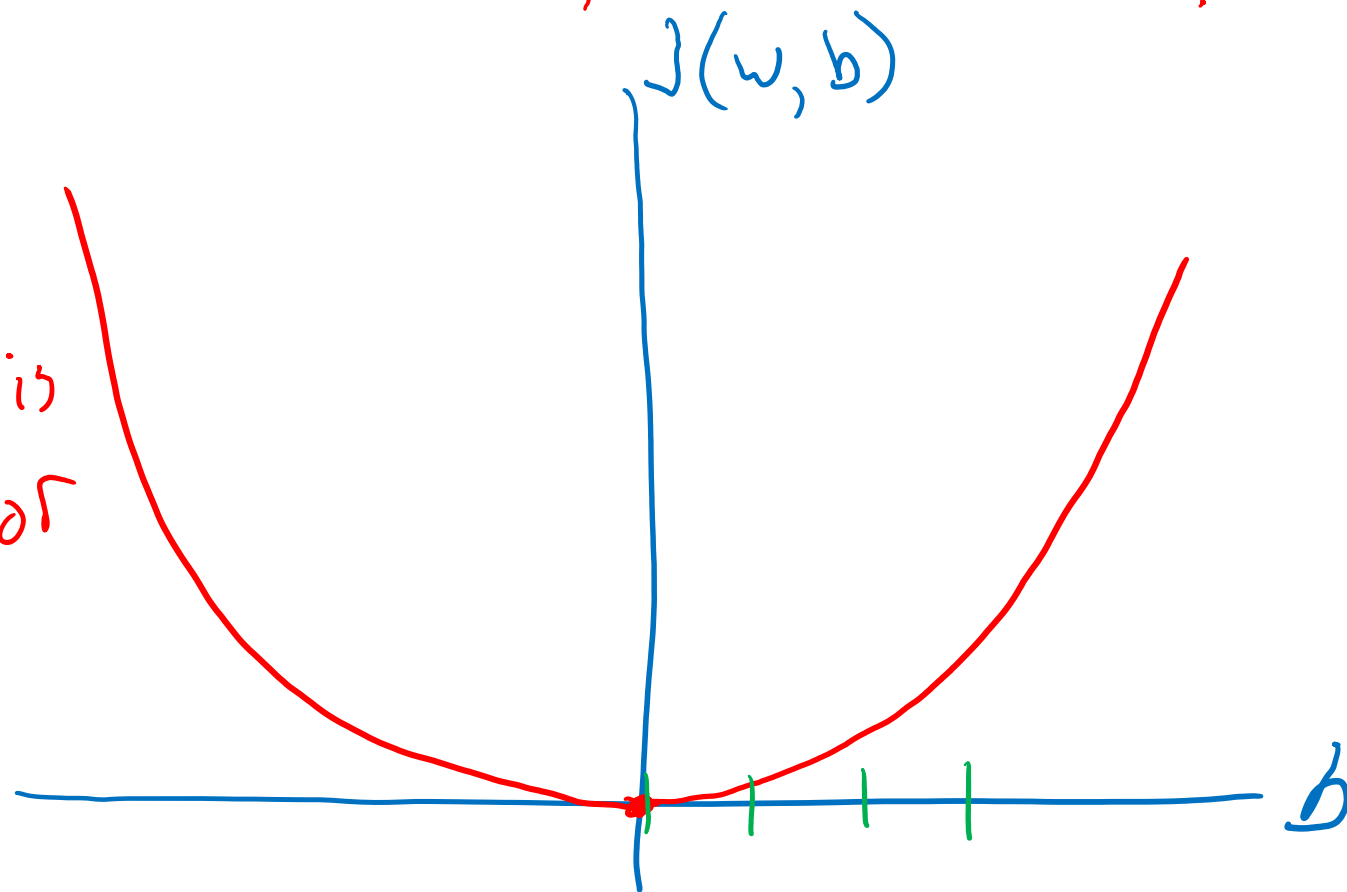
Linear Regression

Optimizing the objective

$$J(w, b) = \frac{1}{2} \left[\left(y^{(1)} - (wx^{(1)} + b) \right)^2 + \left(y^{(2)} - (wx^{(2)} + b) \right)^2 \right]$$



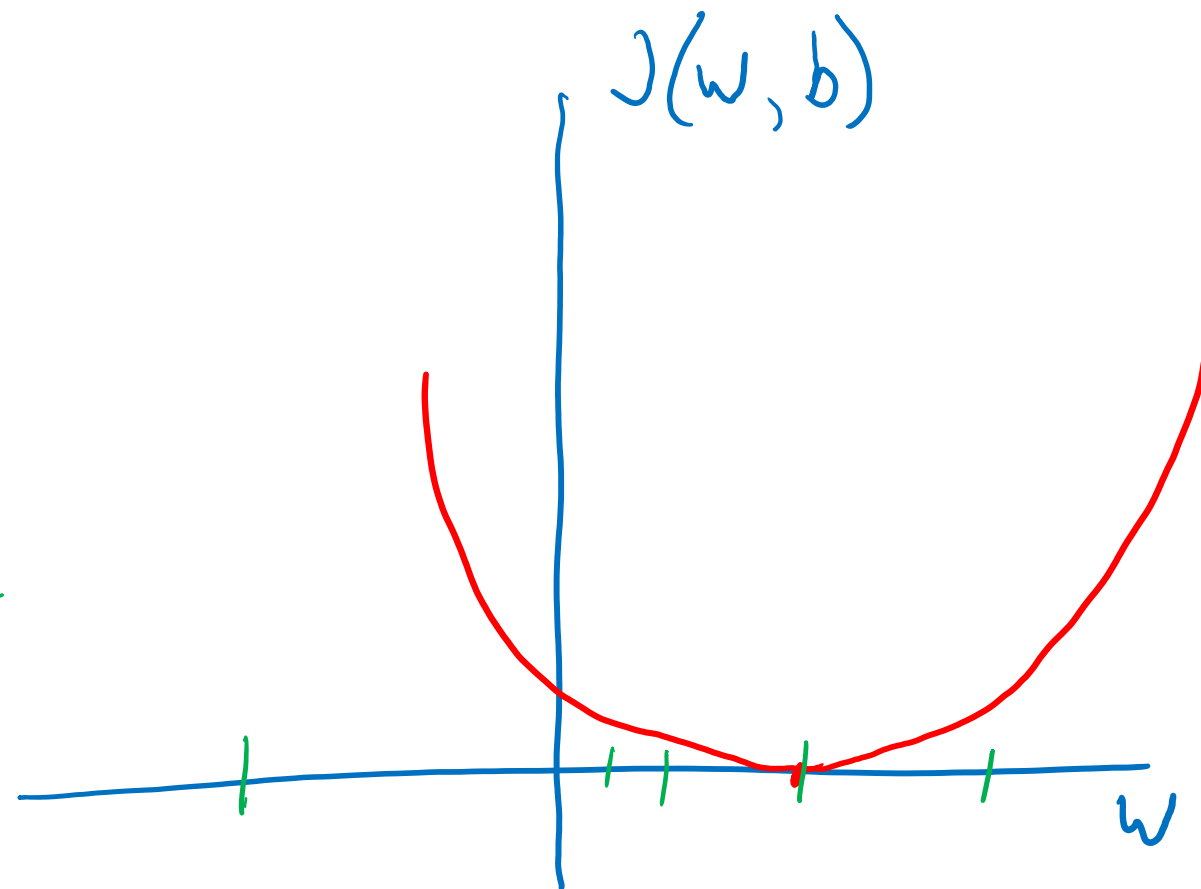
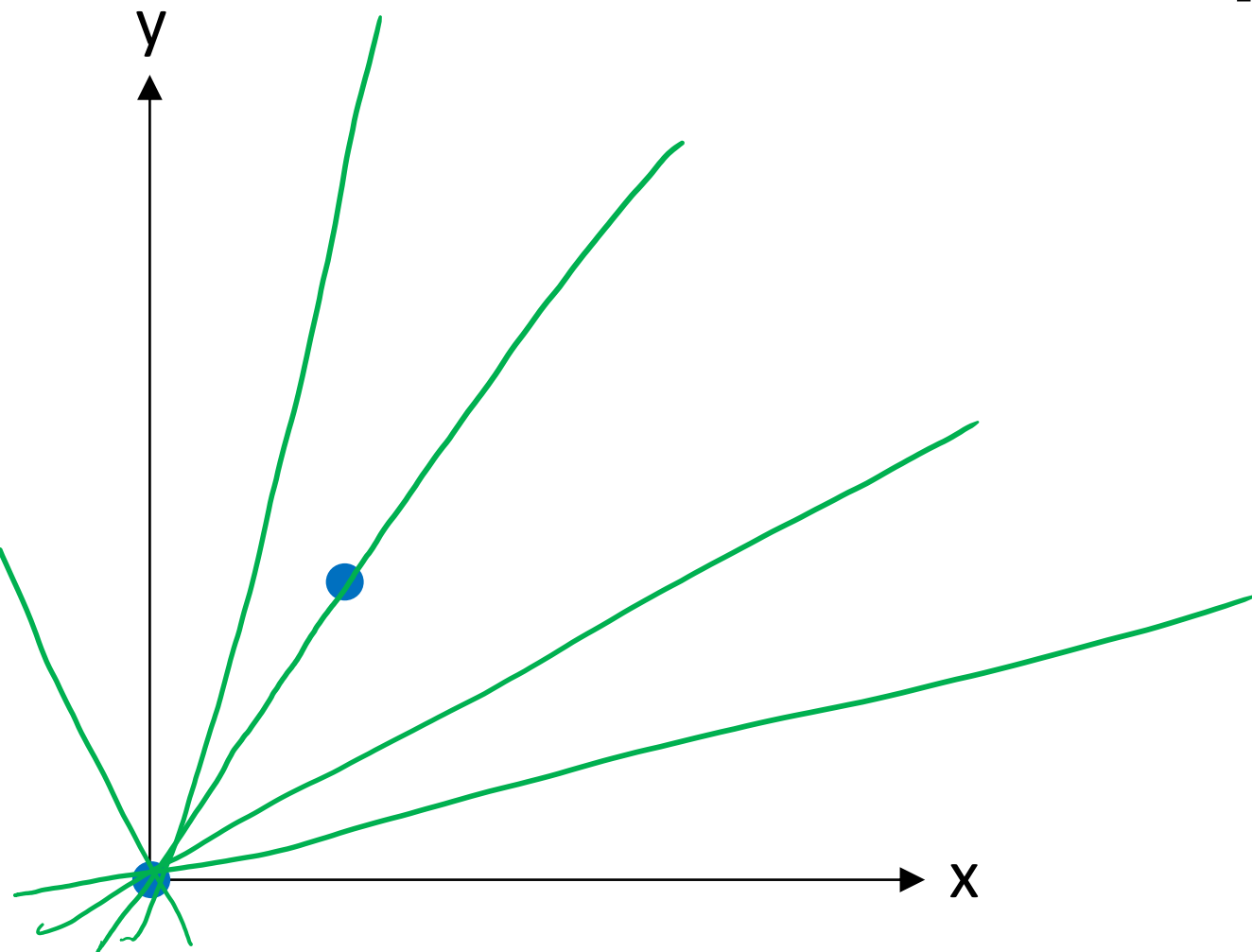
errors change
linearly
BUT $J(w, b)$ is
mean squared error



Linear Regression

Optimizing the objective

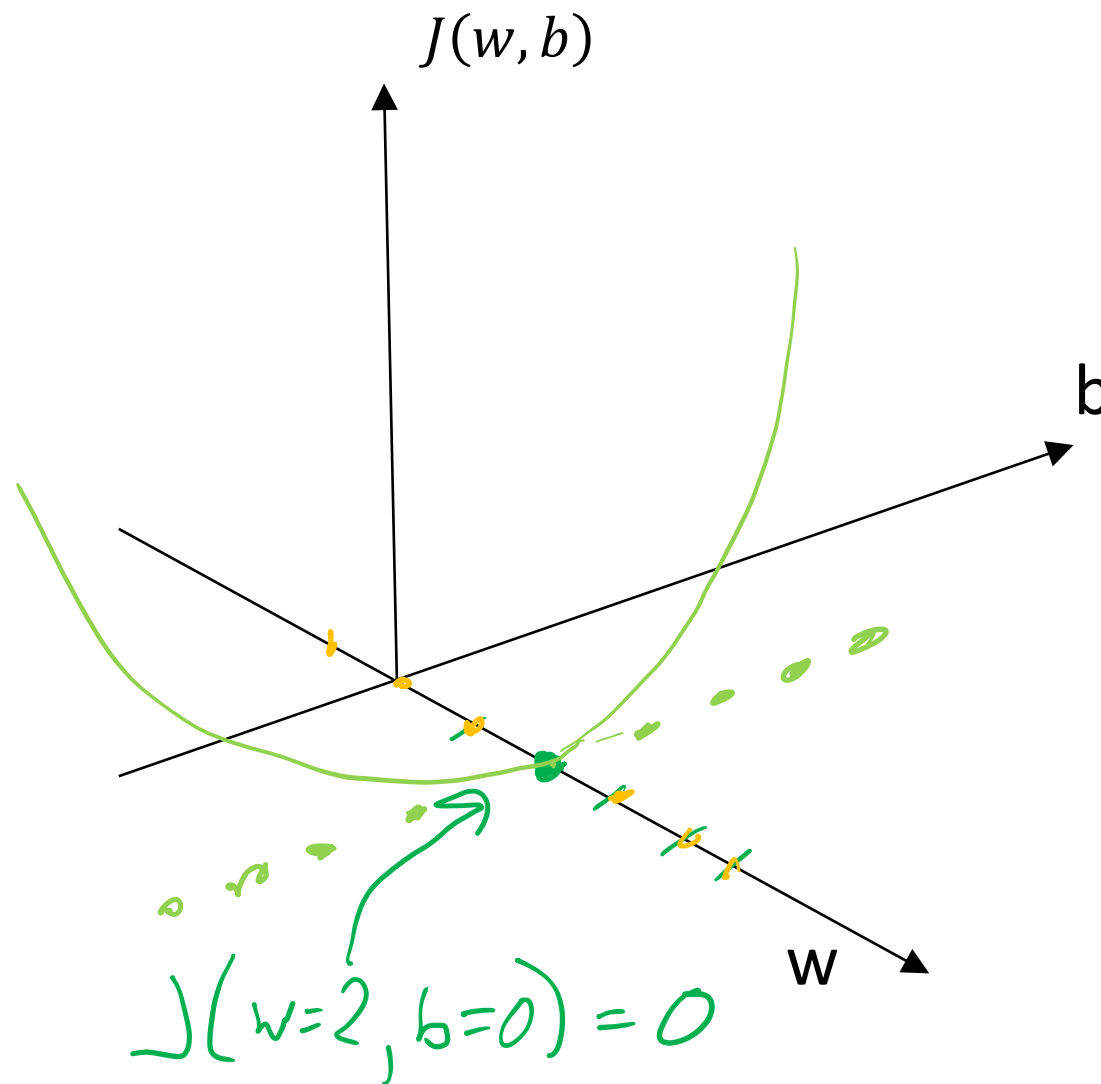
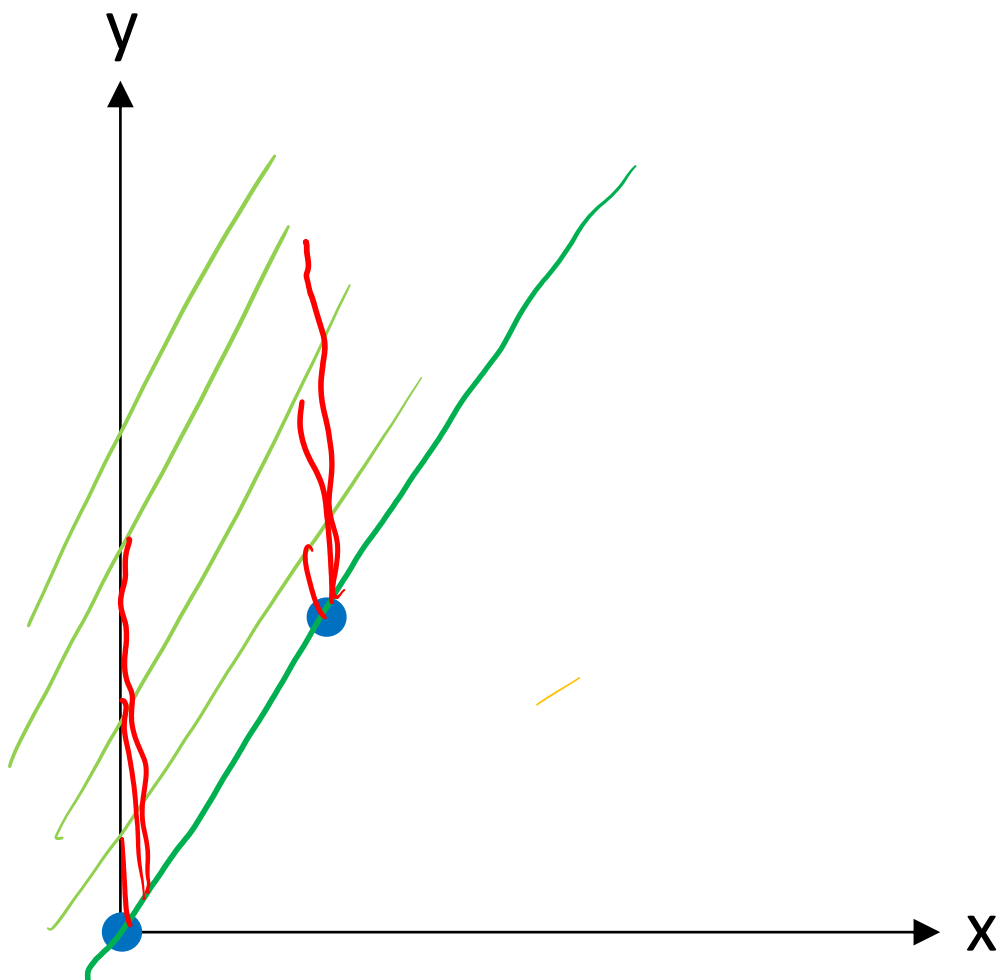
$$J(w, b) = \frac{1}{2} \left[\left(\underline{y^{(1)}} - (\underline{w} \underline{x^{(1)}} + \cancel{b}) \right)^2 + \left(y^{(2)} - (\underline{w} \underline{x^{(2)}} + \cancel{b}) \right)^2 \right]$$



Linear Regression

$$J(w, b) = \frac{1}{2} \left[\left(y^{(1)} - (wx^{(1)} + b) \right)^2 + \left(y^{(2)} - (wx^{(2)} + b) \right)^2 \right]$$

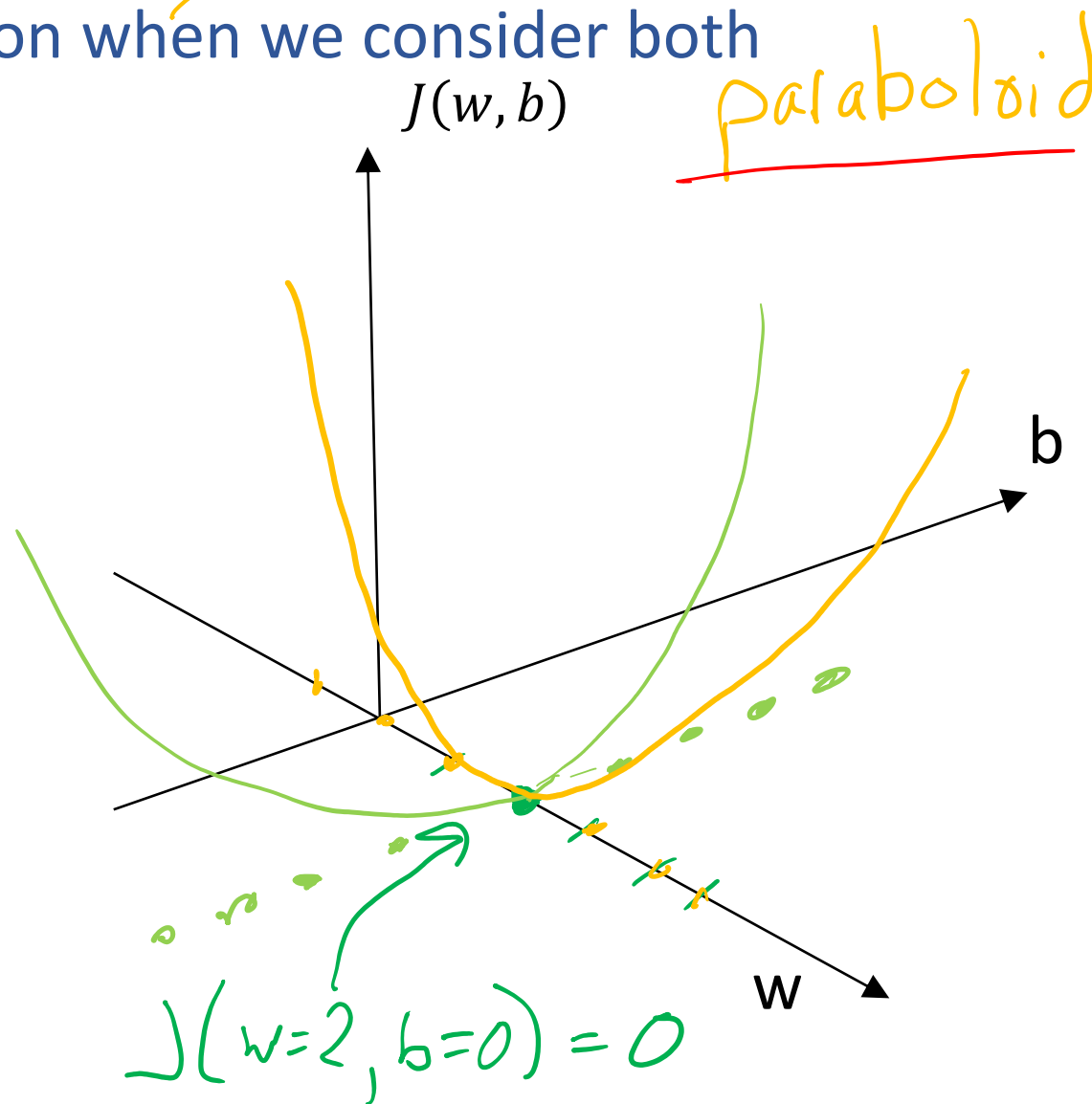
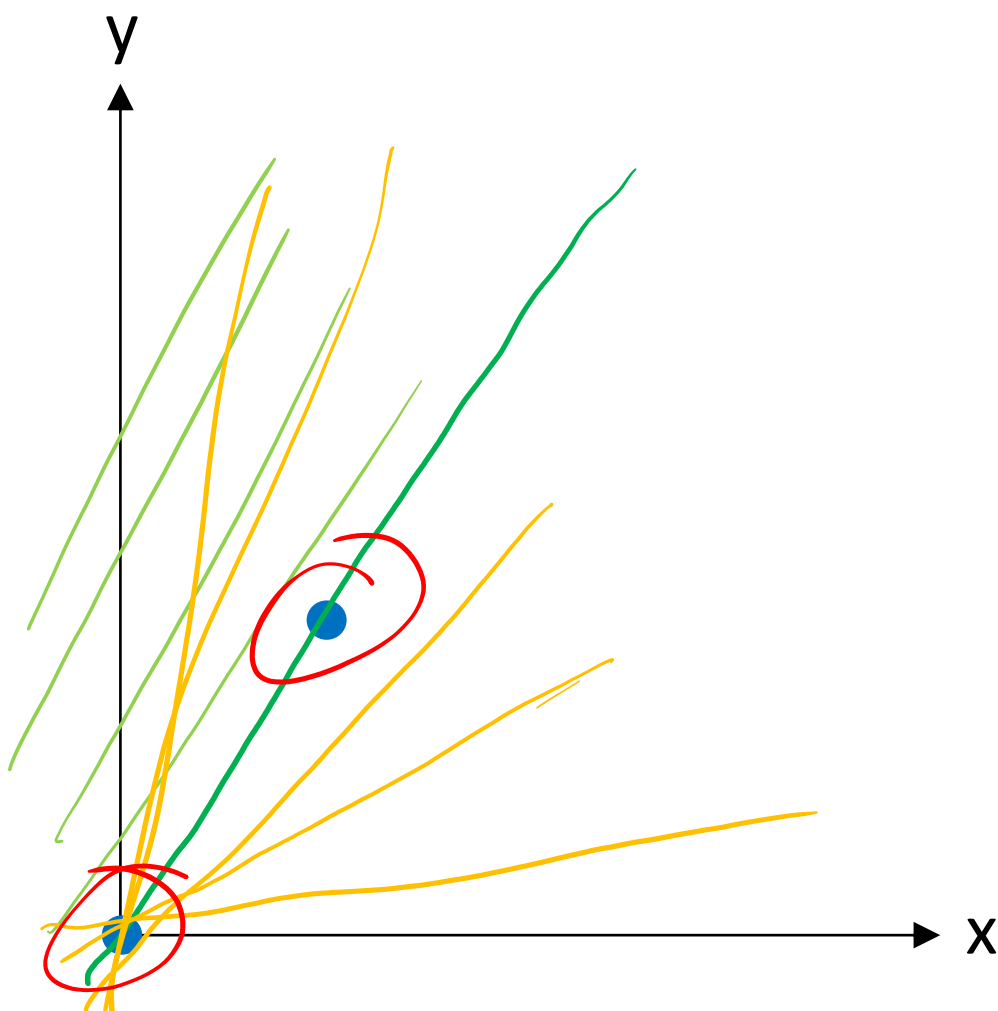
What shape is the objective function when we consider both parameters w and b ?



Linear Regression

$$J(w, b) = \frac{1}{2} \left[\left(\underline{y^{(1)}} - (\underline{wx^{(1)}} + b) \right)^2 + \left(\underline{y^{(2)}} - (\underline{wx^{(2)}} + b) \right)^2 \right]$$

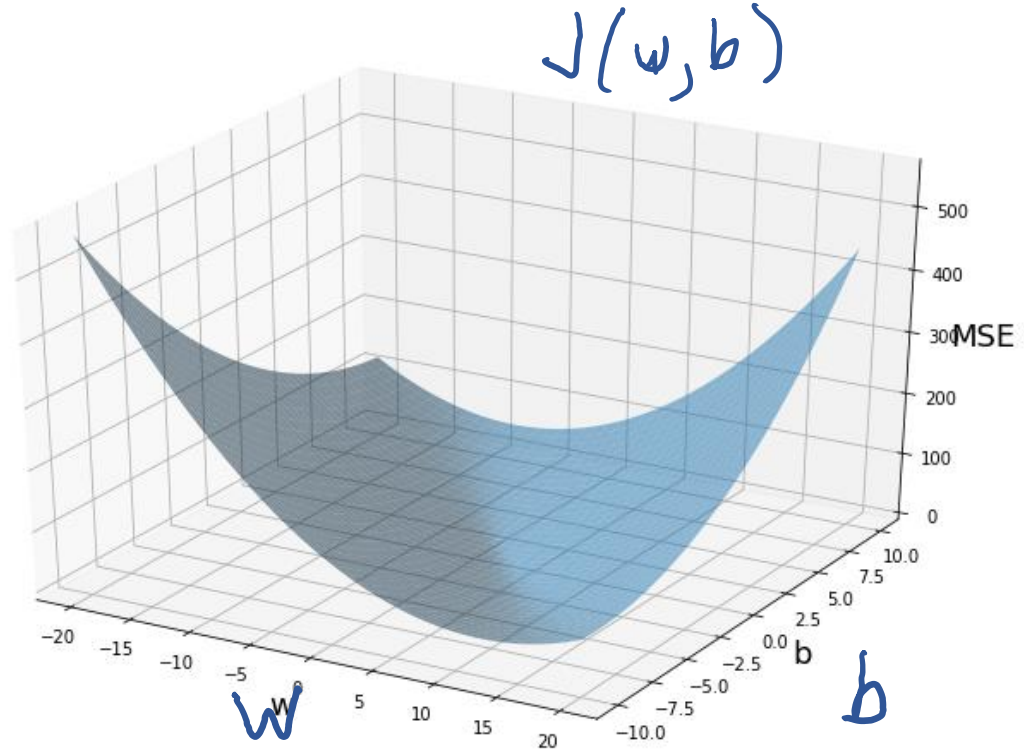
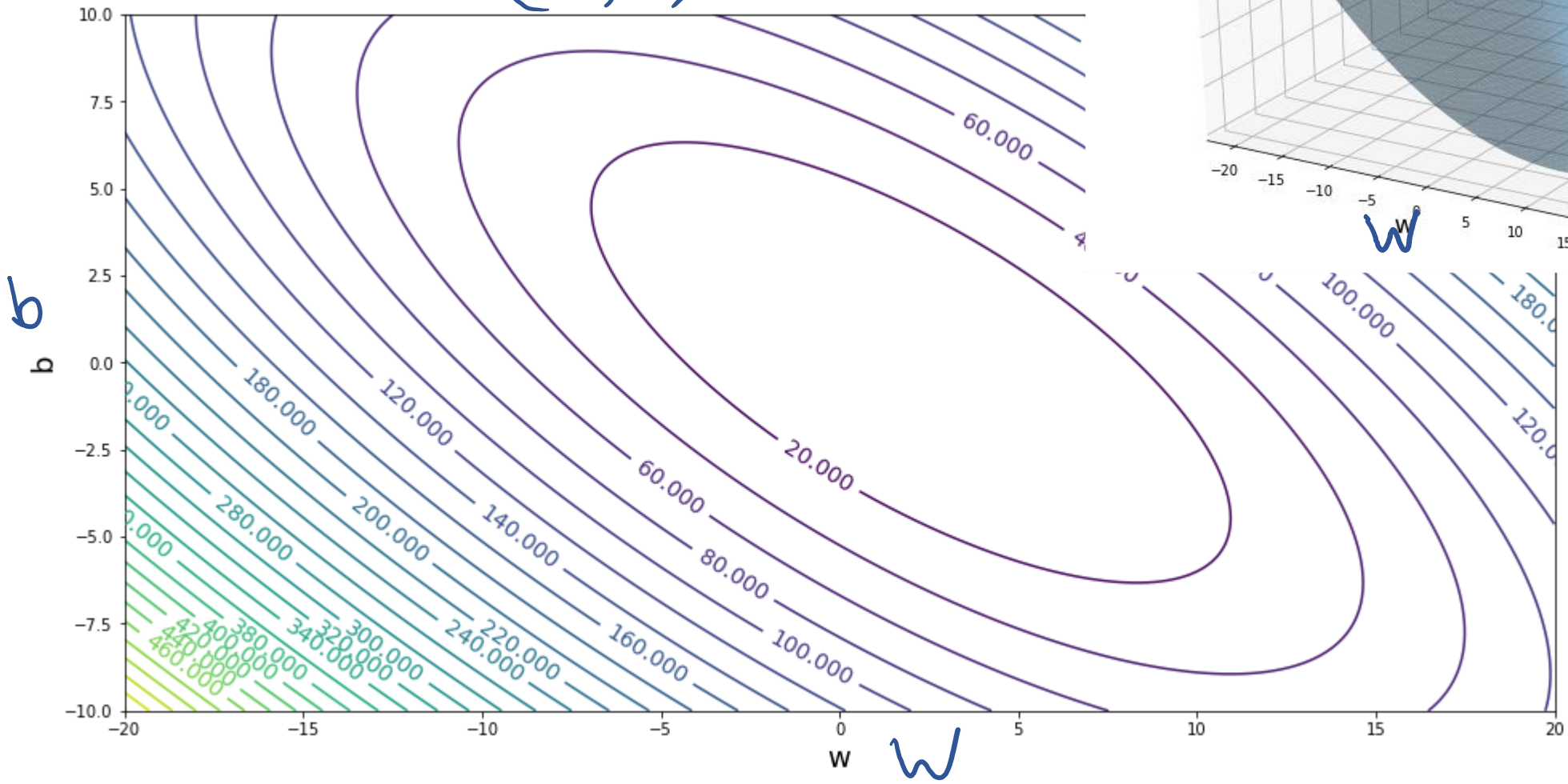
What shape is the objective function when we consider both parameters w and b ?



Linear Regression

Optimizing the objective

$J(w, b)$

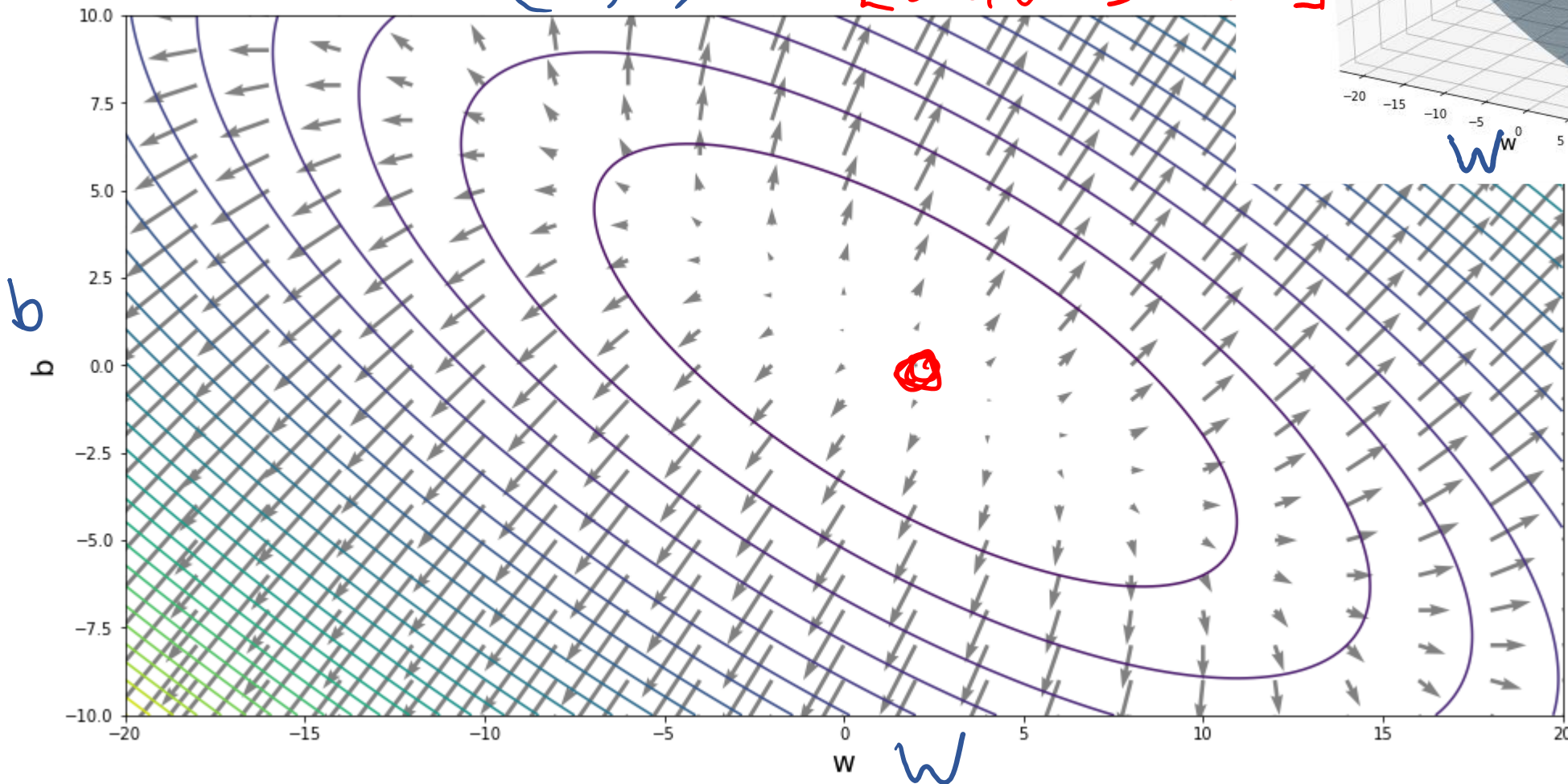
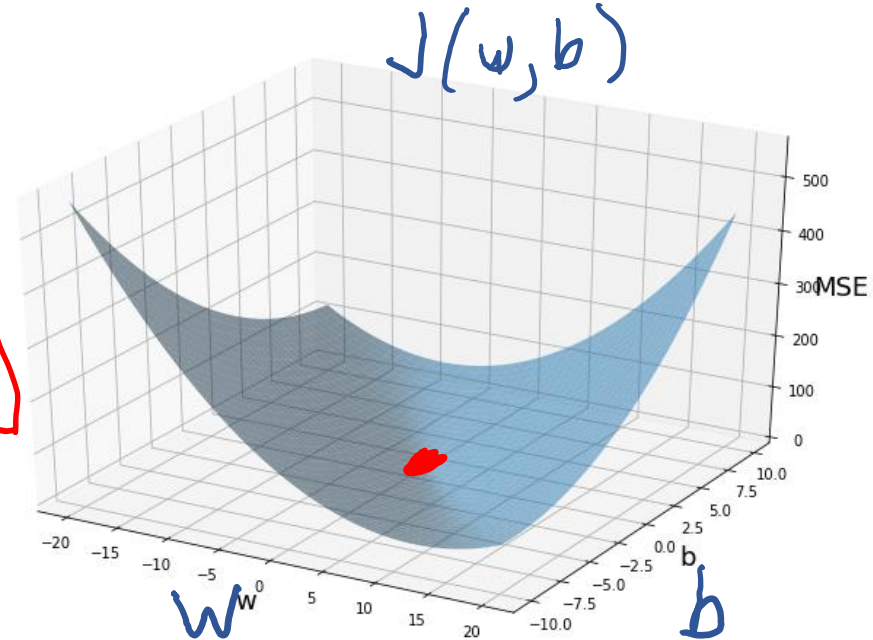


Optimization

Gradients

$$\nabla J(w, b) = \vec{0}$$
$$\begin{bmatrix} dJ/dw \\ dJ/db \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$J(w, b)$



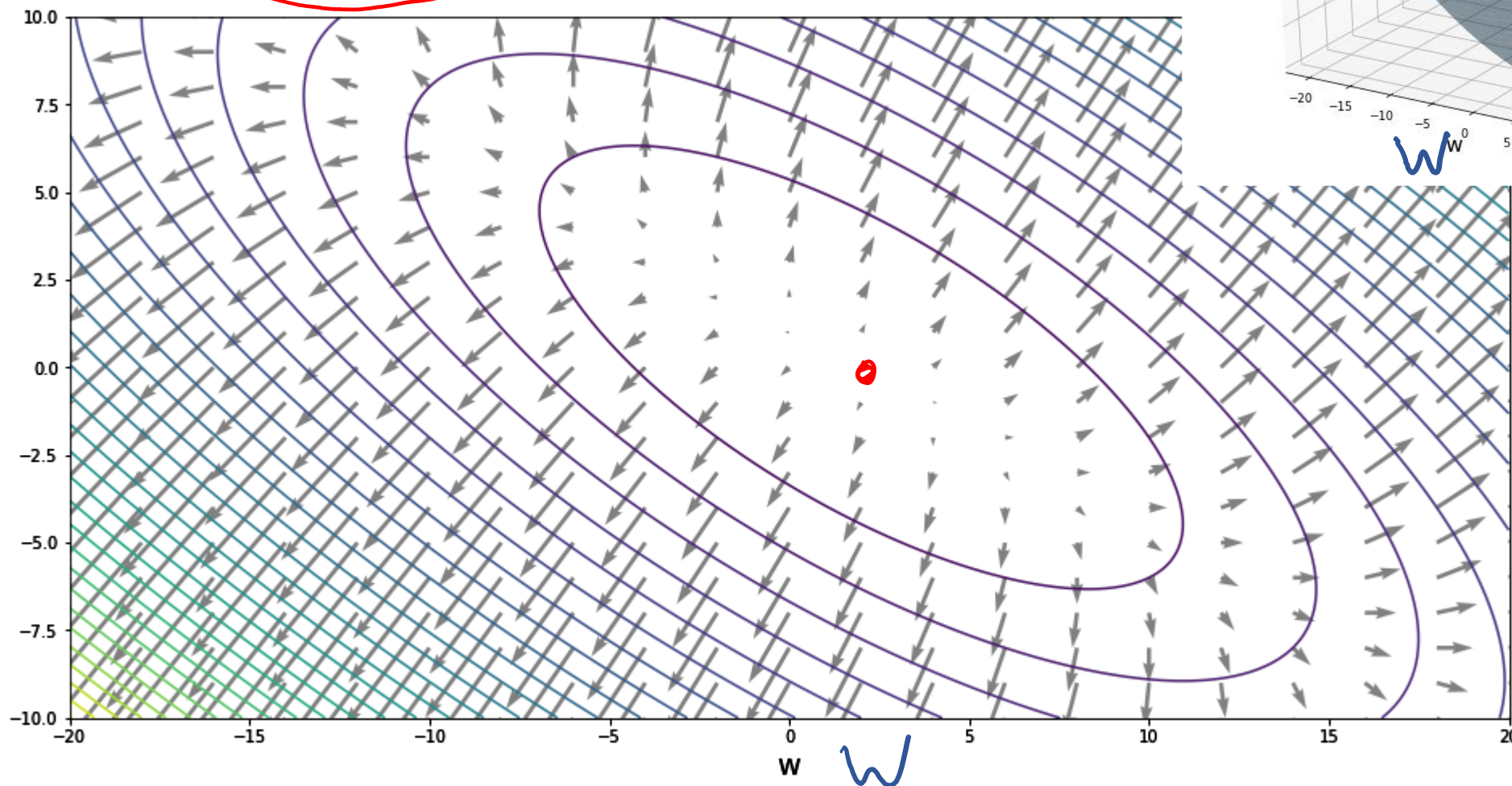
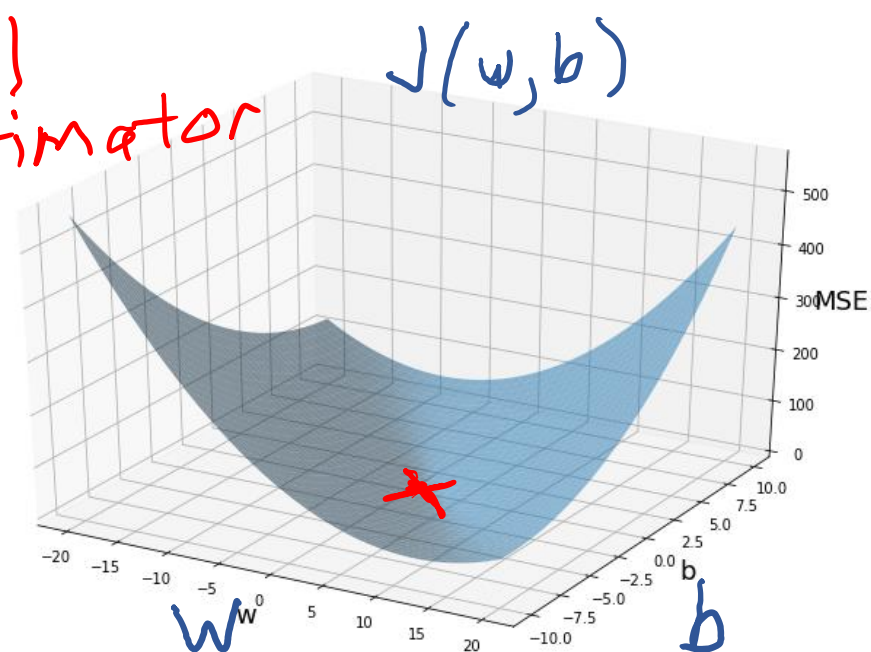
$$\nabla J(w, b)$$

Optimization

Linear regression has a closed-form solution!!

Solve for $\nabla J(w, b) = 0$ for w^*, b^*

optimal estimator
 $\hat{w}$
 $\hat{b}$



$$\nabla J(w, b) = \begin{bmatrix} \frac{\partial J}{\partial w} \\ \frac{\partial J}{\partial b} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Optimization

Gradients

0) Form ✓

1) equation
for
gradient

2) Set

$$\nabla J = 0$$

and
solve for w b

Closed
form

$$J(w, b) = \frac{1}{2} \left[\left(y^{(1)} - (wx^{(1)} + b) \right)^2 + \left(y^{(2)} - (wx^{(2)} + b) \right)^2 \right]$$

$$\frac{dJ}{dw} =$$

$$\frac{dJ}{db}$$

if no closed-form

2) Gradient descent

Calculus Time-out

Partial derivatives

Calculus with Linear Algebra

Vector in, scalar out

Gradient

$y = f(\mathbf{x})$ $y \in \mathbb{R}, \mathbf{x} \in \mathbb{R}^M$

$$\nabla_{\mathbf{x}} f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_M} \end{bmatrix}$$

$\vec{\theta}$

$$\frac{\partial J}{\partial \vec{\theta}}$$

$$\nabla_{\theta} J$$

Calculus with Linear Algebra

Functions with linear algebra

scalar

$$y = f(x)$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = f(x)$$

$$\begin{bmatrix} Y_{1,1} & \cdots & Y_{1,K} \\ \vdots & \ddots & \vdots \\ Y_{N,1} & \cdots & Y_{N,K} \end{bmatrix} = f(x)$$

*input
vector*

$$y = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}\right)$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}\right)$$

$$\begin{bmatrix} Y_{1,1} & \cdots & Y_{1,K} \\ \vdots & \ddots & \vdots \\ Y_{N,1} & \cdots & Y_{N,K} \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}\right)$$

matrix

$$y = f\left(\begin{bmatrix} X_{1,1} & \cdots & X_{1,L} \\ \vdots & \ddots & \vdots \\ X_{M,1} & \cdots & X_{M,L} \end{bmatrix}\right)$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = f\left(\begin{bmatrix} X_{1,1} & \cdots & X_{1,L} \\ \vdots & \ddots & \vdots \\ X_{M,1} & \cdots & X_{M,L} \end{bmatrix}\right)$$

$$\begin{bmatrix} Y_{1,1} & \cdots & Y_{1,K} \\ \vdots & \ddots & \vdots \\ Y_{N,1} & \cdots & Y_{N,K} \end{bmatrix} = f\left(\begin{bmatrix} X_{1,1} & \cdots & X_{1,L} \\ \vdots & \ddots & \vdots \\ X_{M,1} & \cdots & X_{M,L} \end{bmatrix}\right)$$

output

Calculus with Linear Algebra

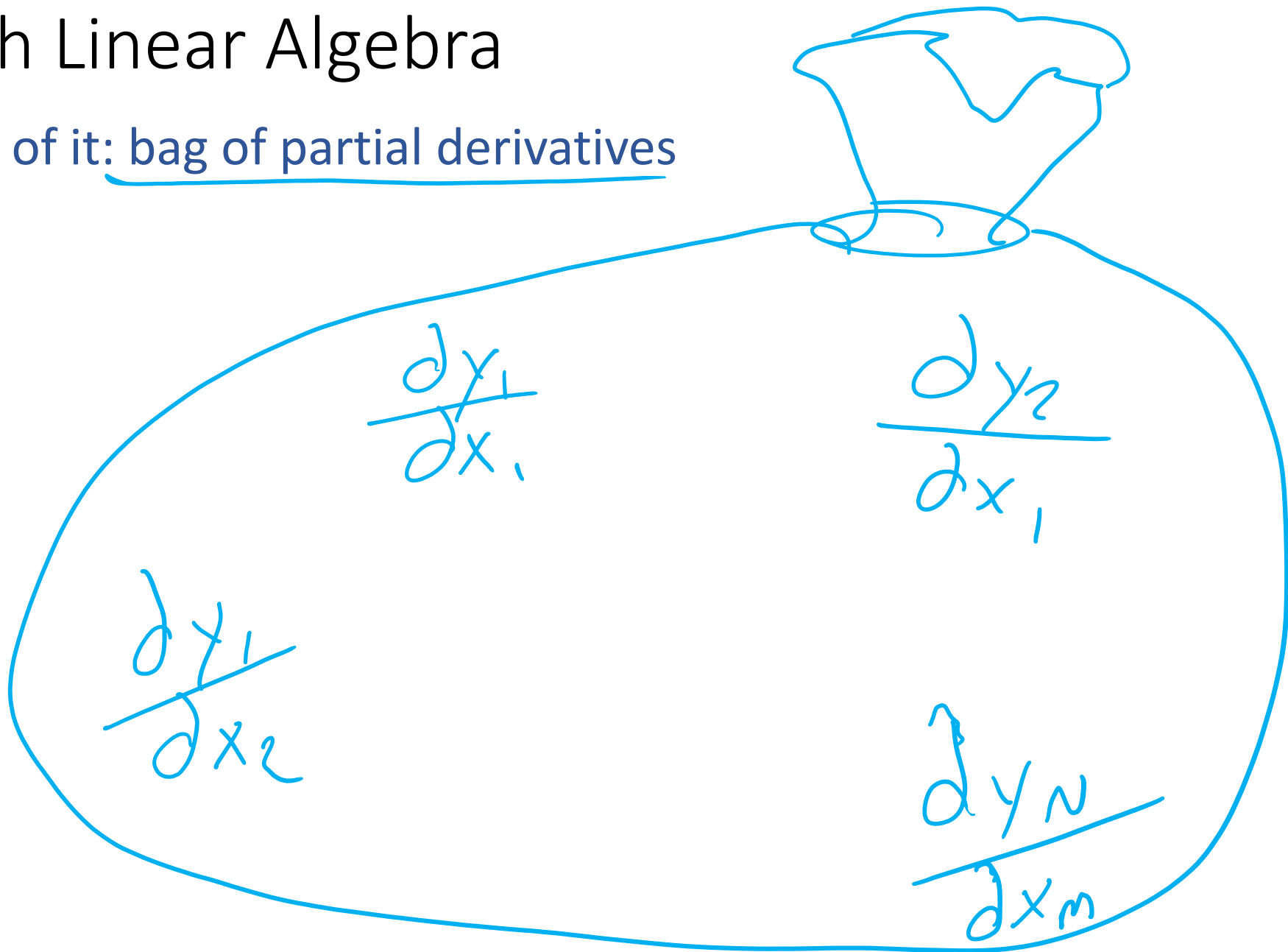
One way to think of it: bag of partial derivatives

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}\right)$$



$M \times N$

$$\frac{\partial \vec{y}}{\partial \vec{x}} = \begin{matrix} M \times N \\ N \times M \end{matrix}$$



Calculus with Linear Algebra

One way to think of it: bag of partial derivatives

$$\begin{bmatrix} Y_{1,1} & \cdots & Y_{1,K} \\ \vdots & \ddots & \vdots \\ Y_{N,K} & \cdots & Y_{N,K} \end{bmatrix} = f \left(\begin{bmatrix} X_{1,1} & \cdots & X_{1,L} \\ \vdots & \ddots & \vdots \\ X_{M,1} & \cdots & X_{M,L} \end{bmatrix} \right)$$

$N \times K$

$M \times L$

$N \cdot K \cdot M \cdot L$

$$\frac{\partial Y_{11}}{\partial X_{11}}$$

$$\frac{\partial Y_{22}}{\partial X_{22}}$$

Numerator Layout

input x

output y

Y

$y = f(x)$	$x \in \mathbb{R}$	$\mathbf{x} \in \mathbb{R}^M$	$X \in \mathbb{R}^{M \times L}$
$y \in \mathbb{R}$	$\frac{\partial y}{\partial x} \in \mathbb{R}$ S	$\frac{\partial y}{\partial \mathbf{x}} \in \mathbb{R}^{1 \times M}$ ✓ Output by input	$\frac{\partial y}{\partial X} \in \mathbb{R}^{L \times M}$ M Transpose of input
$\mathbf{y} \in \mathbb{R}^N$	$\frac{\partial \mathbf{y}}{\partial x} \in \mathbb{R}^{N \times 1}$ ✓ Output by input	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \in \mathbb{R}^{N \times M}$ M Output by input	$\frac{\partial \mathbf{y}}{\partial X} \in \mathbb{R}^?$?
$Y \in \mathbb{R}^{N \times K}$	$\frac{\partial Y}{\partial x} \in \mathbb{R}^{N \times K}$ M Size of output	$\frac{\partial Y}{\partial \mathbf{x}} \in \mathbb{R}^?$?	$\frac{\partial Y}{\partial X} \in \mathbb{R}^?$?

Denominator Layout

$y = f(x)$	$x \in \mathbb{R}$	$\mathbf{x} \in \mathbb{R}^M$	$X \in \mathbb{R}^{M \times L}$
$y \in \mathbb{R}$	$\frac{\partial y}{\partial x} \in \mathbb{R}$	$\frac{\partial y}{\partial \mathbf{x}} \in \mathbb{R}^{M \times 1}$ Input by output	$\frac{\partial y}{\partial X} \in \mathbb{R}^{M \times L}$ Size of input
$\mathbf{y} \in \mathbb{R}^N$	$\frac{\partial \mathbf{y}}{\partial x} \in \mathbb{R}^{1 \times N}$ Input by output	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \in \mathbb{R}^{M \times N}$ Input by output	$\frac{\partial \mathbf{y}}{\partial X} \in \mathbb{R}^?$?
$Y \in \mathbb{R}^{N \times K}$	$\frac{\partial Y}{\partial x} \in \mathbb{R}^{K \times N}$ Transpose of output	$\frac{\partial Y}{\partial \mathbf{x}} \in \mathbb{R}^?$?	$\frac{\partial Y}{\partial X} \in \mathbb{R}^?$?

Calculus with Linear Algebra

Jacobian: Vector in, vector out

→ Numerator-layout

$$\mathbf{y} = f(\mathbf{x}) \quad \mathbf{y} \in \mathbb{R}^N, \quad \mathbf{x} \in \mathbb{R}^M, \quad \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \in \mathbb{R}^{N \times M}$$

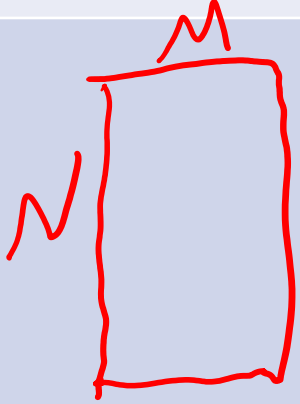
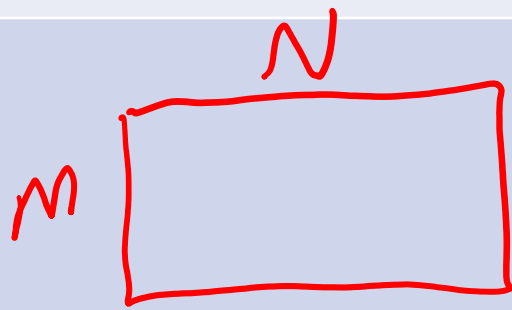
Calculus with Linear Algebra

Numerator-layout vs denominator-layout

Vector in, vector out

$\mathbf{y} = f(\mathbf{x})$ $\mathbf{y} \in \mathbb{R}^N$, $\mathbf{x} \in \mathbb{R}^M$ (assume $N > M$ for illustrative purposes)

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}\right)$$

Numerator layout	Denominator layout
Number of outputs $\times$ number of inputs	Number of inputs $\times$ number of outputs
$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \in \mathbb{R}^{N \times M}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \in \mathbb{R}^{M \times N}$
 <i>Jacobian</i>	

Calculus with Linear Algebra

Vector in, scalar out

Numerator-layout

$$y = f(\mathbf{x}) \quad y \in \mathbb{R}, \quad \mathbf{x} \in \mathbb{R}^M, \quad \frac{\partial y}{\partial \mathbf{x}} \in \mathbb{R}^{1 \times M}$$

$$y = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}\right)$$

Calculus with Linear Algebra

Numerator-layout vs denominator-layout

Vector in, scalar out

$$y = f(\mathbf{x}) \quad y \in \mathbb{R}, \quad \mathbf{x} \in \mathbb{R}^M$$

$$y = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}\right)$$

Numerator layout	Denominator layout
Number of outputs $\times$ number of inputs	Number of inputs $\times$ number of outputs
$\frac{\partial y}{\partial \mathbf{x}} \in \mathbb{R}^{1 \times M}$	$\frac{\partial y}{\partial \mathbf{x}} \in \mathbb{R}^{M \times 1}$

Calculus with Linear Algebra

Scalar in, vector out

Numerator-layout

$$\mathbf{y} = f(x) \quad \mathbf{y} \in \mathbb{R}^N, \quad x \in \mathbb{R}, \quad \frac{\partial \mathbf{y}}{\partial x} \in \mathbb{R}^{N \times 1}$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = f(x)$$

Calculus with Linear Algebra

Gradient: Vector in, scalar out

Transpose of numerator-layout

$$y = f(\mathbf{x}) \quad y \in \mathbb{R}, \quad \mathbf{x} \in \mathbb{R}^M, \quad \frac{\partial y}{\partial \mathbf{x}} \in \mathbb{R}^{1 \times M}, \quad \nabla_{\mathbf{x}} f \in \mathbb{R}^{M \times 1}$$

$$y = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix}\right)$$

Calculus with Linear Algebra

Matrix in, scalar out

Numerator layout: Transpose of matrix dimensions

Denominator layout: Keep same dimensions as matrix

$$y = f(\mathbf{X}) \quad y \in \mathbb{R}, \quad \mathbf{X} \in \mathbb{R}^{N \times M}, \quad \frac{\partial y}{\partial \mathbf{X}} \in \mathbb{R}^{N \times M} \text{ (denominator layout)}$$

$$y = f\left(\begin{bmatrix} X_{1,1} & \cdots & X_{1,L} \\ \vdots & \ddots & \vdots \\ X_{M,1} & \cdots & X_{M,L} \end{bmatrix}\right)$$

Exercise

$$\tilde{\vec{\theta}} \leftarrow \vec{\theta} - \alpha \nabla J(\vec{\theta})$$

Suppose we have a function that takes in a vector and squares each element individually, returning another vector, $\mathbf{y} = f(\mathbf{x})$.

$$f\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) \rightarrow \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \end{bmatrix} \quad \begin{matrix} y_1 \\ y_2 \\ y_3 \end{matrix} \quad \text{Example: } f\left(\begin{bmatrix} 7 \\ 3 \\ 5 \end{bmatrix}\right) \rightarrow \begin{bmatrix} 49 \\ 9 \\ 25 \end{bmatrix}$$

What is $\partial \mathbf{y} / \partial \mathbf{x}$? (use numerator layout)

$$\begin{bmatrix} 2x_1 \\ 2x_2 \\ 2x_3 \end{bmatrix}$$

$$\frac{\partial y_2}{\partial x_1} \begin{bmatrix} 2x_1 & 0 & 0 \\ 0 & 2x_2 & 0 \\ 0 & 0 & 2x_3 \end{bmatrix}$$

Derivatives of Functions with Respect to Vectors

	Numerator layout	Denominator layout	Notes
$\frac{\partial}{\partial \mathbf{v}} \mathbf{v}$	I_N	I_N	$\mathbf{v} \in \mathbb{R}^N$
$\frac{\partial}{\partial \mathbf{v}} \mathbf{v}^T$	I_N	I_N	$\mathbf{v} \in \mathbb{R}^N$
$\frac{\partial}{\partial \mathbf{v}} t\mathbf{v}$	tI_N	tI_N	$\mathbf{v} \in \mathbb{R}^N$
$\frac{\partial}{\partial \mathbf{u}} \mathbf{u}^T \mathbf{v}$	$\mathbf{v}^T$	$\mathbf{v}$	
$\frac{\partial}{\partial \mathbf{v}} \mathbf{u}^T \mathbf{v}$	$\mathbf{u}^T$	$\mathbf{u}$	
$\frac{\partial}{\partial \mathbf{v}} \mathbf{v}^T \mathbf{v}$	$2\mathbf{v}^T$	$2\mathbf{v}$	
$\frac{\partial}{\partial \mathbf{v}} \mathbf{v}^T A \mathbf{v}$	$\mathbf{v}^T (A + A^T)$	$(A + A^T) \mathbf{v}$	
$\frac{\partial}{\partial \mathbf{v}} \mathbf{v}^T A \mathbf{v}$	$2\mathbf{v}^T A$	$2A \mathbf{v}$	If $A = A^T$
$\frac{\partial}{\partial \mathbf{v}} A \mathbf{v}$	A	A^T	
$\frac{\partial}{\partial \mathbf{v}} \mathbf{v}^T A$	A^T	A	



Numerator Layout

$y = f(x)$	$x \in \mathbb{R}$	$\mathbf{x} \in \mathbb{R}^M$	$X \in \mathbb{R}^{M \times L}$
$y \in \mathbb{R}$	$\frac{\partial y}{\partial x} \in \mathbb{R}$	$\frac{\partial y}{\partial \mathbf{x}} \in \mathbb{R}^{1 \times M}$ Output by input	$\frac{\partial y}{\partial X} \in \mathbb{R}^{L \times M}$ Transpose of input
$\mathbf{y} \in \mathbb{R}^N$	$\frac{\partial \mathbf{y}}{\partial x} \in \mathbb{R}^{N \times 1}$ Output by input	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \in \mathbb{R}^{N \times M}$ Output by input	$\frac{\partial \mathbf{y}}{\partial X} \in \mathbb{R}^?$?
$Y \in \mathbb{R}^{N \times K}$	$\frac{\partial Y}{\partial x} \in \mathbb{R}^{N \times K}$ Size of output	$\frac{\partial Y}{\partial \mathbf{x}} \in \mathbb{R}^?$?	$\frac{\partial Y}{\partial X} \in \mathbb{R}^?$?

Denominator Layout

$y = f(x)$	$x \in \mathbb{R}$	$\mathbf{x} \in \mathbb{R}^M$	$X \in \mathbb{R}^{M \times L}$
$y \in \mathbb{R}$	$\frac{\partial y}{\partial x} \in \mathbb{R}$	$\frac{\partial y}{\partial \mathbf{x}} \in \mathbb{R}^{M \times 1}$ Input by output	$\frac{\partial y}{\partial X} \in \mathbb{R}^{M \times L}$ Size of input
$\mathbf{y} \in \mathbb{R}^N$	$\frac{\partial \mathbf{y}}{\partial x} \in \mathbb{R}^{1 \times N}$ Input by output	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \in \mathbb{R}^{M \times N}$ Input by output	$\frac{\partial \mathbf{y}}{\partial X} \in \mathbb{R}^?$?
$Y \in \mathbb{R}^{N \times K}$	$\frac{\partial Y}{\partial x} \in \mathbb{R}^{K \times N}$ Transpose of output	$\frac{\partial Y}{\partial \mathbf{x}} \in \mathbb{R}^?$?	$\frac{\partial Y}{\partial X} \in \mathbb{R}^?$?

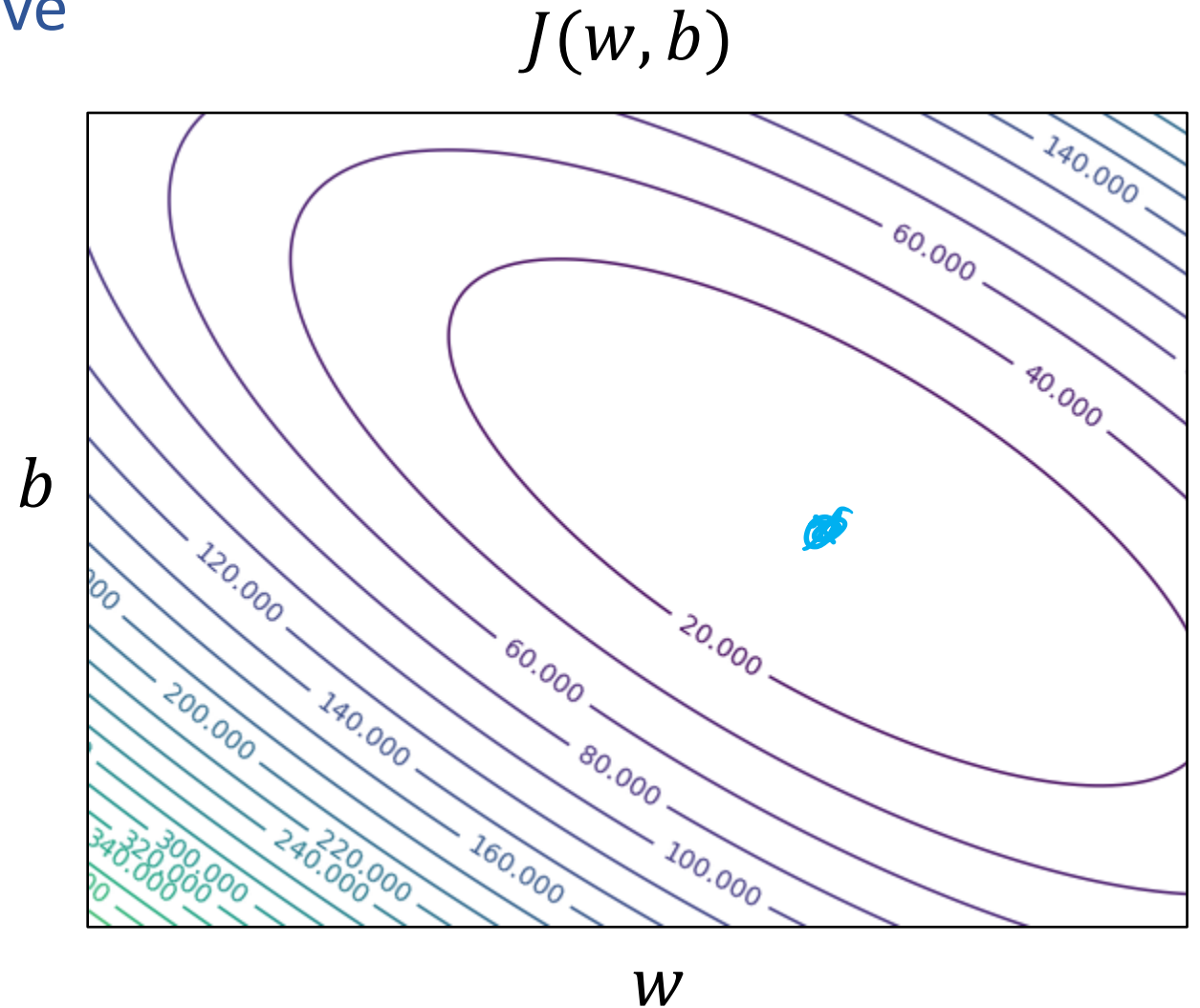
Linear Regression

Closed-form Solution

Linear Regression

Methods for optimizing the objective

- Closed-form solution



Linear Regression

Expanding objective before computing gradient

$$\begin{aligned} J(\boldsymbol{\theta}) &= \frac{1}{N} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 \\ &= \frac{1}{N} (\mathbf{y} - X\boldsymbol{\theta})^T (\mathbf{y} - X\boldsymbol{\theta}) \\ &= \frac{1}{N} (\mathbf{y}^T - \boldsymbol{\theta}^T X^T) (\mathbf{y} - X\boldsymbol{\theta}) \\ &= \frac{1}{N} (\mathbf{y}^T \mathbf{y} - \boldsymbol{\theta}^T X^T \mathbf{y} - \mathbf{y}^T X \boldsymbol{\theta} + \boldsymbol{\theta}^T X^T X \boldsymbol{\theta}) \\ &= \frac{1}{N} (\mathbf{y}^T \mathbf{y} - 2\boldsymbol{\theta}^T X^T \mathbf{y} + \boldsymbol{\theta}^T X^T X \boldsymbol{\theta}) \end{aligned}$$

Linear Regression

Gradient of objective with respect to parameters

$$\begin{aligned} J(\boldsymbol{\theta}) &= \frac{1}{N} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 \\ &= \frac{1}{N} (\mathbf{y}^T \mathbf{y} - 2\boldsymbol{\theta}^T X^T \mathbf{y} + \boldsymbol{\theta}^T X^T X \boldsymbol{\theta}) \end{aligned}$$

$$\nabla J(\boldsymbol{\theta}) = \frac{1}{N}$$

$$\frac{\partial \mathbf{z}^T \mathbf{u}}{\partial \mathbf{z}} = \mathbf{u}$$

-- or --

$$\frac{\partial \mathbf{z}^T \mathbf{u}}{\partial \mathbf{z}} = \mathbf{u}^T$$

$$\frac{\partial \mathbf{z}^T A \mathbf{z}}{\partial \mathbf{z}} = (A + A^T) \mathbf{z}$$

-- or --

$$\frac{\partial \mathbf{z}^T A \mathbf{z}}{\partial \mathbf{z}} = \mathbf{z}^T (A + A^T)$$

Linear Regression $y \in \mathbb{R}^N$ $X \in \mathbb{R}^{N \times M}$ $\theta \in \mathbb{R}^M$

$$\frac{\partial z^T u}{\partial z} = u$$

Gradient of objective with respect to parameters

-- or --

$$J(\theta) = \frac{1}{N} \|y - X\theta\|_2^2$$

$$\frac{\partial z^T u}{\partial z} = u^T$$

$$= \frac{1}{N} (y^T y - 2\theta^T X^T y + \theta^T X^T X \theta)$$

$$\nabla J(\theta) = \frac{1}{N} (0 - 2X^T y + 2\theta^T X^T X) \leftarrow \ddot{}$$

$$\frac{\partial z^T A z}{\partial z} = (A + A^T)z$$

-- or --

$$\begin{matrix} [N \times N] & [N \times 1] \\ \hline & [1 \times 1] \end{matrix}$$

$$[M \times 1]$$

$$[1 \times M] [M \times N] [N \times M]$$

$$[1 \times M]$$

$$\frac{\partial z^T A z}{\partial z} = z^T (A + A^T)$$

Linear Regression

Gradient of objective with respect to parameters

$$J(\boldsymbol{\theta}) = \frac{1}{N} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$$
$$= \frac{1}{N} (\mathbf{y}^T \mathbf{y} - 2\boldsymbol{\theta}^T X^T \mathbf{y} + \boldsymbol{\theta}^T X^T X \boldsymbol{\theta})$$

$$\nabla J(\boldsymbol{\theta}) = \frac{1}{N} (0 \quad -2X^T \mathbf{y} + \cancel{2\boldsymbol{\theta}^T X^T X})$$

$$= \frac{1}{N} (0 \quad -2X^T \mathbf{y} + 2X^T X \boldsymbol{\theta})$$

$$\frac{\partial \mathbf{z}^T \mathbf{u}}{\partial \mathbf{z}} = \mathbf{u}$$

-- or --

$$\frac{\partial \mathbf{z}^T \mathbf{u}}{\partial \mathbf{z}} = \mathbf{u}^T$$

$$\frac{\partial \mathbf{z}^T A \mathbf{z}}{\partial \mathbf{z}} = (A + A^T) \mathbf{z}$$

-- or --

$$\frac{\partial \mathbf{z}^T A \mathbf{z}}{\partial \mathbf{z}} = \mathbf{z}^T (A + A^T)$$

Linear Regression

Gradient of objective with respect to parameters

$$\begin{aligned} J(\boldsymbol{\theta}) &= \frac{1}{N} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 \\ &= \frac{1}{N} (\mathbf{y}^T \mathbf{y} - 2\boldsymbol{\theta}^T X^T \mathbf{y} + \boldsymbol{\theta}^T X^T X \boldsymbol{\theta}) \end{aligned}$$

$$\begin{aligned} \nabla J(\boldsymbol{\theta}) &= \frac{1}{N} (0 \quad -2X^T \mathbf{y} + \cancel{2\boldsymbol{\theta}^T X^T X}) \\ &= \frac{1}{N} (0 \quad -2X^T \mathbf{y} + 2X^T X \boldsymbol{\theta}) \\ &= \frac{2}{N} (-X^T \mathbf{y} + X^T X \boldsymbol{\theta}) \end{aligned}$$

$$\frac{\partial \mathbf{z}^T \mathbf{u}}{\partial \mathbf{z}} = \mathbf{u}$$

-- or --

$$\frac{\partial \mathbf{z}^T \mathbf{u}}{\partial \mathbf{z}} = \mathbf{u}^T$$

$$\frac{\partial \mathbf{z}^T A \mathbf{z}}{\partial \mathbf{z}} = (A + A^T) \mathbf{z}$$

-- or --

$$\frac{\partial \mathbf{z}^T A \mathbf{z}}{\partial \mathbf{z}} = \mathbf{z}^T (A + A^T)$$

Linear Regression

Closed-form solution

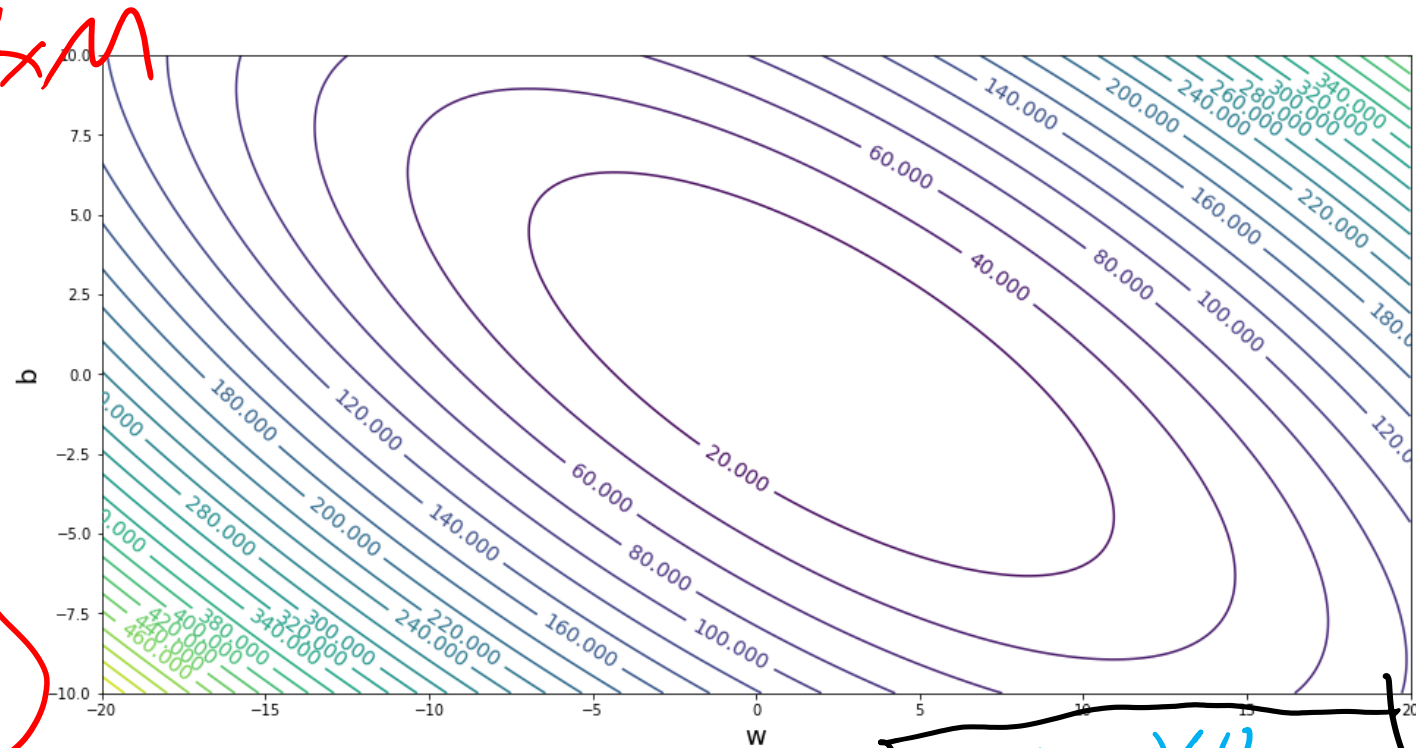
1) $\nabla J(\theta) = \frac{2}{N}(-X^T y + X^T X \theta)$

2) $0 = \frac{2}{N}(-X^T y + X^T X \theta)$

$X^T X \theta = X^T y$ ← Normal Equations

$\hat{\theta} = (X^T X)^{-1} X^T y$

$[M \times M][M \times N][N \times 1] \rightarrow [M \times 1]$



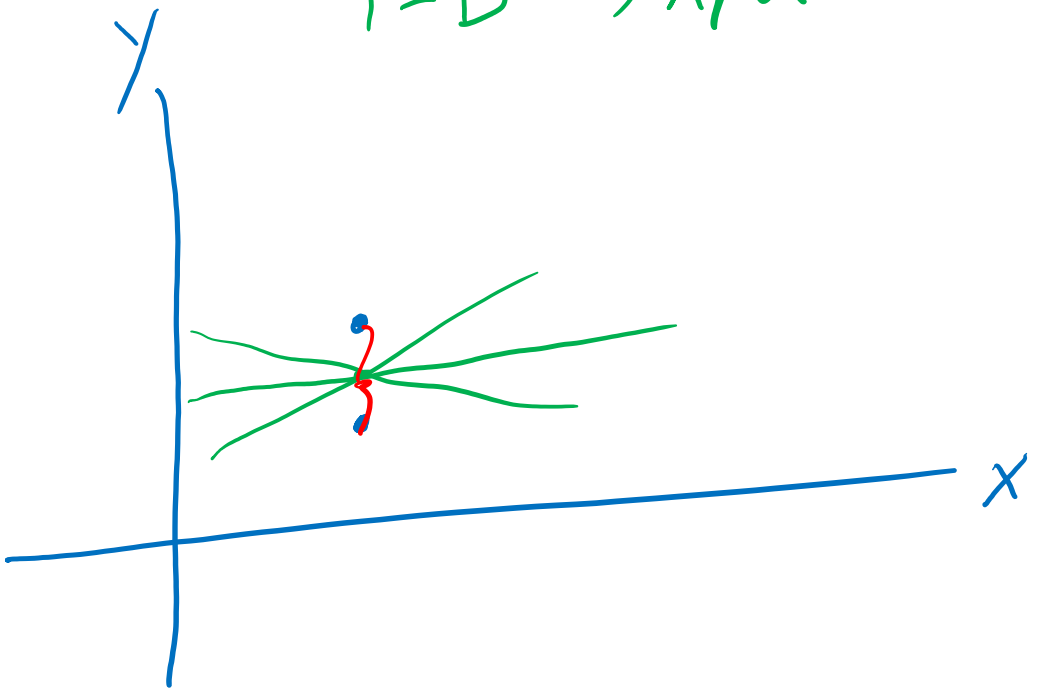
$X = X\theta$
No! $\theta = X^{-1}y$

$X^T X$ invertable

Linear Regression

Number of solutions

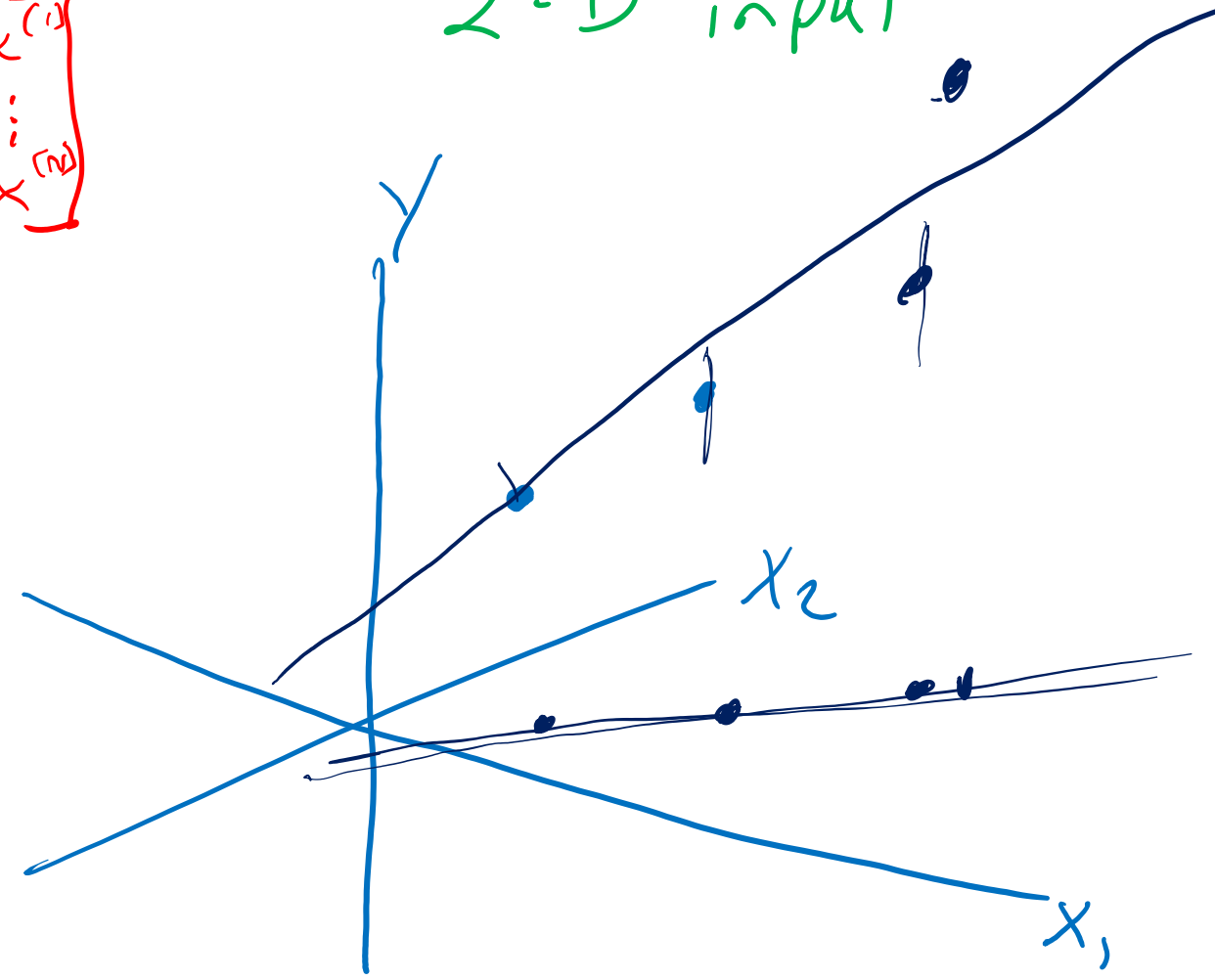
1-D input



$$\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(n)} \end{bmatrix}$$

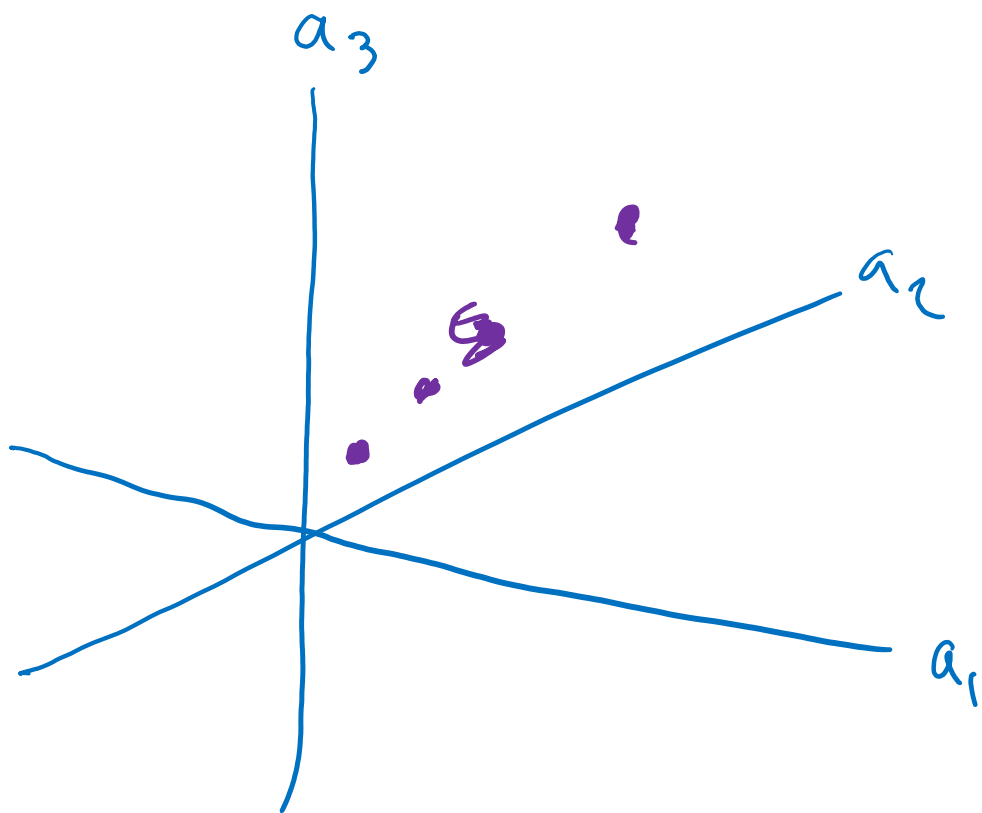
X

2-D input



A Note on Matrix Rank

Underlying dimensionality of the data



$$X^T X \rightarrow A = \begin{bmatrix} a_1 & a_2 & a_3 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \\ 5 & 5 & 5 \\ \hline 3 & 3 & 3 \end{bmatrix}$$

$$\text{Rank}(A) = 1$$

Want $X^T X$
to be full rank
 $\rightarrow$ invertible

Linear Regression

Gradient Descent

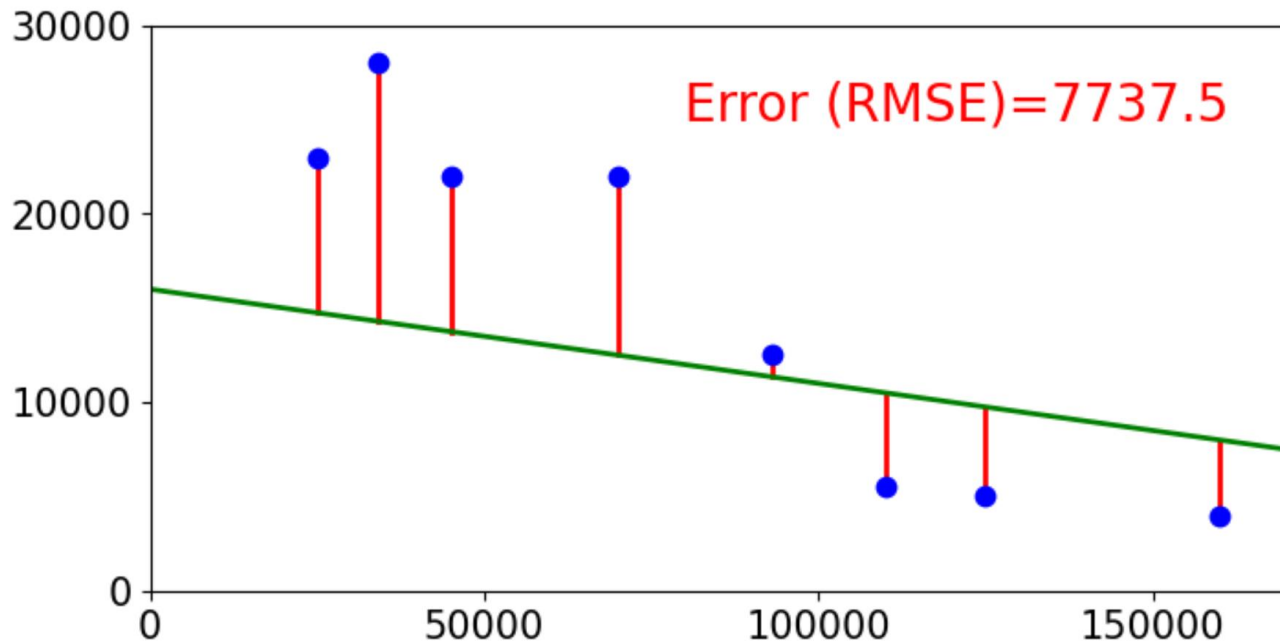
Linear Regression

Demos to optimize the objective

Linear model optimization

[regression_interactive.ipynb](#)

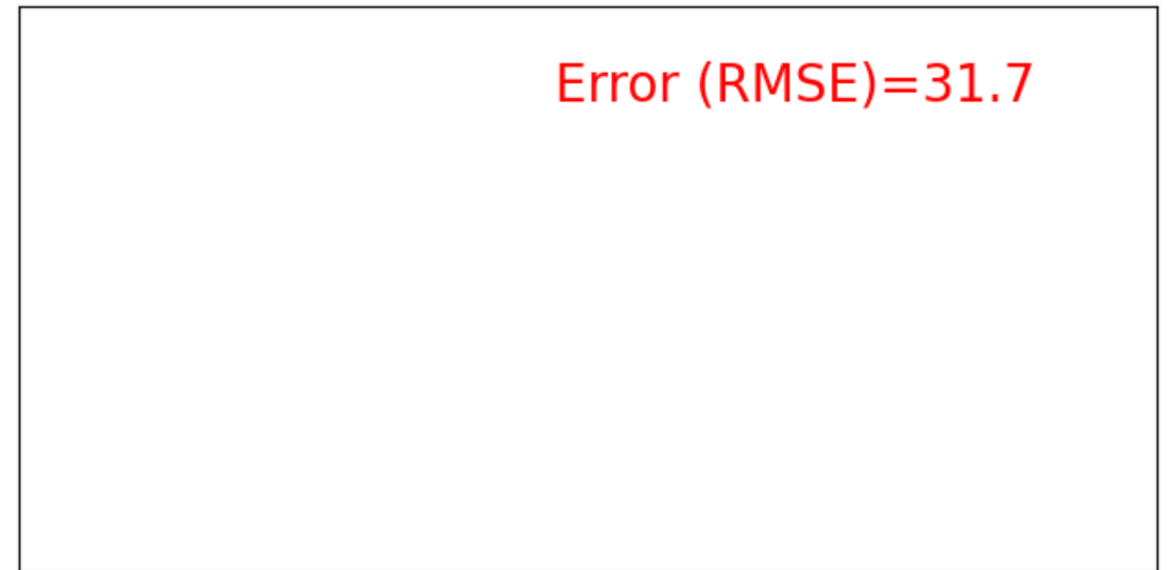
m -0.05
b 16000



Blind version

[regression_blind_interactive.ipynb](#)

m -20.00
b -10.00



Poll 6 (unused)

Open blind optimization on course website

Mess with sliders to find best model

What's the best RMSE that you got?

What's are the corresponding best parameters?

Linear Regression

Methods for optimizing the objective

■ slider search (coordinate descent-ish)

■ Closed-form solution

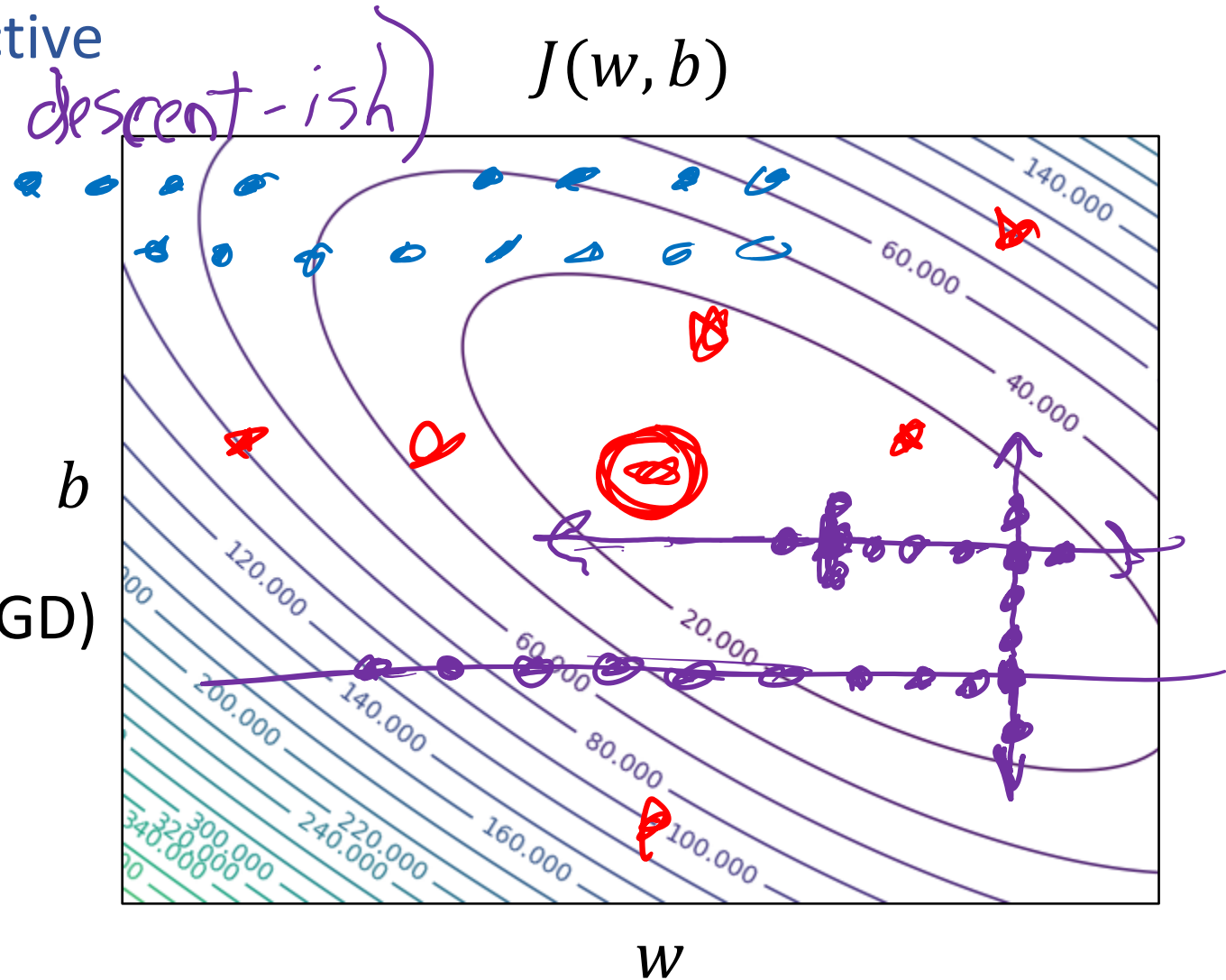
■ Grid search

■ Random search

■ Gradient descent

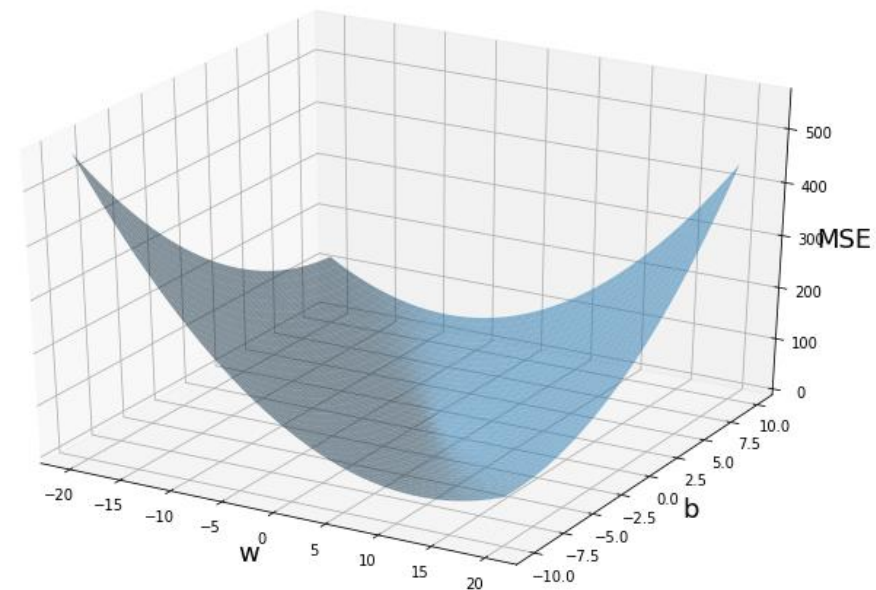
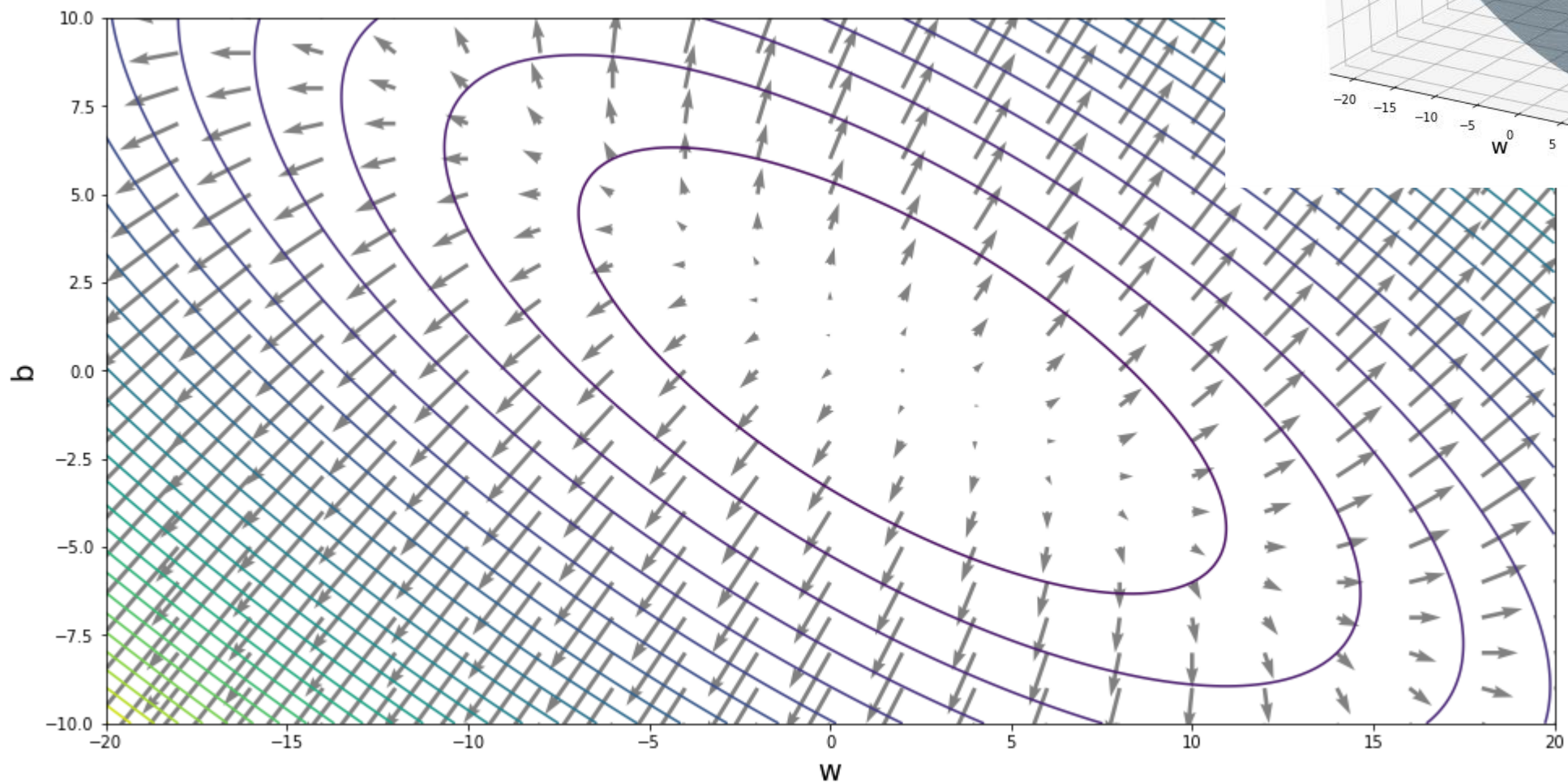
■ Stochastic gradient descent (SGD)

■ Minibatch SGD



Optimization

Gradients

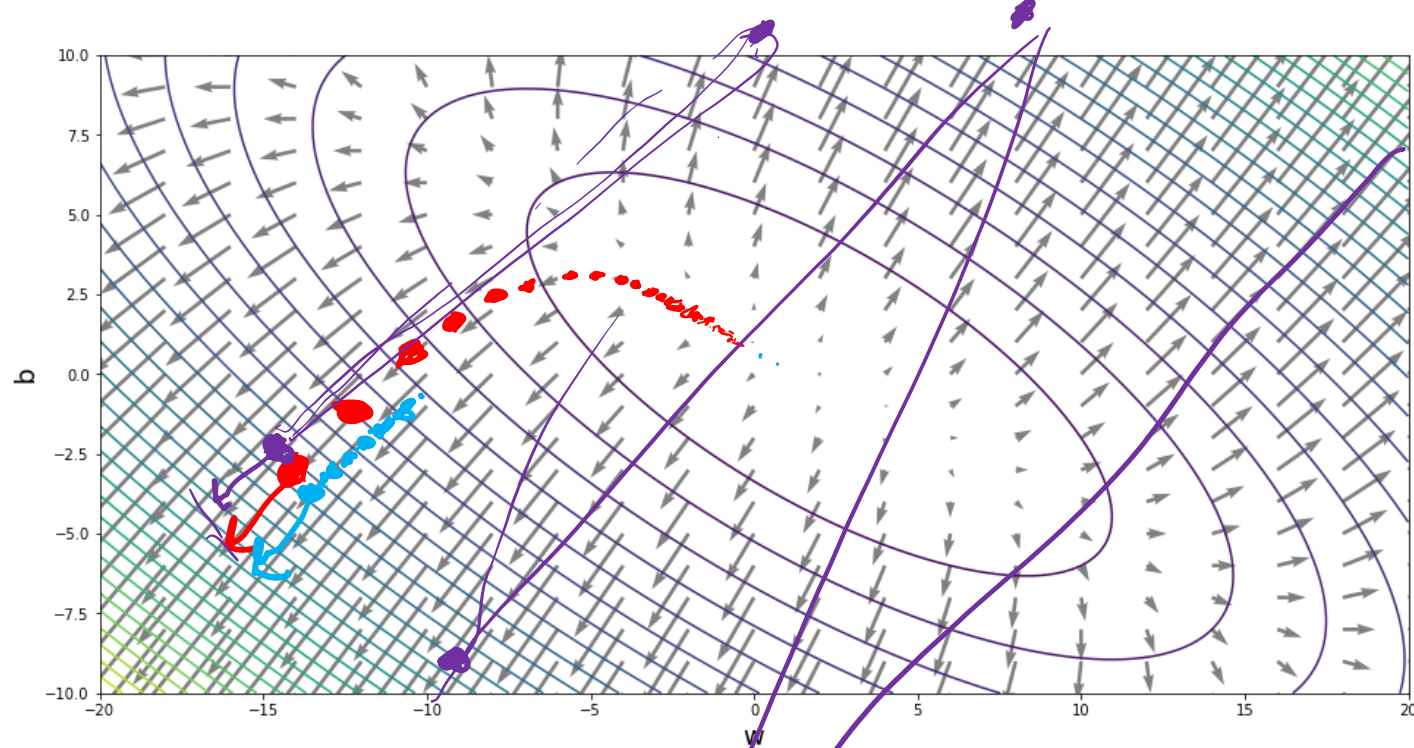


Optimization

Gradient descent

Hyperparameter:

learning rate α (or sometimes γ, η)



While not converged

Compute gradient: $\nabla J = \left[\frac{\partial J}{\partial w}, \frac{\partial J}{\partial b} \right]^T$

Update parameters

$$w \leftarrow w - \alpha \frac{\partial J}{\partial w}$$

$$b \leftarrow b - \alpha \frac{\partial J}{\partial b}$$

$\alpha = 10$ Too big

$\alpha = 1$

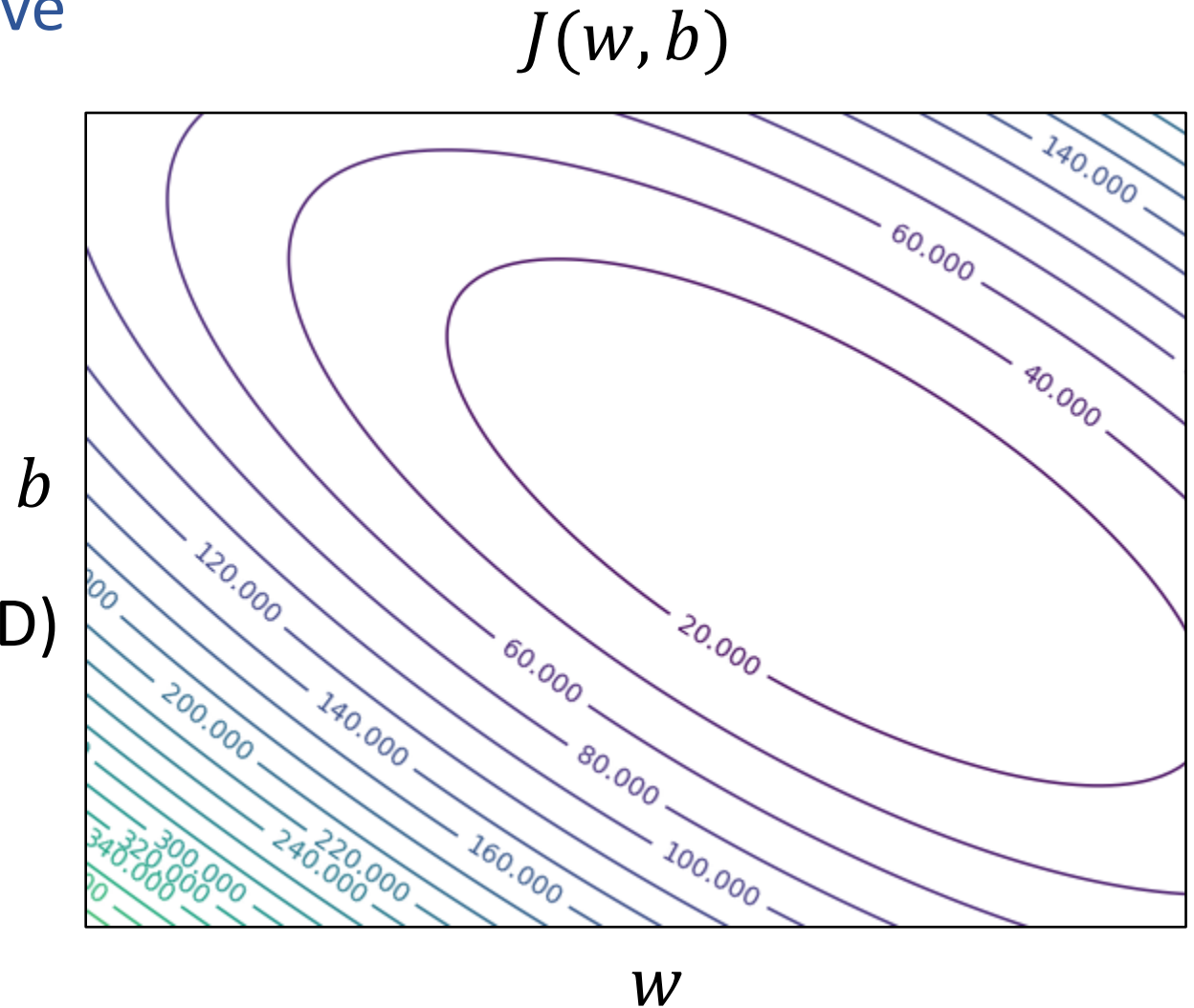
$\alpha = 0.1$ Too small

ADAM

Linear Regression

Methods for optimizing the objective

- Closed-form solution
- Grid search
- Random search
- Gradient descent
- **Stochastic gradient descent (SGD)**
- Minibatch SGD

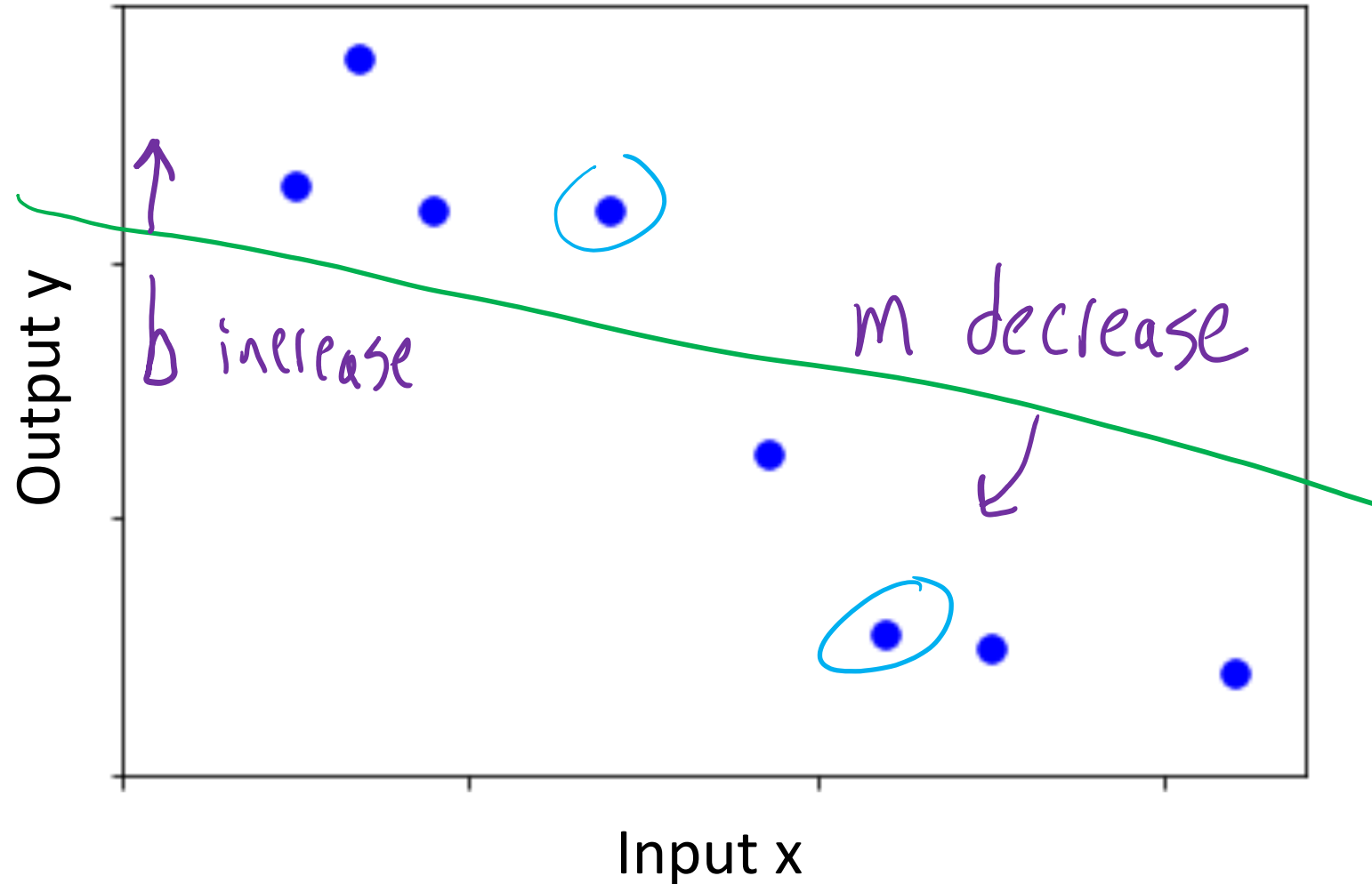


Linear Regression Gradient Descent

What happens in gradient descent when we have $N=1,000,000$ training points?

current model

Picking just one or two random points will allow us to take a reasonable gradient step



Gradient Descent

Batch -> Mini-batch -> Stochastic

Basically, just a difference in how many training points are each time we compute a gradient (and then take a step with that gradient)

(Batch) GD



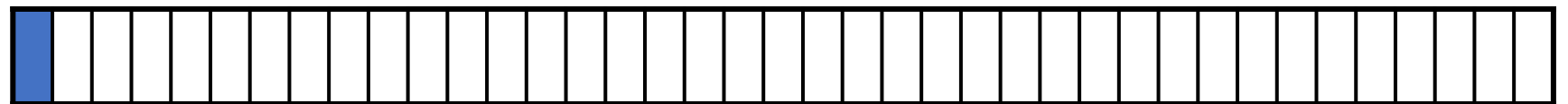
Effective batch size = N

Mini-batch SGD



Batch size = B , e.g., $B = 10$

SGD

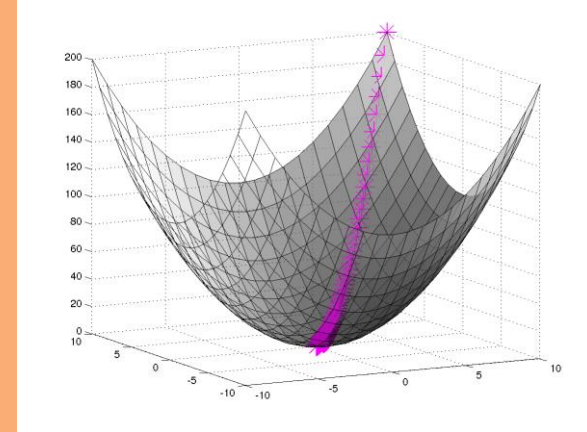


Effective batch size = 1

(Batch) Gradient Descent

Algorithm 1 Gradient Descent

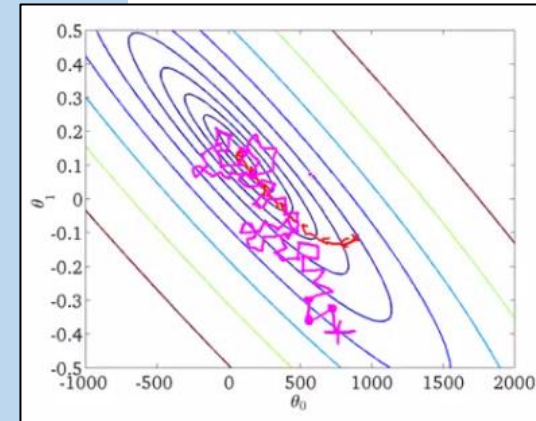
```
1: procedure GD( $\mathcal{D}$ ,  $\theta^{(0)}$ )  
2:    $\theta \leftarrow \theta^{(0)}$   
3:   while not converged do  
4:      $\theta \leftarrow \theta - \gamma \nabla_{\theta} J(\theta)$   
5:   return  $\theta$ 
```



Stochastic Gradient Descent (SGD)

Algorithm 2 Stochastic Gradient Descent (SGD)

```
1: procedure SGD( $\mathcal{D}, \theta^{(0)}$ )  
2:    $\theta \leftarrow \theta^{(0)}$   
3:   while not converged do  
4:      $i \sim \text{Uniform}(\{1, 2, \dots, N\})$   
5:      $\theta \leftarrow \theta - \gamma \nabla_{\theta} J^{(i)}(\theta)$   
6:   return  $\theta$ 
```



We need a per-example objective:

$$\text{Let } J(\theta) = \sum_{i=1}^N J^{(i)}(\theta)$$

(Batch) Gradient Descent

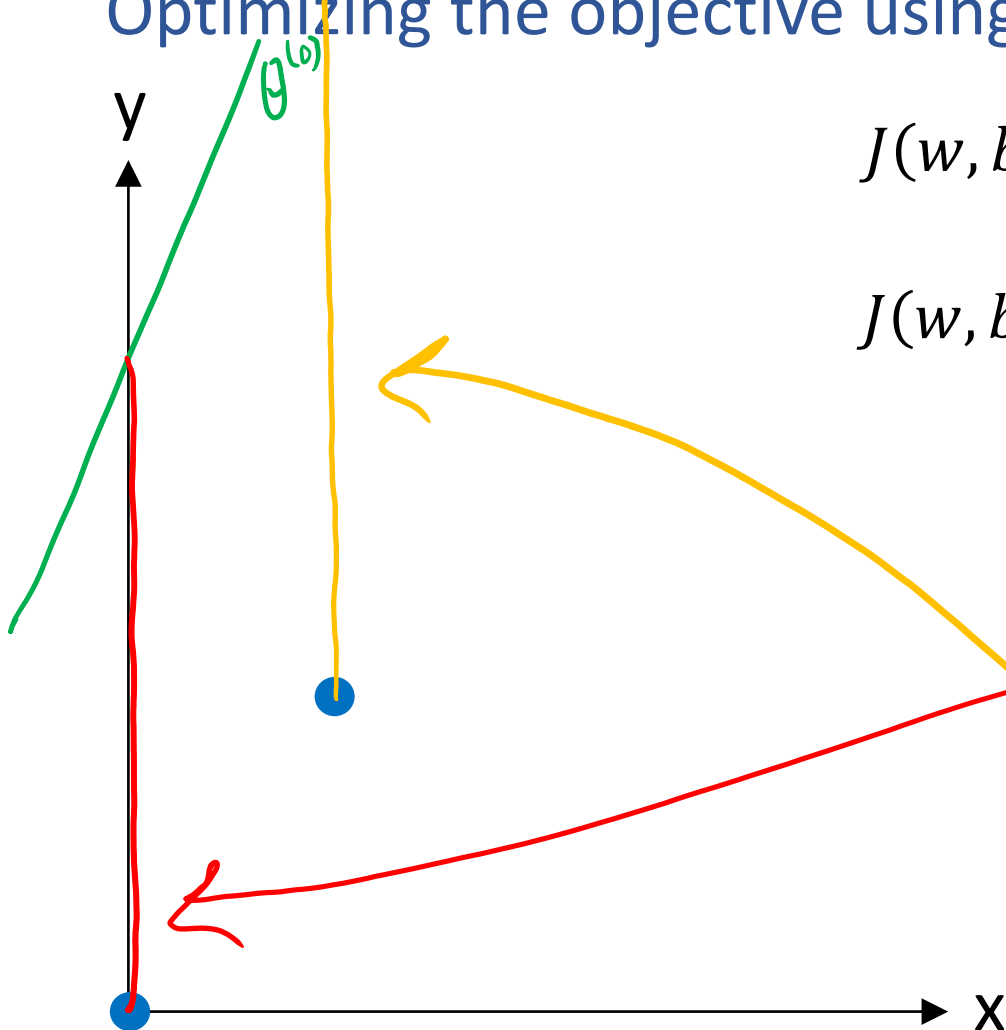
Optimizing the objective using all points (just two in this example)

$$J(w, b) = \frac{1}{2} \left[\left(y^{(1)} - (wx^{(1)} + b) \right)^2 + \left(y^{(2)} - (wx^{(2)} + b) \right)^2 \right]$$

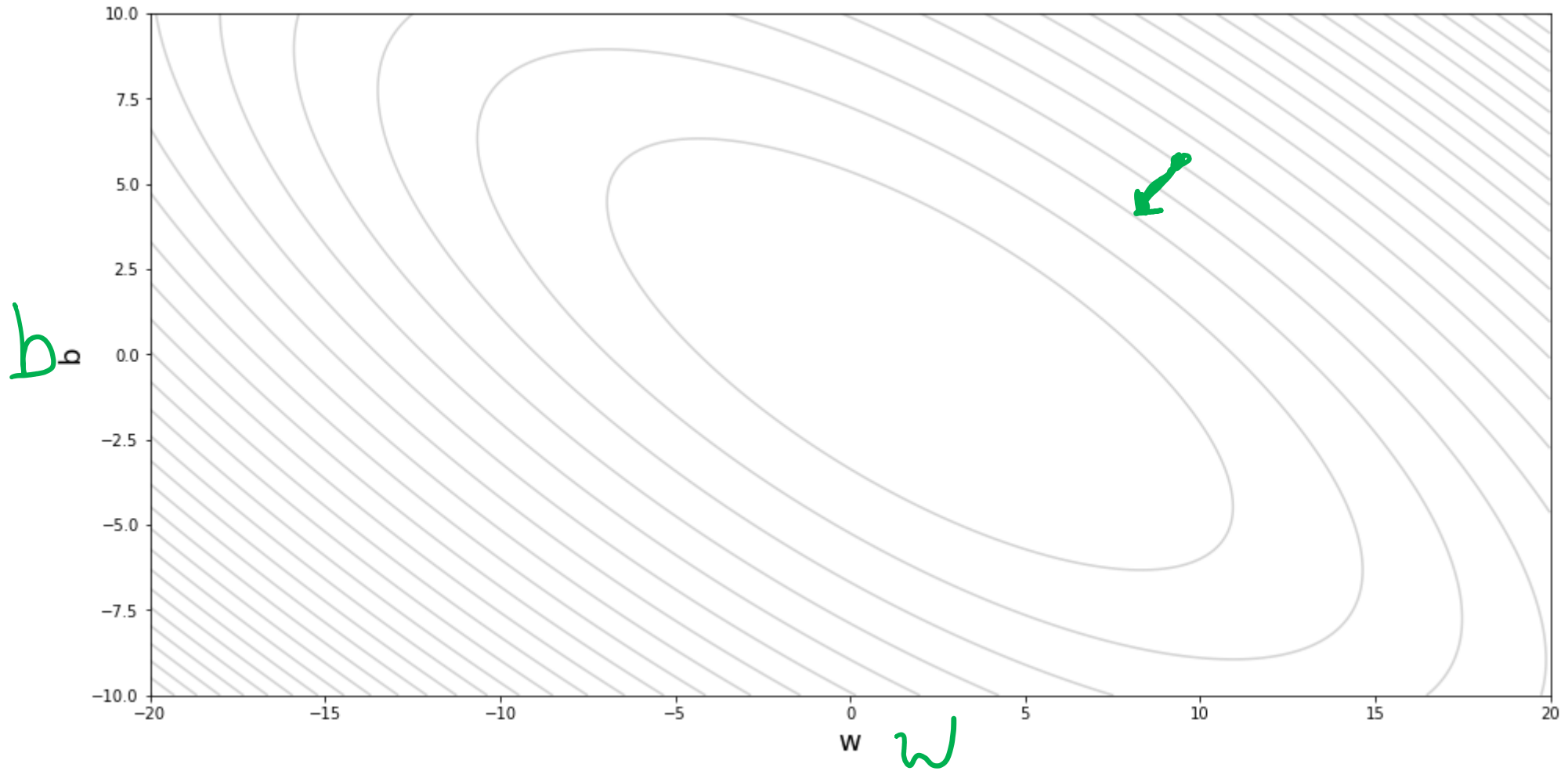
$$J(w, b) = \frac{1}{2} \left[J^{(1)}(w, b) + J^{(2)}(w, b) \right]$$

$$J^{(1)}(w, b) = \left(y^{(1)} - (wx^{(1)} + b) \right)^2$$

$$J^{(2)}(w, b) = \left(y^{(2)} - (wx^{(2)} + b) \right)^2$$



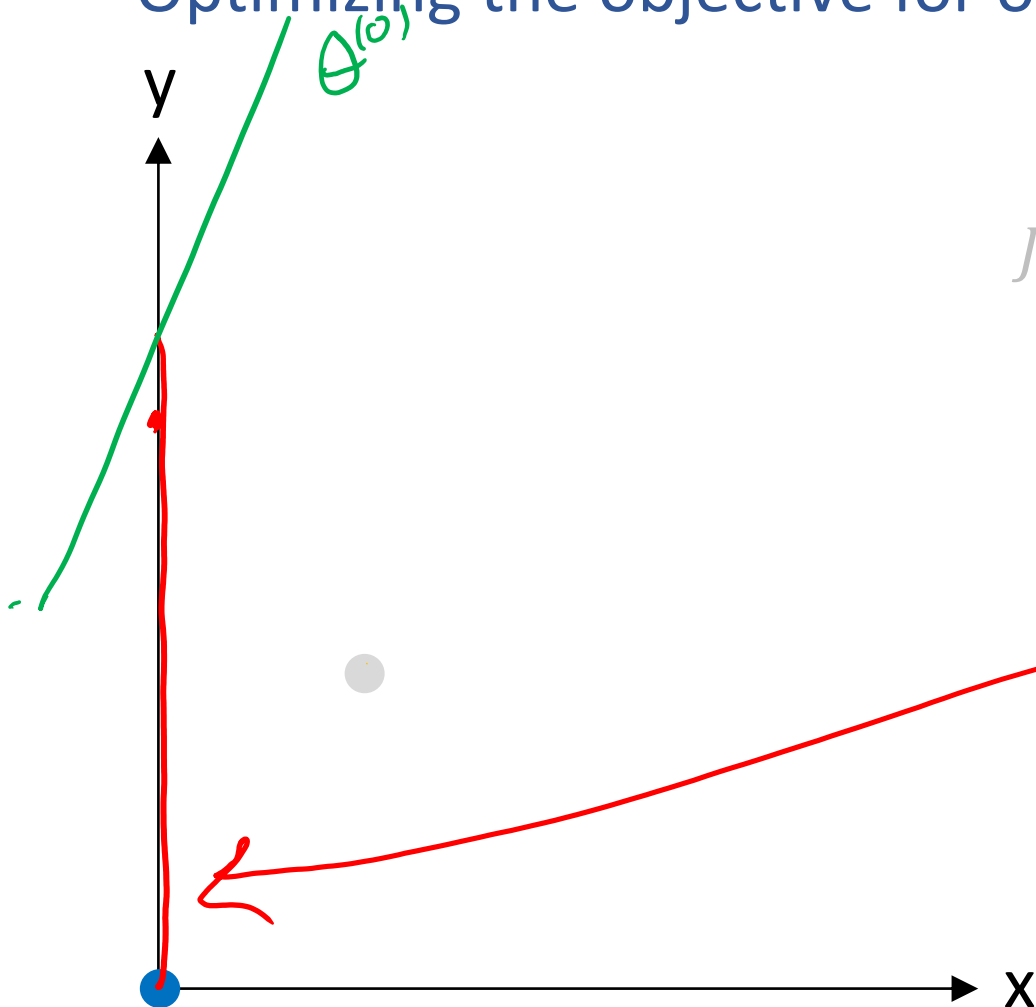
(Batch) Gradient Descent



$$J(w, b) = \frac{1}{2} \left[J^{(1)}(w, b) + J^{(2)}(w, b) \right]$$

Stochastic Gradient Descent

Optimizing the objective for one point at a time

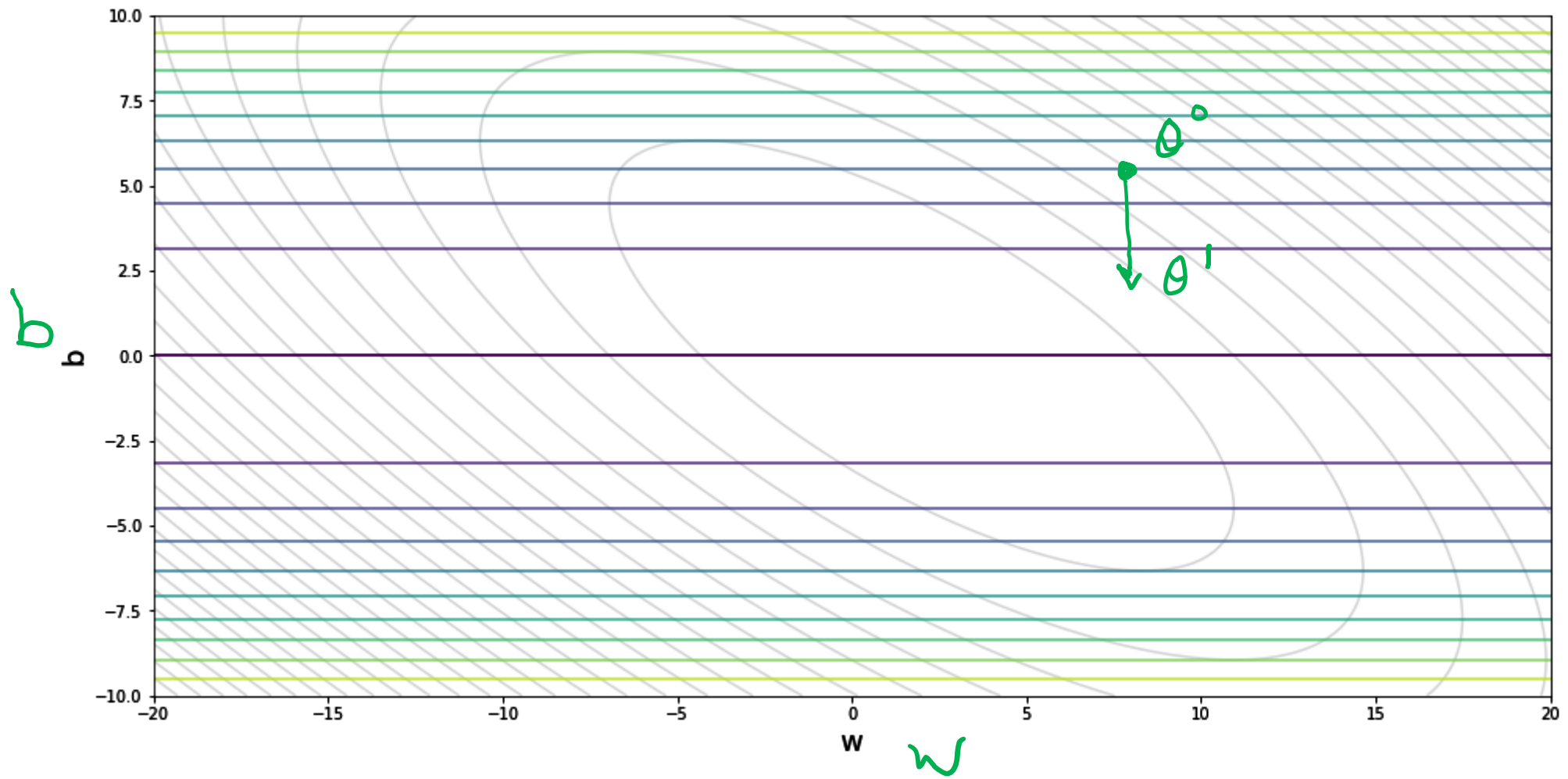


$$J(w, b) = \frac{1}{2} [J^{(1)}(w, b) + J^{(2)}(w, b)]$$

$$J^{(1)}(w, b) = \left(y^{(1)} - (wx^{(1)} + b) \right)^2$$

$$J^{(2)}(w, b) = \left(y^{(2)} - (wx^{(2)} + b) \right)^2$$

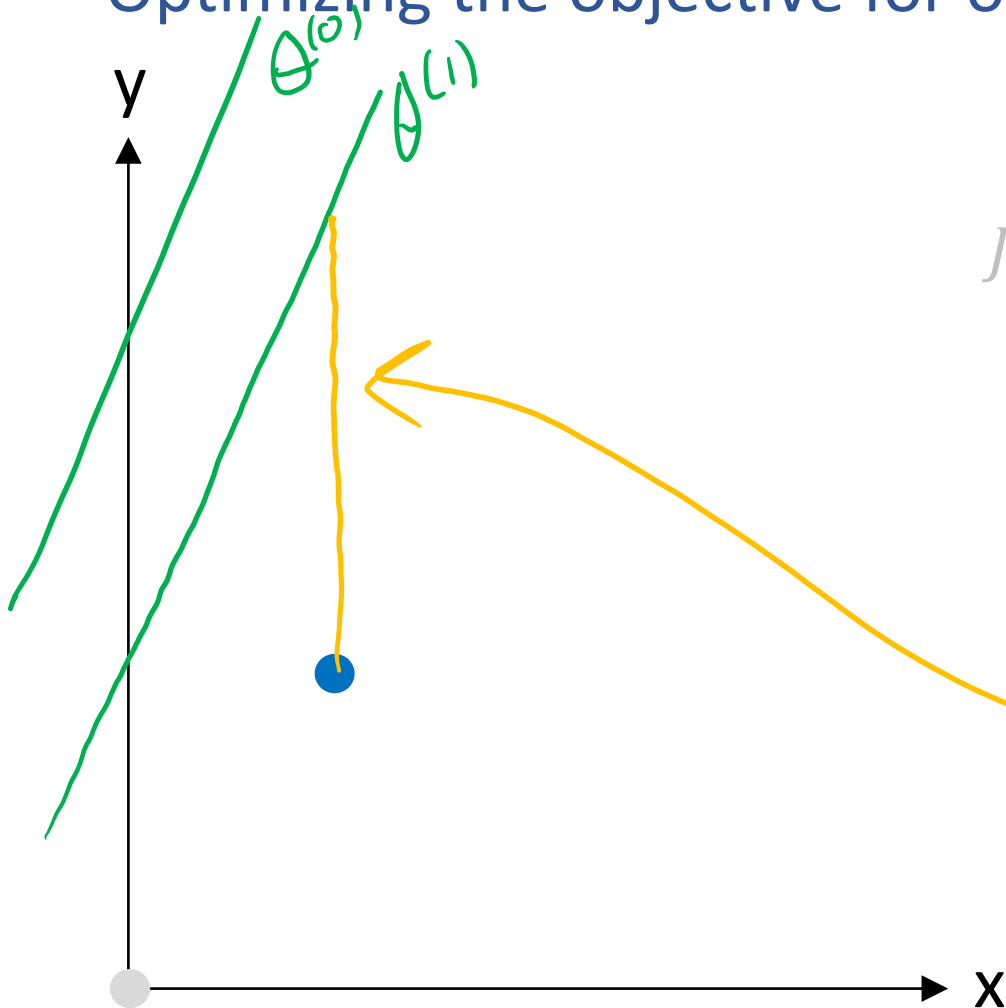
Stochastic Gradient Descent



$$J^{(1)}(w, b) = \left(\underline{y^{(1)} - (wx^{(1)} + b)} \right)^2$$

Stochastic Gradient Descent

Optimizing the objective for one point at a time

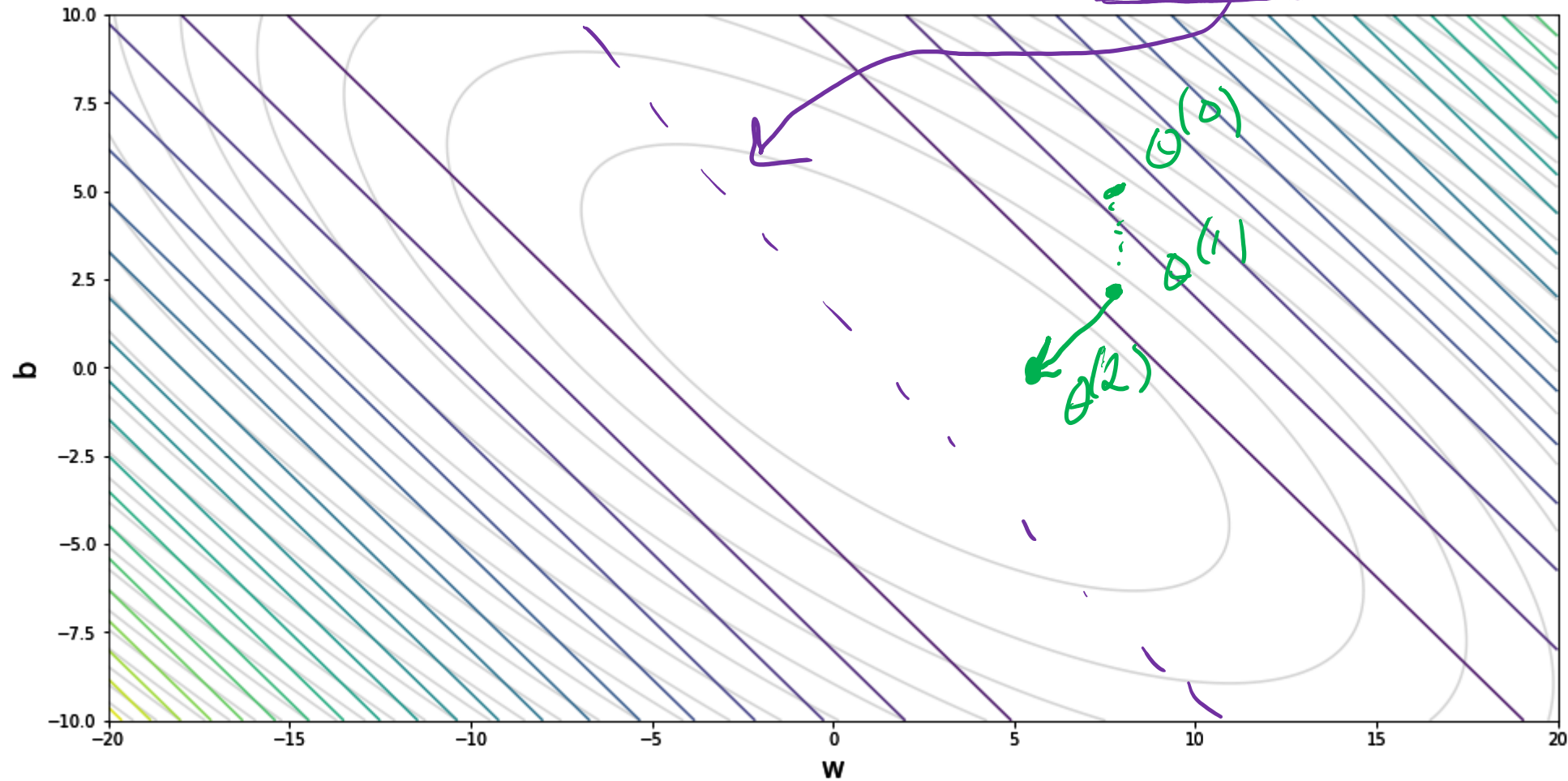


$$J(w, b) = \frac{1}{2} [J^{(1)}(w, b) + J^{(2)}(w, b)]$$

$$J^{(1)}(w, b) = \left(y^{(1)} - (wx^{(1)} + b) \right)^2$$

$$J^{(2)}(w, b) = \left(y^{(2)} - (wx^{(2)} + b) \right)^2$$

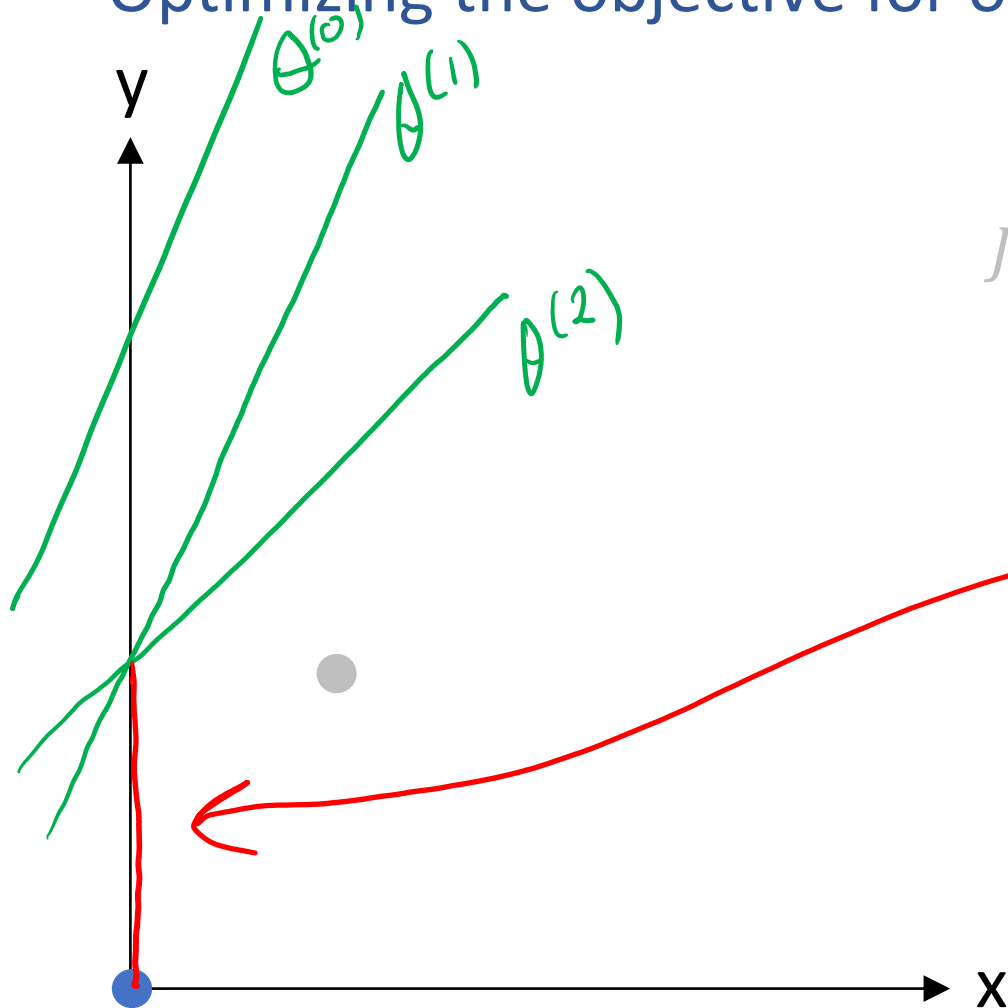
Stochastic Gradient Descent



$$J^{(2)}(w, b) = \left(\underline{y^{(2)} - (wx^{(2)} + b)} \right)^2$$

Stochastic Gradient Descent

Optimizing the objective for one point at a time



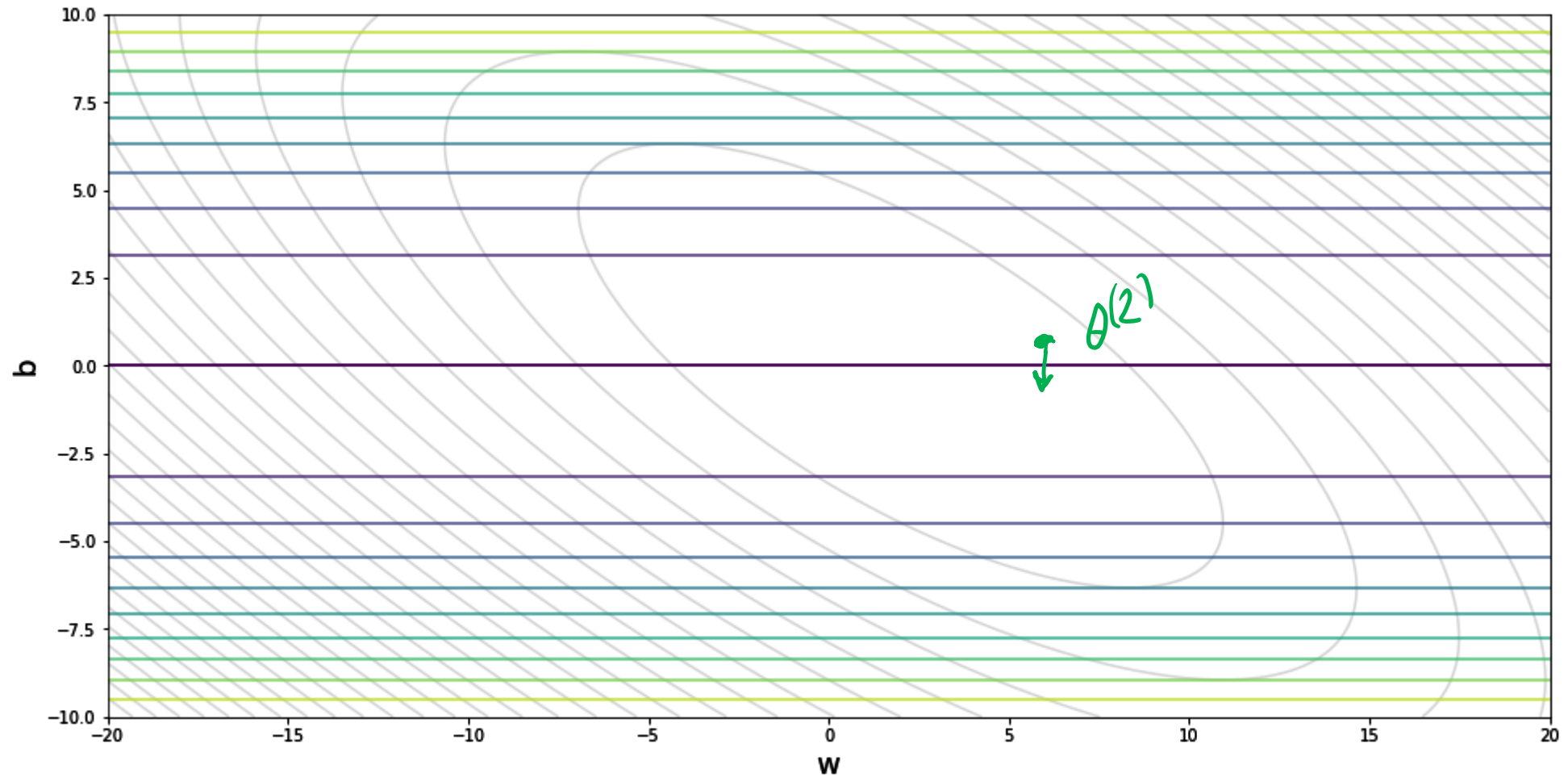
$$J(w, b) = \frac{1}{2} [J^{(1)}(w, b) + J^{(2)}(w, b)]$$

$$J^{(1)}(w, b) = \left(y^{(1)} - (wx^{(1)} + b) \right)^2$$

$$J^{(2)}(w, b) = \left(y^{(2)} - (wx^{(2)} + b) \right)^2$$

Back to optimize
point $i=1$

Stochastic Gradient Descent

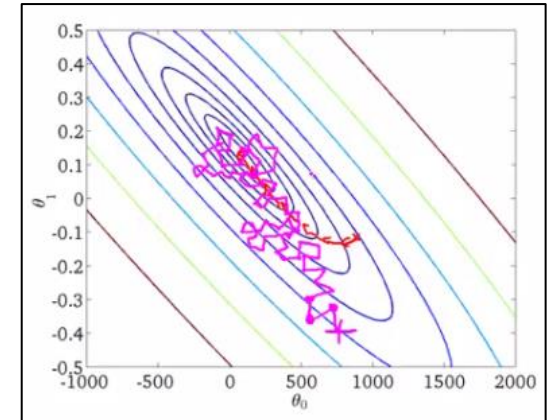


$$J^{(1)}(w, b) = \left(y^{(1)} - \underbrace{(wx^{(1)} + b)} \right)^2$$

Stochastic Gradient Descent (SGD)

Algorithm 2 Stochastic Gradient Descent (SGD)

```
1: procedure SGD( $\mathcal{D}, \theta^{(0)}$ )  
2:    $\theta \leftarrow \theta^{(0)}$   
3:   while not converged do  
4:      $i \sim \text{Uniform}(\{1, 2, \dots, N\})$   
5:      $\theta \leftarrow \theta - \gamma \nabla_{\theta} J^{(i)}(\theta)$   
6:   return  $\theta$ 
```



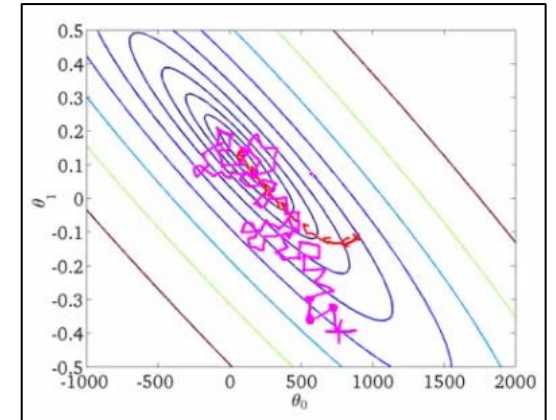
We need a per-example objective:

$$\text{Let } J(\theta) = \sum_{i=1}^N J^{(i)}(\theta)$$

Stochastic Gradient Descent (SGD)

Algorithm 2 Stochastic Gradient Descent (SGD)

```
1: procedure SGD( $\mathcal{D}, \theta^{(0)}$ )
2:    $\theta \leftarrow \theta^{(0)}$ 
3:   while not converged do
4:     for  $i \in \text{shuffle}(\{1, 2, \dots, N\})$  do
5:        $\theta \leftarrow \theta - \gamma \nabla_{\theta} J^{(i)}(\theta)$ 
6:   return  $\theta$ 
```



We need a per-example objective:

$$\text{Let } J(\theta) = \sum_{i=1}^N J^{(i)}(\theta)$$

In practice, it is common to implement SGD using sampling **without** replacement (i.e. $\text{shuffle}(\{1, 2, \dots, N\})$), even though most of the theory is for sampling **with** replacement (i.e. $\text{Uniform}(\{1, 2, \dots, N\})$).

Gradient Descent

Batch -> Mini-batch -> Stochastic

Basically, just a difference in how many training points are each time we compute a gradient (and then take a step with that gradient)

(Batch) GD



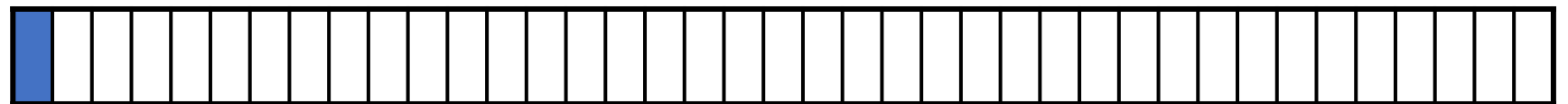
Effective batch size = N

Mini-batch SGD



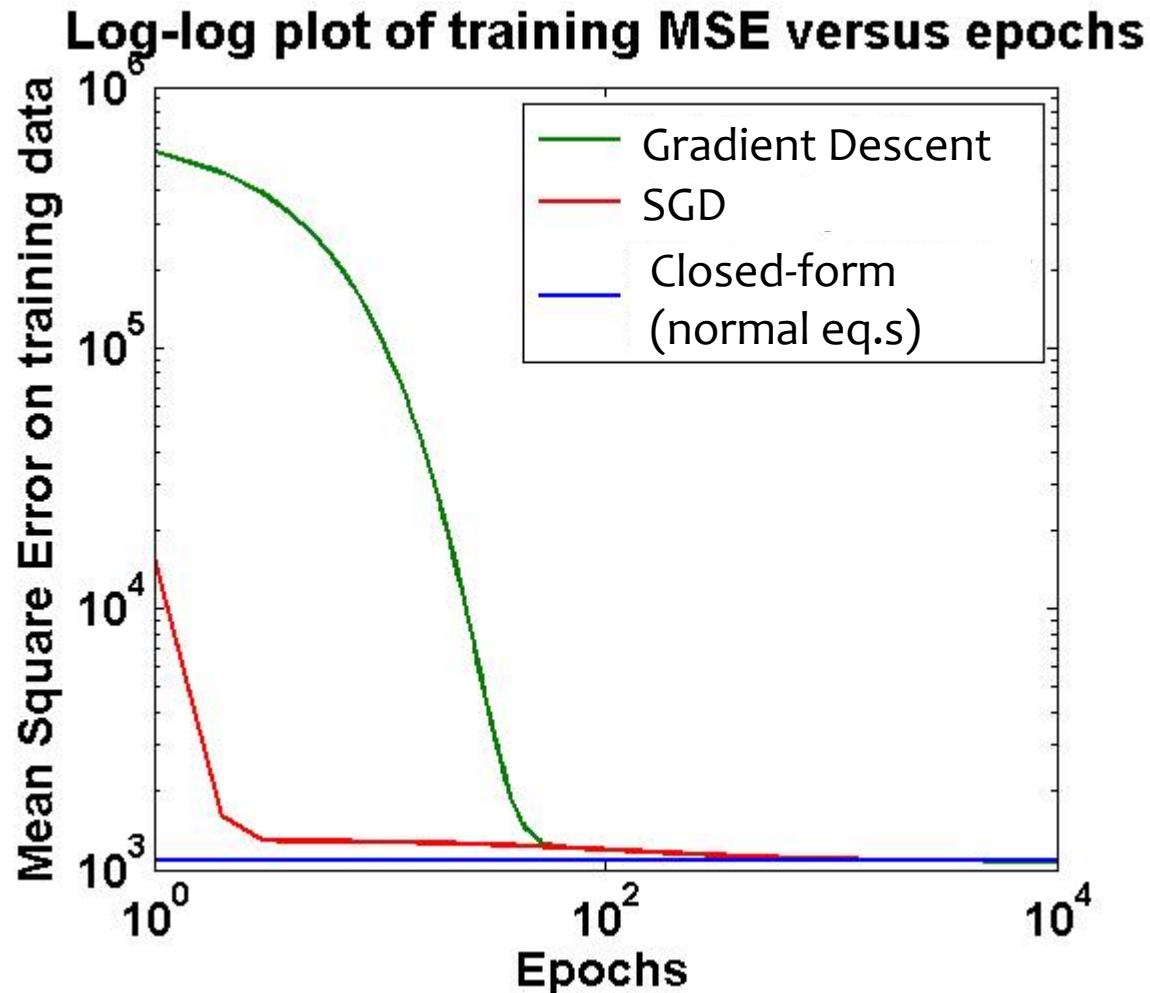
Batch size = B , e.g., $B = 10$

SGD



Effective batch size = 1

Convergence Curves



- Def: an **epoch** is a single pass through the training data

1. For GD, only **one update** per epoch
2. For SGD, **N updates** per epoch
 $N = (\# \text{ train examples})$

- SGD reduces MSE much more rapidly than GD
- For GD / SGD, training MSE is initially large due to uninformed initialization