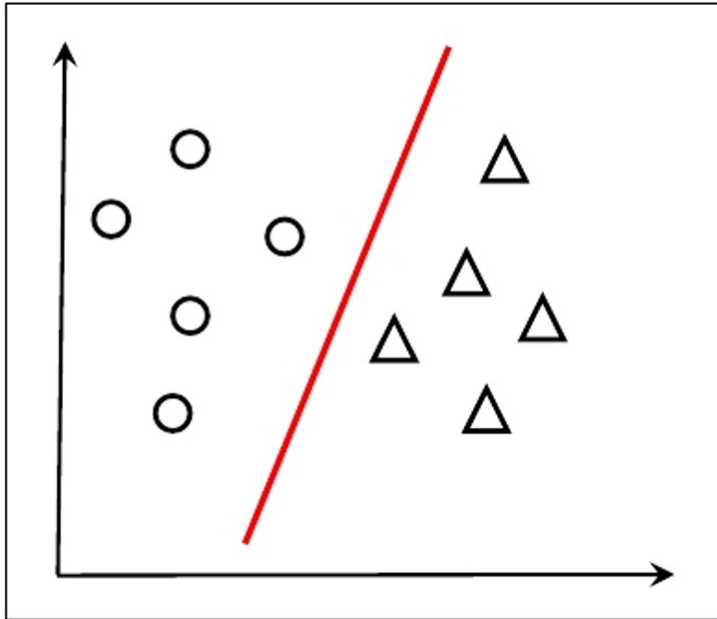


Linear Model

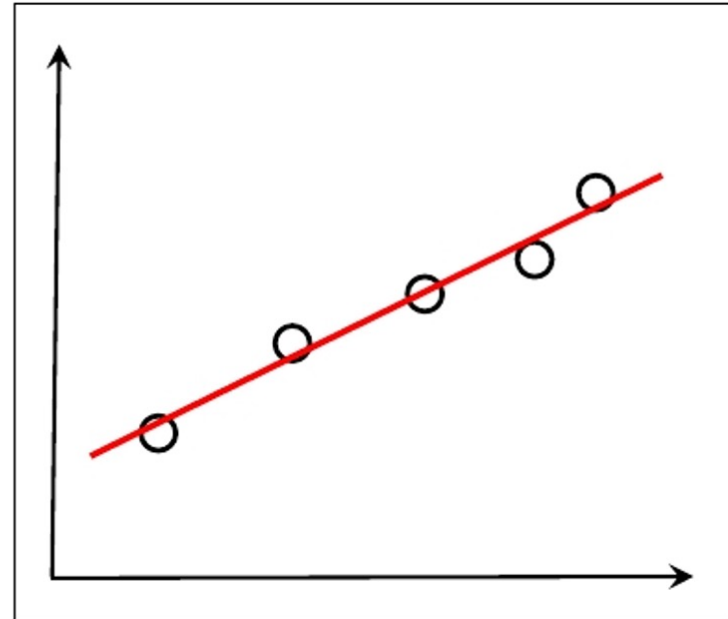
Alymzhan Toleu
alymzhan.toleu@gmail.com

Linear Model

Classification



Regression



A **linear model** specifies a linear relationship between a **dependent variable** ($Y/f(x)$) and d **independent variables** (X):

$$f(\mathbf{x}) = w_1x_1 + w_2x_2 + \dots + w_dx_d + b$$

Vectorization: $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$

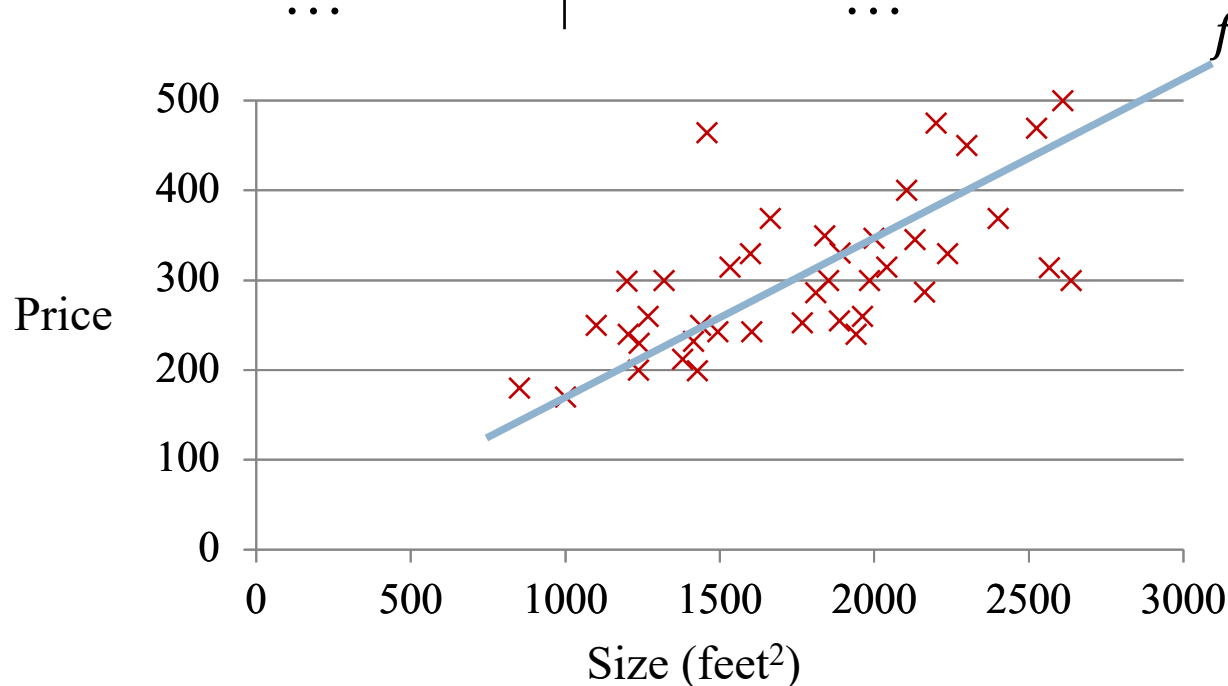
Linear Regression

Training set

Size in feet ² (x)	Price (\$) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

Supervised Learning

Given the “real value” for each example.



Regression Problem

Predict real-valued output

Linear Regression with One Variable

Size in feet ² (x)	Price (\$) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

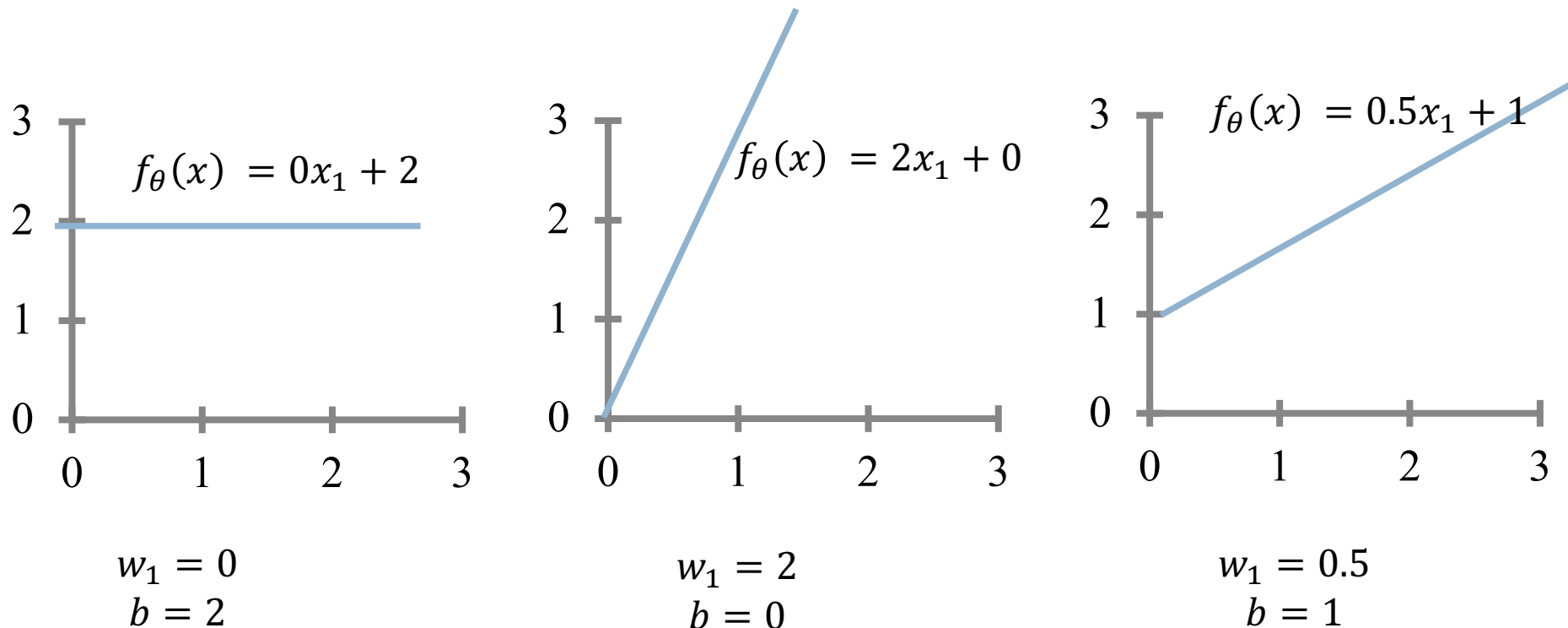
Hypothesis/Model: $y \approx f_{\theta}(x) = w_1 x_1 + b$

Parameters are: $\theta = \{w_1, b\}$

How to choose θ ?

Linear Regression: Example

Hypothesis/Model: $f_{\theta}(x) = w_1x_1 + b$ Parameters: $\theta = \{w_1, b\}$



Idea: Choose $\theta = \{w_1, b\}$ so that $f_{\theta}(x)$ is close to y for our training examples (x, y) .

Linear Regression: Example

Hypothesis/Model: $f_{\theta}(x) = w_1x_1 + b$

Parameters: $\theta = \{w_1, b\}$

Cost Function:

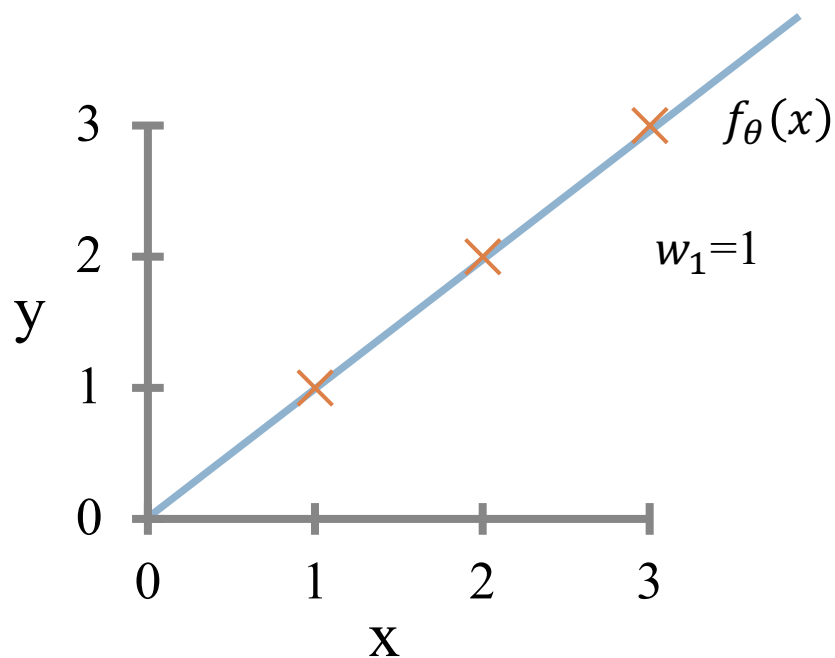
	Size in feet ² (x)	Price (\$) in 1000's (y)
m	2104	460
	1416	232
	1534	315
	852	178

$$E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$$

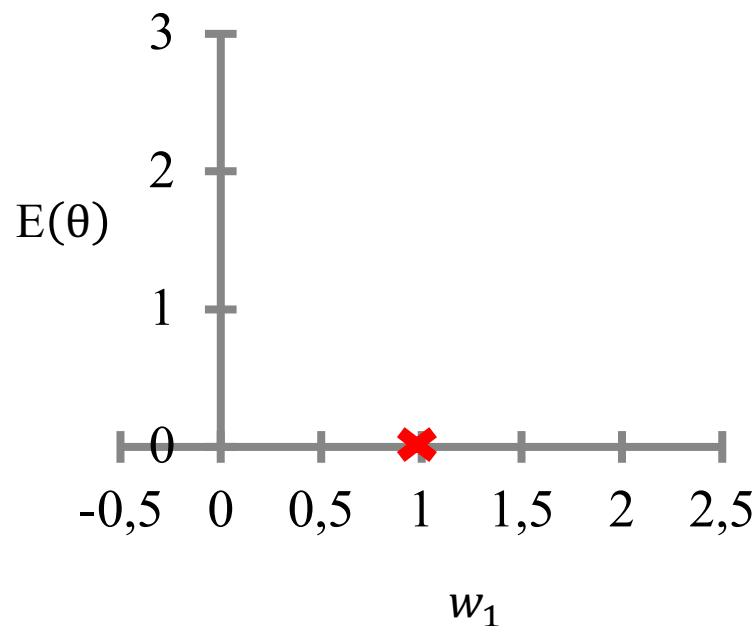
Goal: minimize $E(\theta)$

$f_{\theta}(x)$ Vs. $E(\theta)$

Simple hypothesis : $f_{\theta}(x) = w_1 x_1$
(ignoring parameter b)



$$E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$$

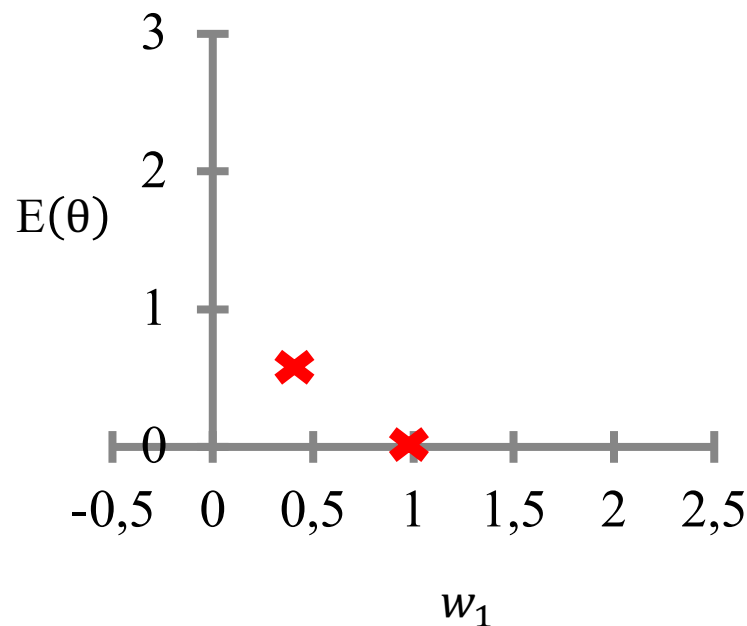
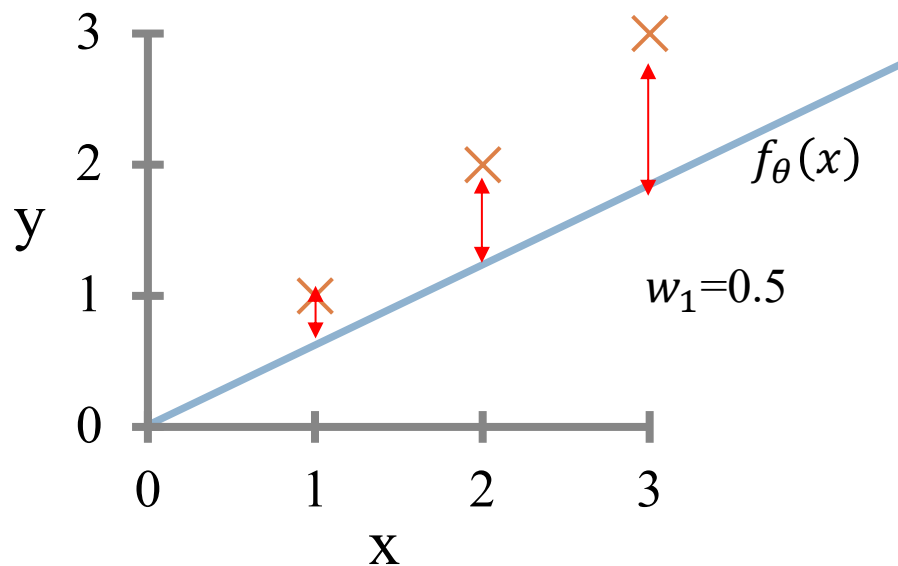


$$E(w_1=1) = \frac{1}{2m} [(1 - 1)^2 + (2 - 2)^2 + (3 - 3)^2] = 0$$

$f_{\theta}(x)$ Vs. $E(\theta)$

Simple hypothesis : $f_{\theta}(x) = w_1 x_1$
(ignoring parameter b)

$$E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$$

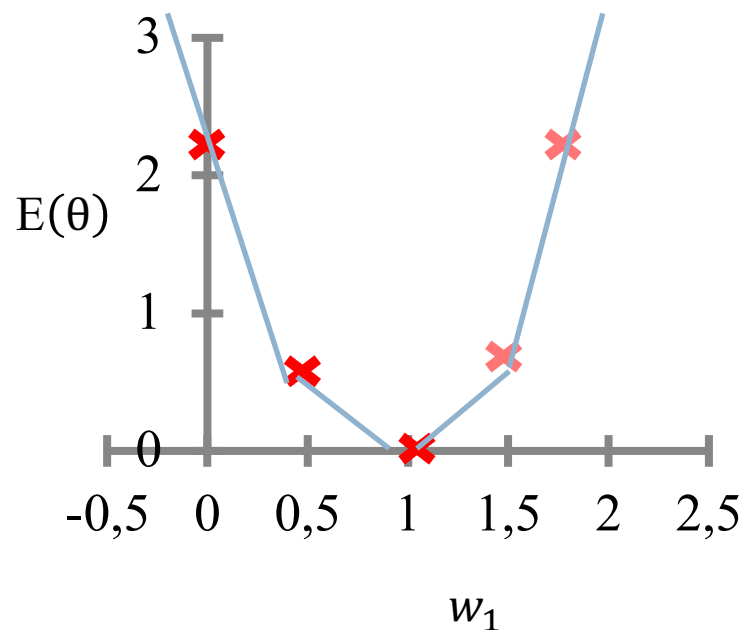
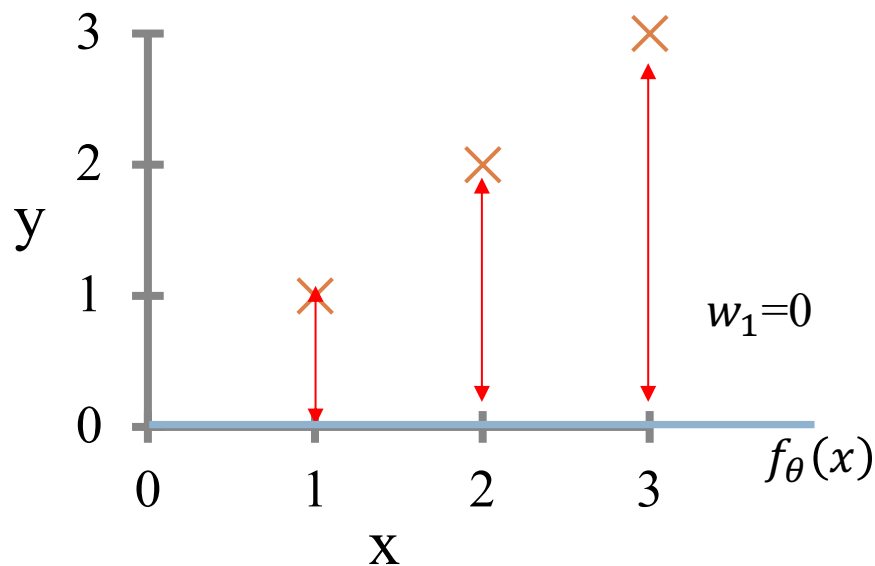


$$E(w_1=0.5) = \frac{1}{2m} [(0.5 - 1)^2 + (1 - 2)^2 + (1.5 - 3)^2] = 0.548$$

$f_{\theta}(x)$ Vs. $E(\theta)$

Simple hypothesis : $f_{\theta}(x) = w_1 x_1$
(ignoring parameter b)

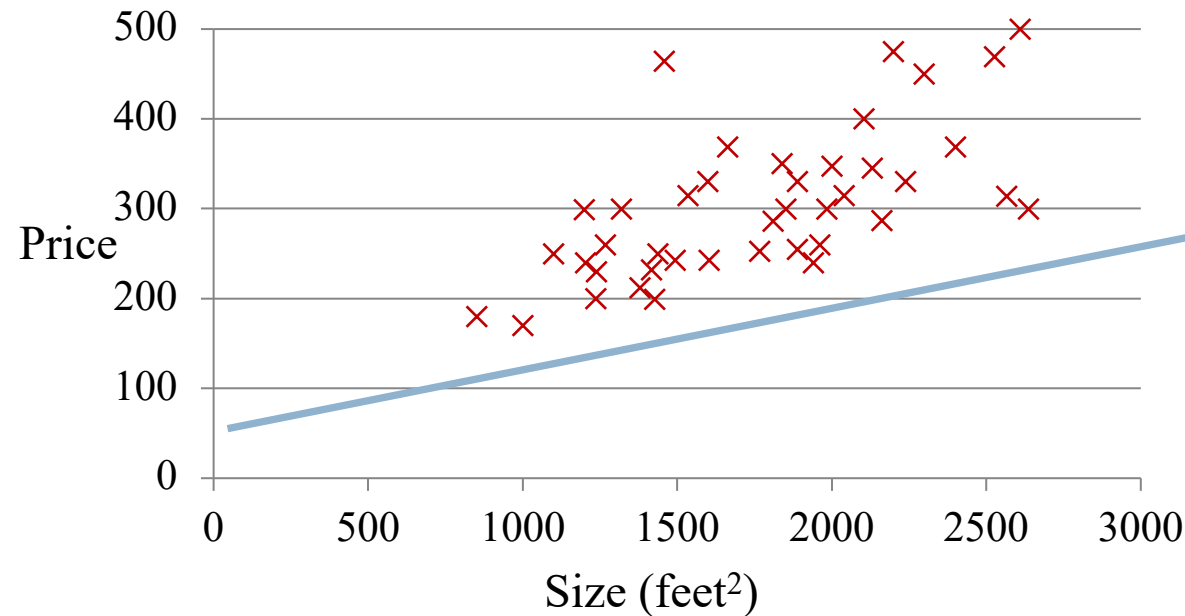
$$E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$$



$$E(w_1=0) = \frac{1}{2m} [(0-1)^2 + (0-2)^2 + (0-3)^2] \approx 2.3$$

Choose parameter θ that minimizes cost function $E(\theta)$, which corresponds to finding a straight line that fits the data well.

Example: House price prediction



Hypothesis: $f_{\theta}(x) = 0.05x_1 + 50$

Parameters: $\theta = \{w_1, b\}$

Cost Function: $E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$

Goal: minimize $E(\theta)$

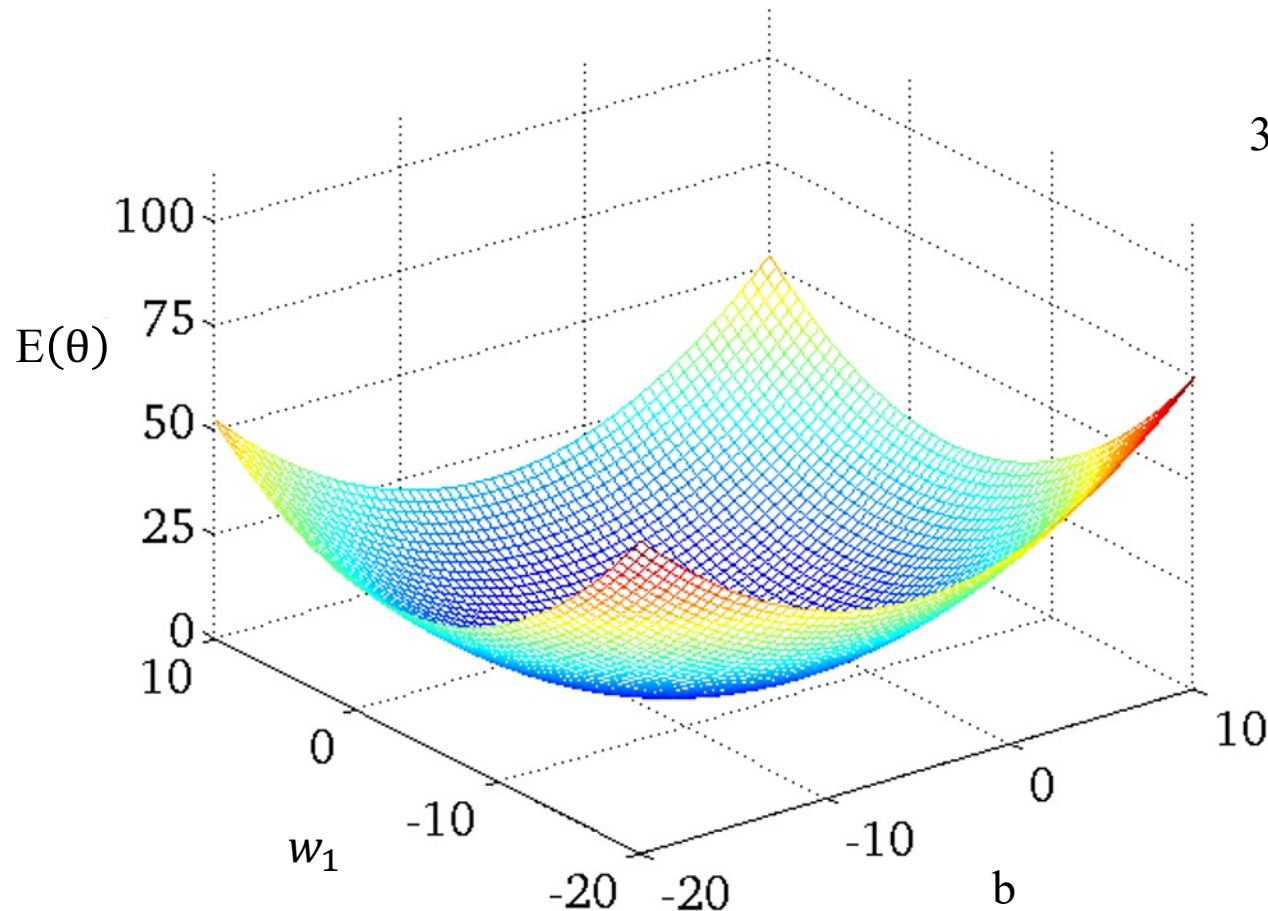
How to plot the $E(\theta)$?

Hypothesis: $f_{\theta}(x) = w_1x_1 + b$

Parameters: $\theta = \{w_1, b\}$

Cost Function: $E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$

Goal: minimize $E(\theta)$

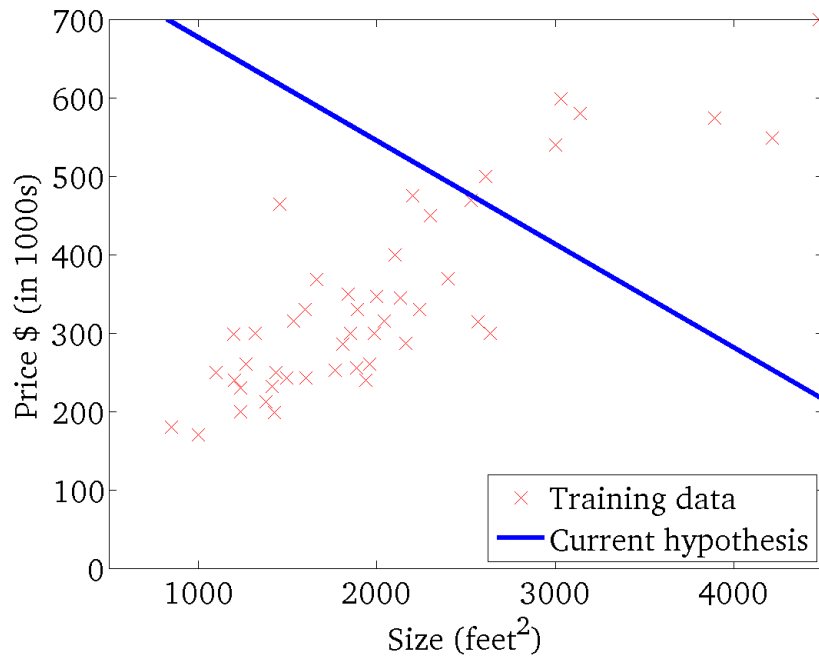


3-d surface plot

Contour figure

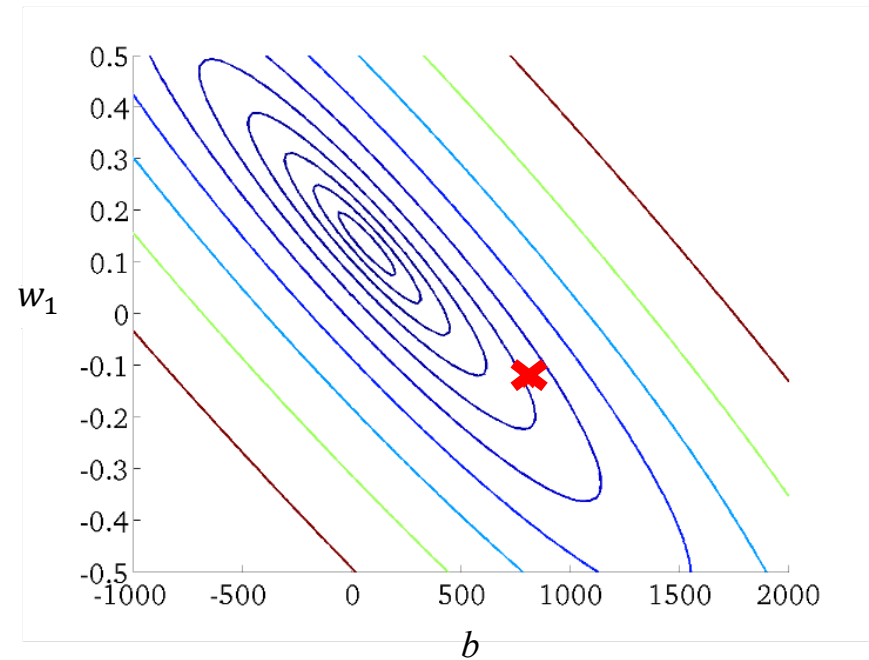
$$f_{\theta}(x) = w_1 x_1 + b$$

(for fixed $\theta = \{w_1, b\}$, this is a function of x)



$$E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$$

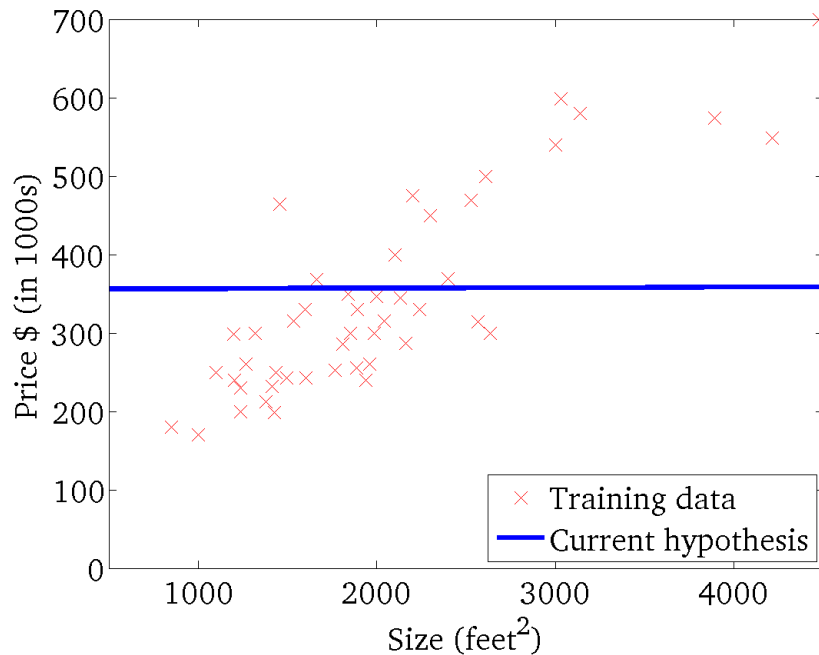
(function of the parameters $\theta = \{w_1, b\}$)



Contour figure

$$f_{\theta}(x) = w_1 x_1 + b$$

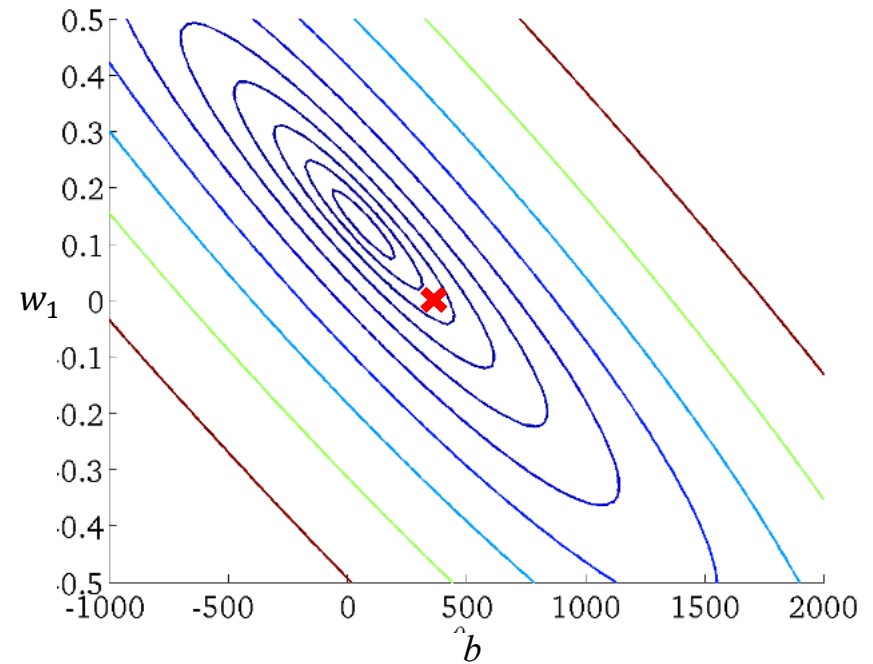
(for fixed $\theta = \{w_1, b\}$, this is a function of x)



$$f_{\theta}(x) = 0x_1 + 370$$

$$E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$$

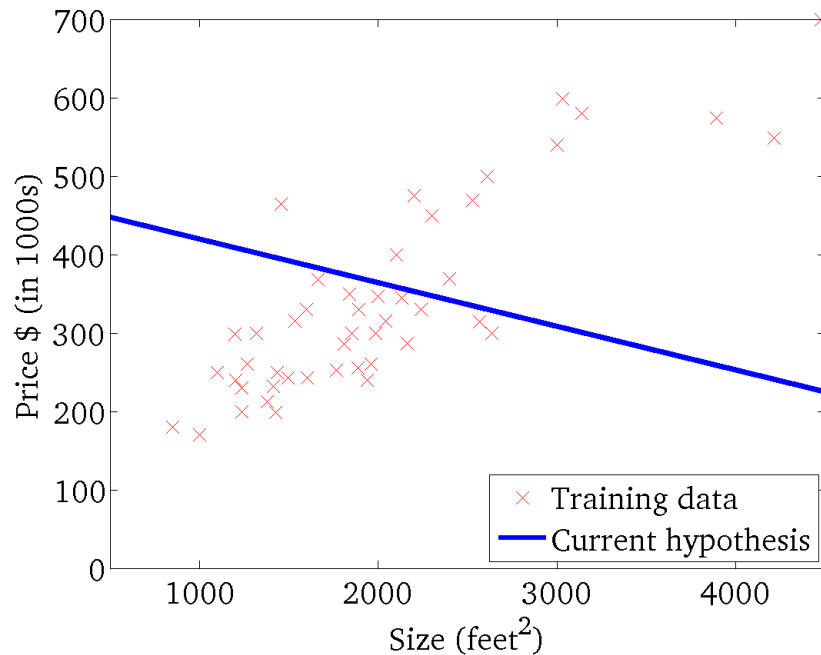
(function of the parameters $\theta = \{w_1, b\}$)



Contour figure

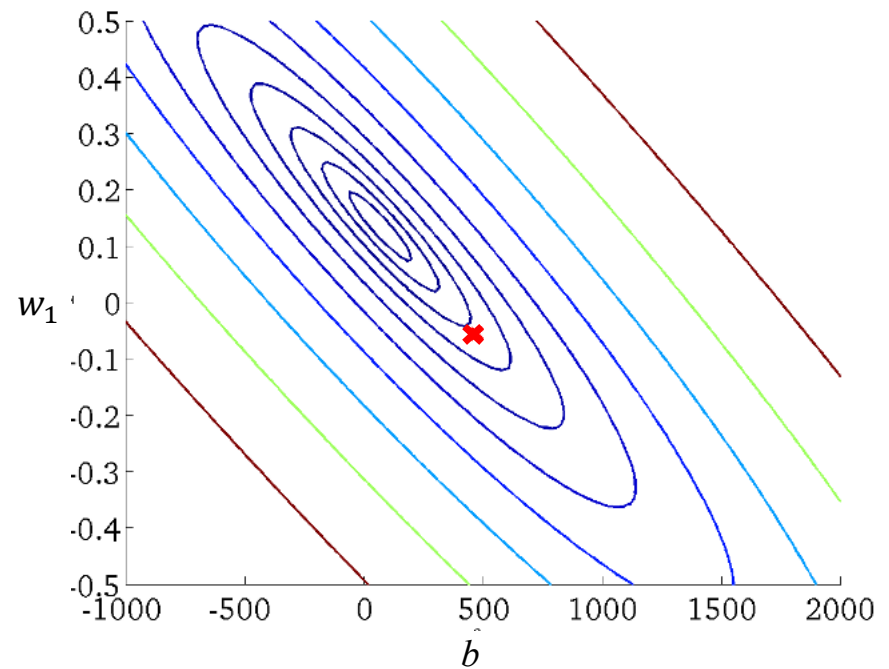
$$f_{\theta}(x) = w_1 x_1 + b$$

(for fixed $\theta = \{w_1, b\}$, this is a function of x)



$$E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$$

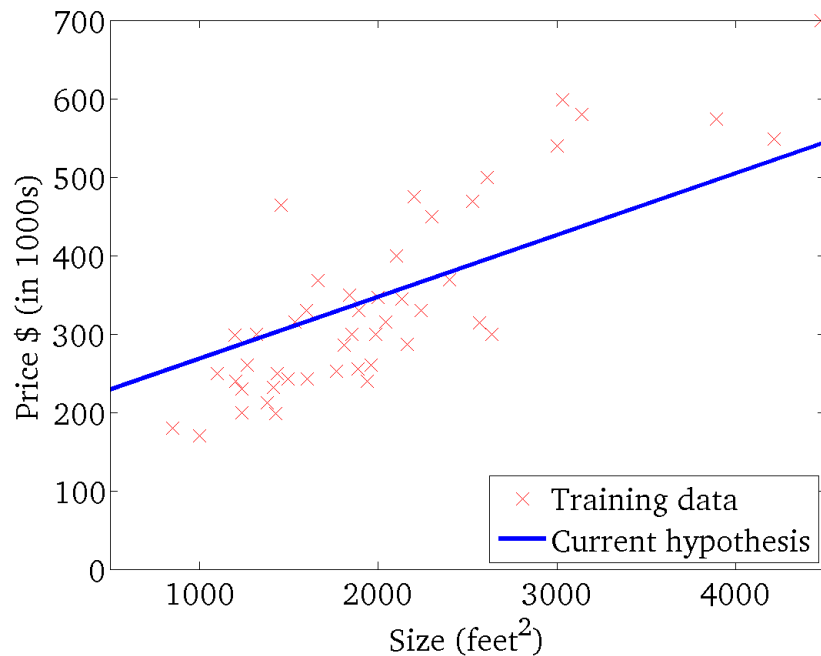
(function of the parameters $\theta = \{w_1, b\}$)



Contour figure

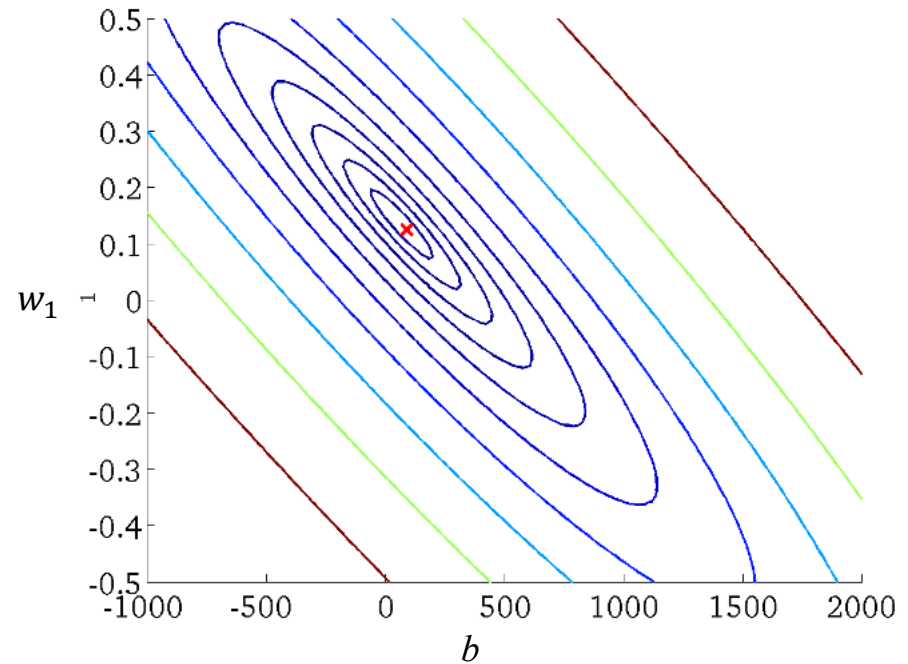
$$f_{\theta}(x) = w_1 x_1 + b$$

(for fixed $\theta = \{w_1, b\}$, this is a function of x)



$$E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$$

(function of the parameters $\theta = \{w_1, b\}$)



Gradient descent

Hypothesis: $f_{\theta}(x) = w_1x_1 + b$

Parameters: $\theta = \{w_1, b\}$

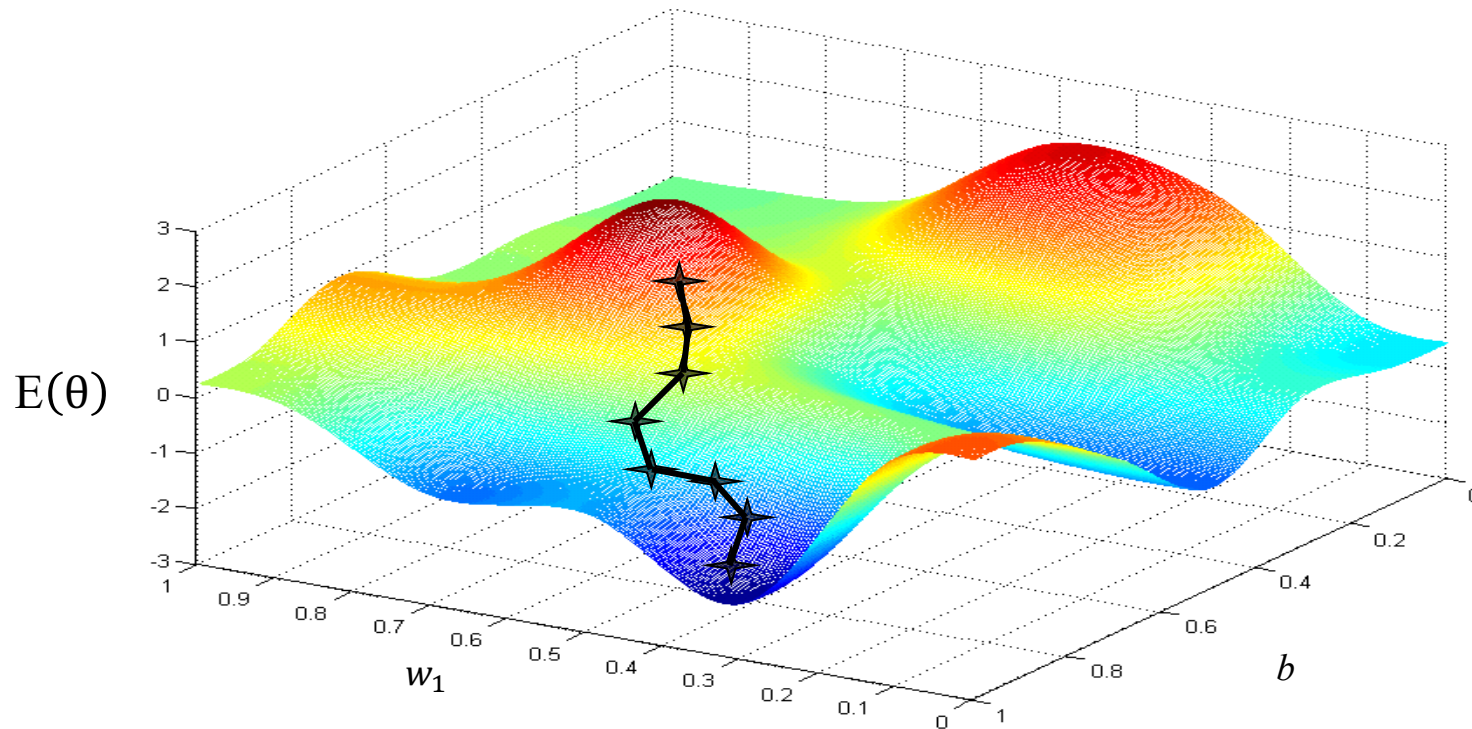
Cost Function: $E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$

Goal: minimize $E(\theta)$

Steps:

- Start with some $\theta = \{w_1, b\}$; *e.g.* $w_1 = 0, b = 0$.
- Keep changing θ to reduce $E(\theta)$ until end up at a minimum.

Intuition picture of gradient descent

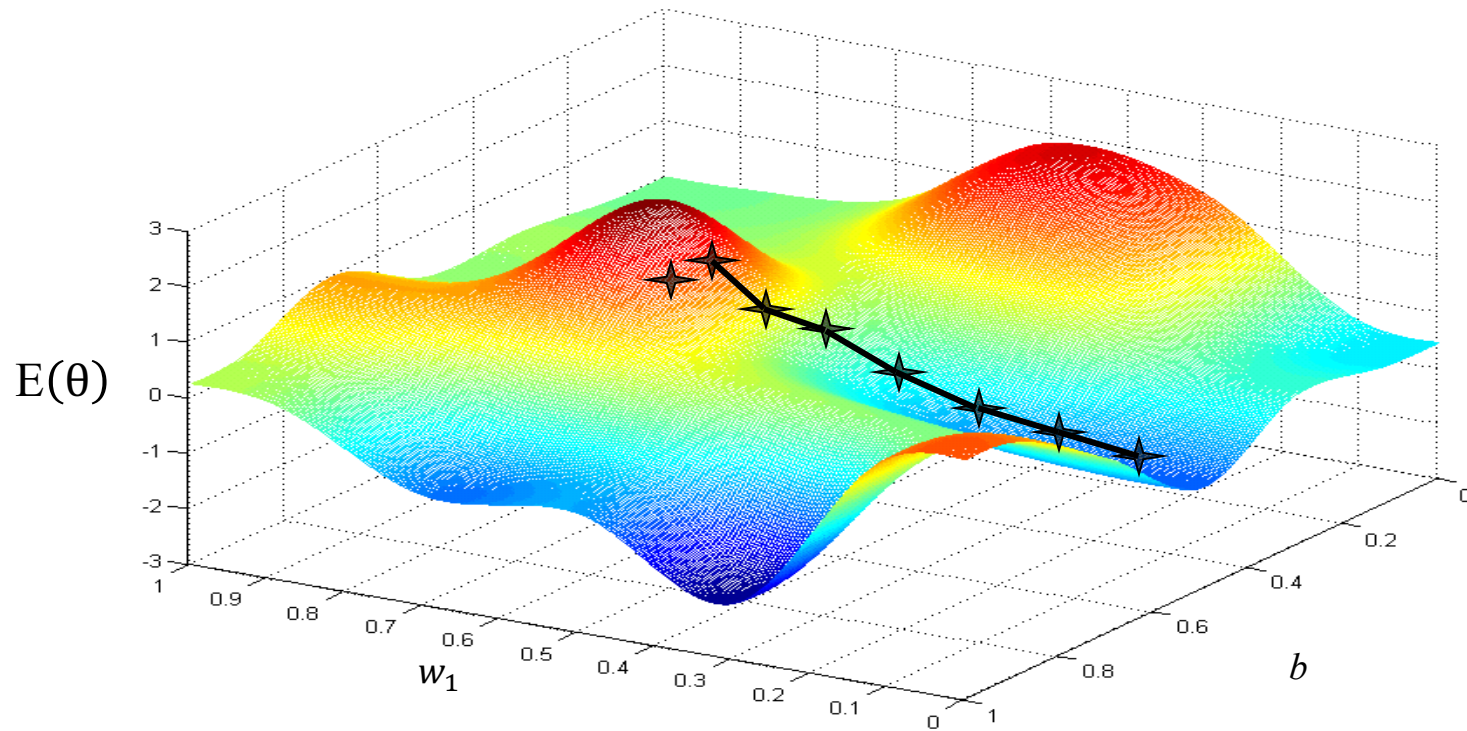


Starting at some points on the surface of this function.

Take a step in the direction of steepest descent.

Each step changes parameter θ to reduce $E(\theta)$ until end up at a local minimum.

Intuition picture of gradient descent

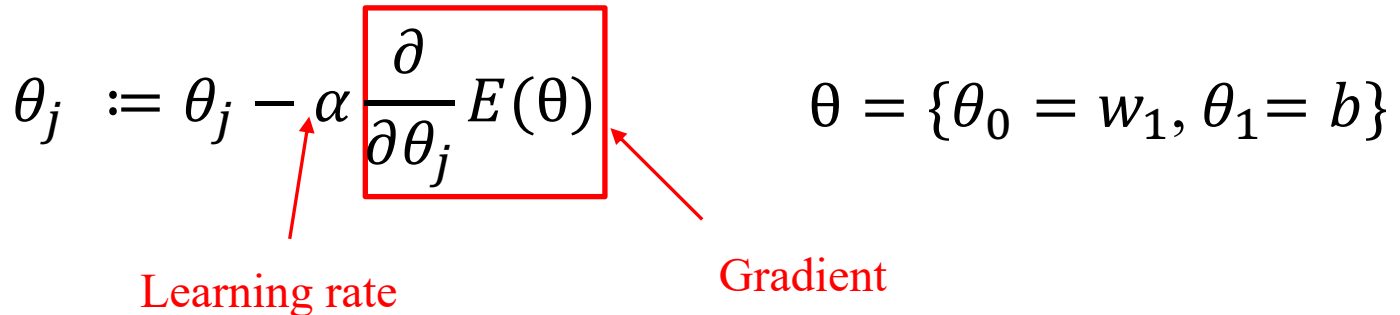


Starting at **another point** on the surface of this function.
Each step changes parameter θ to reduce $E(\theta)$ until end up at a local minimum.
There are many local minimums.

Gradient descent algorithm

- Repeat until convergence

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} E(\theta)$$



Learning rate Gradient

- Simultaneous update
- Learning rate
 - α determines the step size at each iteration while moving toward a minimum of a cost function.
 - If α is too **small**, gradient descent can be slow;
 - If α is too **large**, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.
- Gradient descent can converge to **a local minimum**, even with the learning rate α fixed.

Gradient descent for linear regression

Hypothesis:

$$f(x) = wx_i + b \quad f(x_i) \simeq y_i$$

Cost Function:

$$(w^*, b^*) = \arg \min_{(w, b)} \sum_{i=1}^m (f(x_i) - y_i)^2$$
$$= \arg \min_{(w, b)} \sum_{i=1}^m (y_i - wx_i - b)^2$$

Minimize:

$$E(\theta) = E_{(w, b)} = \sum_{i=1}^m (y_i - wx_i - b)^2$$

Derivates

Finding partial derivates of function $E_{(w,b)}$ with two variables:
 w and b

$$\begin{aligned}\frac{\partial E_{(w,b)}}{\partial w} &= \frac{\partial}{\partial w} \left[\sum_{i=1}^m (y_i - wx_i - b)^2 \right] \\ &= \sum_{i=1}^m \frac{\partial}{\partial w} [(y_i - wx_i - b)^2] \\ &= \sum_{i=1}^m [2 \cdot (y_i - wx_i - b) \cdot (-x_i)] \\ &= \sum_{i=1}^m [2 \cdot (wx_i^2 - y_i x_i + bx_i)] \\ &= 2 \cdot \left(w \sum_{i=1}^m x_i^2 - \sum_{i=1}^m y_i x_i + b \sum_{i=1}^m x_i \right) \\ &= 2 \left(w \sum_{i=1}^m x_i^2 - \sum_{i=1}^m (y_i - b) x_i \right)\end{aligned}$$

$$\begin{aligned}\frac{\partial E_{(w,b)}}{\partial b} &= \frac{\partial}{\partial b} \left[\sum_{i=1}^m (y_i - wx_i - b)^2 \right] \\ &= \sum_{i=1}^m \frac{\partial}{\partial b} [(y_i - wx_i - b)^2] \\ &= \sum_{i=1}^m [2 \cdot (y_i - wx_i - b) \cdot (-1)] \\ &= \sum_{i=1}^m [2 \cdot (b - y_i + wx_i)] \\ &= 2 \cdot \left[\sum_{i=1}^m b - \sum_{i=1}^m y_i + \sum_{i=1}^m wx_i \right] \\ &= 2 \left(mb - \sum_{i=1}^m (y_i - wx_i) \right)\end{aligned}$$

Gradient descent for linear regression

Hypothesis: $f_{\theta}(x) = w_1 x_1 + b$

Parameters: $\theta = \{w_1, b\}$


Cost Function: $E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$

Goal: minimize $E(\theta)$

- Repeat until convergence

$$w_1 := w_1 - \alpha \frac{1}{m} \sum_{i=1}^m [f_{\theta}(x^{(i)}) - y^{(i)}] x^{(i)}$$

$$b := b - \alpha \frac{1}{m} \sum_{i=1}^m [f_{\theta}(x^{(i)}) - y^{(i)}]$$

$$\frac{\partial}{\partial \theta_j} E(\theta)$$


update w_1 and b
simultaneously

Linear Regression: Example

Size in feet ² (x)	Price (\$) in 1000's (y)
2104	460
1416	232
1534	315
852	178

starting point(can be random):

$$w = 0, \quad b = 0$$

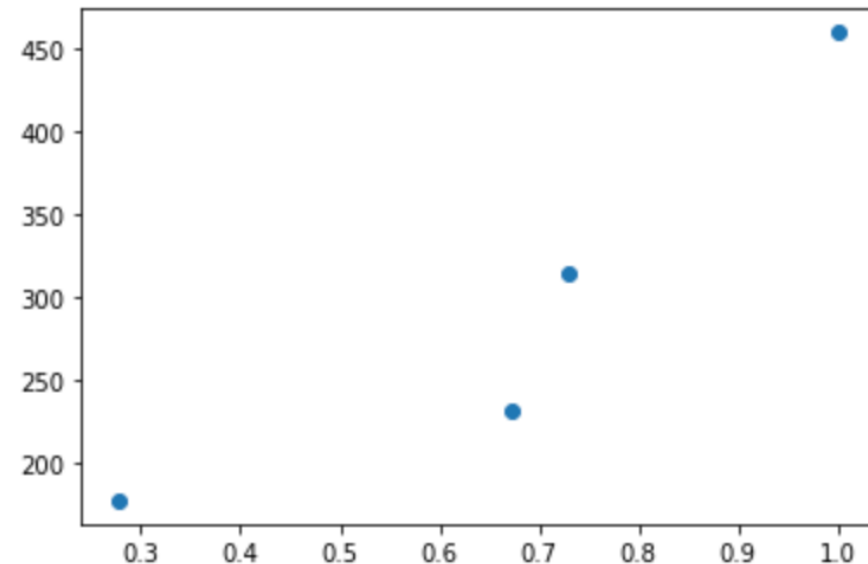
learning rate $a = 0.5$

max_inter = 10



Normalizing & add a column with value one

```
array([[1.,          1.,          ],
       [1.,          0.6730038 ],
       [1.,          0.72908745],
       [1.,          0.27661597]])
```



Iteration 1

Estimation: ($w = 0$, $b = 0$)

$$f_{\theta}(1) = (0 * 1 + 0) = 0$$

$$f_{\theta}(0.67) = (0 * 0.67 + 0) = 0$$

$$f_{\theta}(0.72) = (0 * 0.72 + 0) = 0$$

$$f_{\theta}(0.27) = (0 * 0.27 + 0) = 0$$

Hypothesis: $f_{\theta}(x) = w_1 x_1 + b$

Parameters: $\theta = \{w_1, b\}$

Cost Function: $E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$

Normalized input x

```
array([[1., 1., 1.],  
       [1., 0.6730038],  
       [1., 0.72908745],  
       [1., 0.27661597]])
```

Cost: $E(\theta) = \frac{1}{2*4} [(0 - 460)^2 + (0 - 232)^2 + (0 - 315)^2 + (0 - 178)^2] \approx 49541.62$

Gradient:

$$w' = \frac{\partial E(\theta)}{\partial w} = \frac{1}{4} [(0 - 460) * 1 + (0 - 232) * 0.67 + (0 - 315) * 0.72 + (0 - 178) * 0.27] \approx -222.575$$

$$b' = \frac{\partial E(\theta)}{\partial b} = \left(\frac{1}{4}\right) [(0 - 460) + (0 - 232) + (0 - 315) + (0 - 178)] \approx -296.25$$

Update:

$$w_1 := w_1 - \alpha * w' = 0 - 0.5 * (-222.575) \approx 111.2875$$

$$b := b - \alpha * b' = 0 - 0.5 * (-296.25) \approx 148.125$$

New
parameters



Iteration 2

Estimation: ($w = 111.28$, $b = 148.12$)

Hypothesis: $f_{\theta}(x) = w_1 x_1 + b$

Parameters: $\theta = \{w_1, b\}$

Cost Function: $E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$

$$f_{\theta}(1) = (111.28 * 1 + 148.12) \approx 260$$

Normalized input x

$$f_{\theta}(0.67) = (111.28 * 0.67 + 148.12) \approx 223$$

$$f_{\theta}(0.72) = (111.28 * 0.72 + 148.12) \approx 229$$

$$f_{\theta}(0.27) = (111.28 * 0.27 + 148.12) \approx 179$$

```
array([[1., 1.],
       [1., 0.6730038],
       [1., 0.72908745],
       [1., 0.27661597]])
```

Cost: $E(\theta) = (\frac{1}{2*4})[(260 - 460)^2 + (223 - 232)^2 + (229 - 315)^2 + (179 - 178)^2] \approx 5918.73$

Gradient:

$$w' = \frac{\partial E(\theta)}{\partial w} = (\frac{1}{4}) [(260 - 460) * 1 + (223 - 232) * 0.67 + (229 - 315) * 0.72 + (179 - 178) * 0.27] \approx -66.92$$

$$b' = \frac{\partial E(\theta)}{\partial b} = (\frac{1}{4}) [(260 - 460) + (223 - 232) + (229 - 315) + (179 - 178)] \approx -73.5$$

Update:

$$w_1 := w_1 - \alpha * w' = 0 - 0.5 * (-66.92) \approx 184.72$$

$$b := b - \alpha * b' = 0 - 0.5 * (-73.5) \approx 145.33$$



New
parameters

Continue...

Iteration 3

```
iter: 3 cost: 2778.85472457512  
prediction: [330.06397483 282.53897763 290.69006727 224.92873394]  
gradient: [-14.19456158 -25.16648269]  
parameters: [191.8231828 157.92131416]
```

Iteration 4

```
iter: 4 cost: 2487.2205920097786  
prediction: [349.74449696 298.10482769 306.96163143 235.50674024]  
gradient: [ 1.32942408 -13.9300234 ]  
parameters: [191.15847076 164.88632586]
```

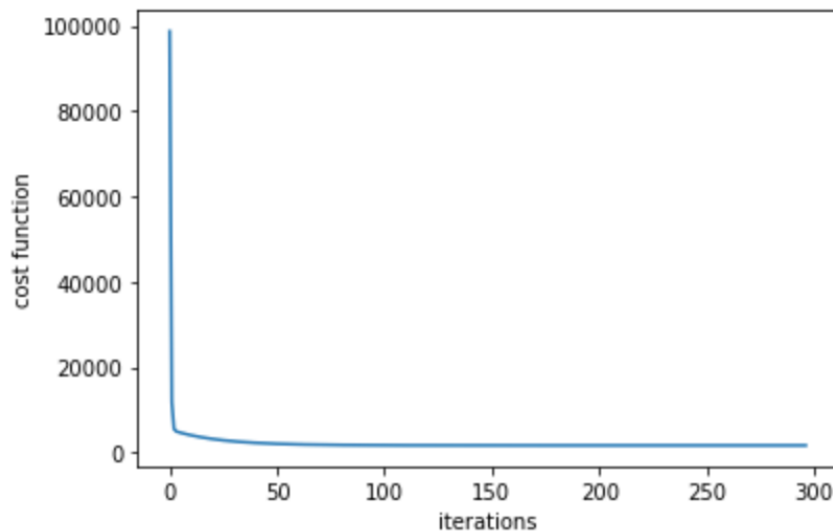
Iteration 5

```
iter: 5 cost: 2398.9324937073943  
prediction: [356.04479662 302.12759501 311.37502203 236.76866166]  
gradient: [ 5.32901883 -10.78641023]  
parameters: [188.49396135 170.27953098]
```

....

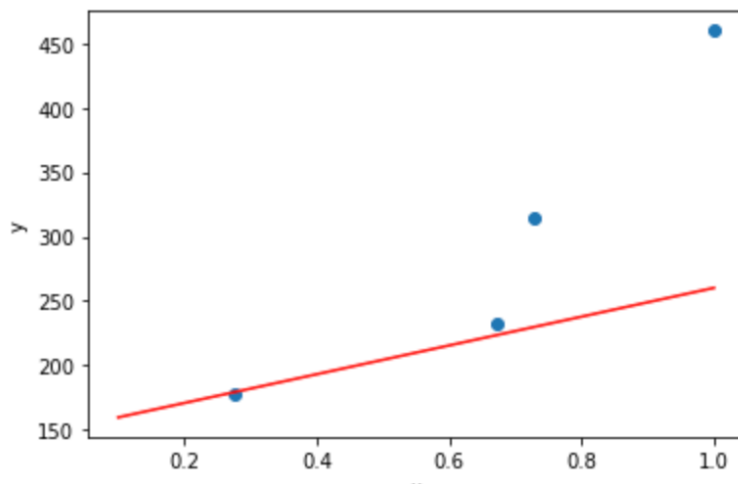
Until it converges

Linear Regression: Example

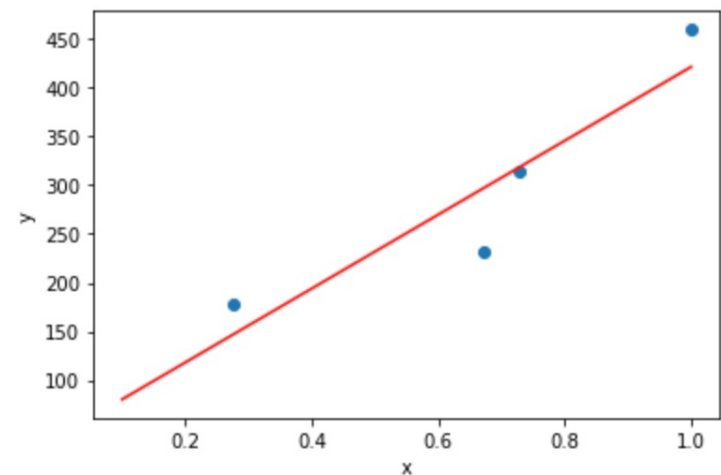


learning_rate = 0.5

iter: 0 cost: 98663.42912068815

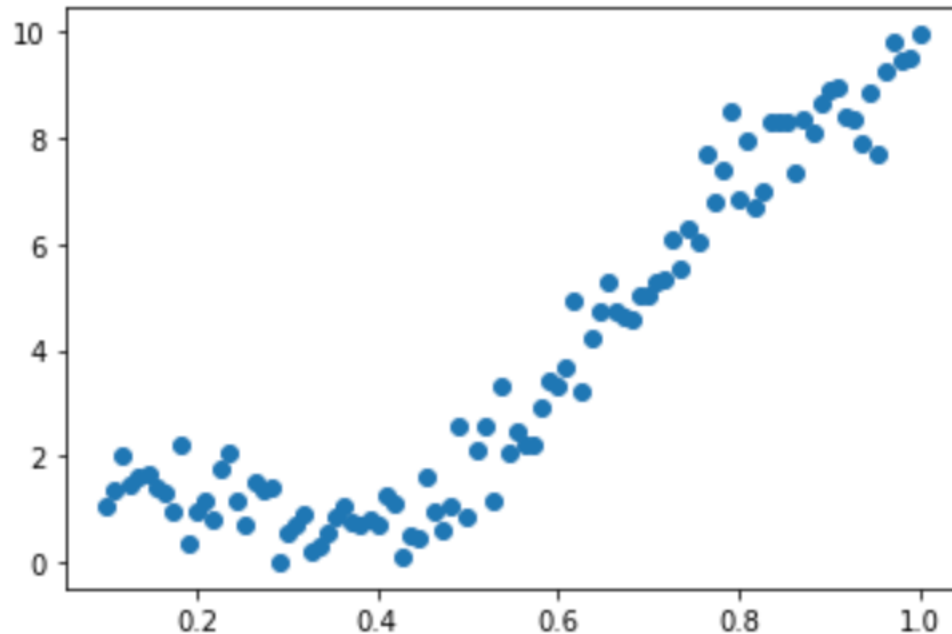


iter: 296 cost: 1684.0276765926678



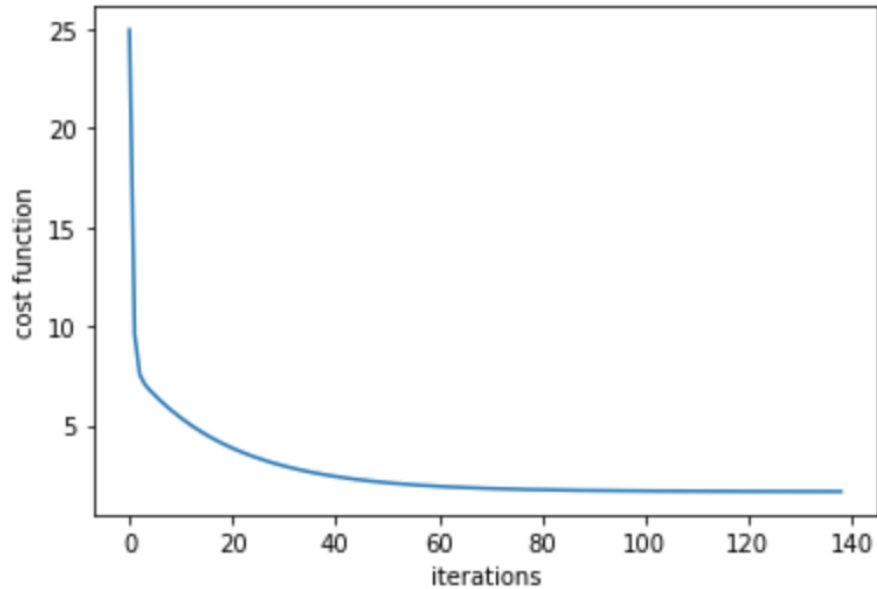
Converged.

Linear Regression: Example-2

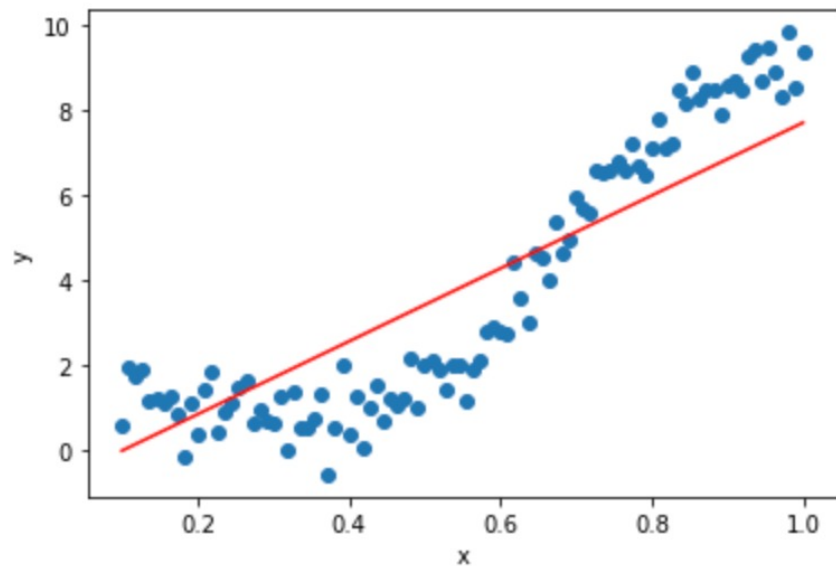


$N = 100$ samples

learning_rate = 0.5

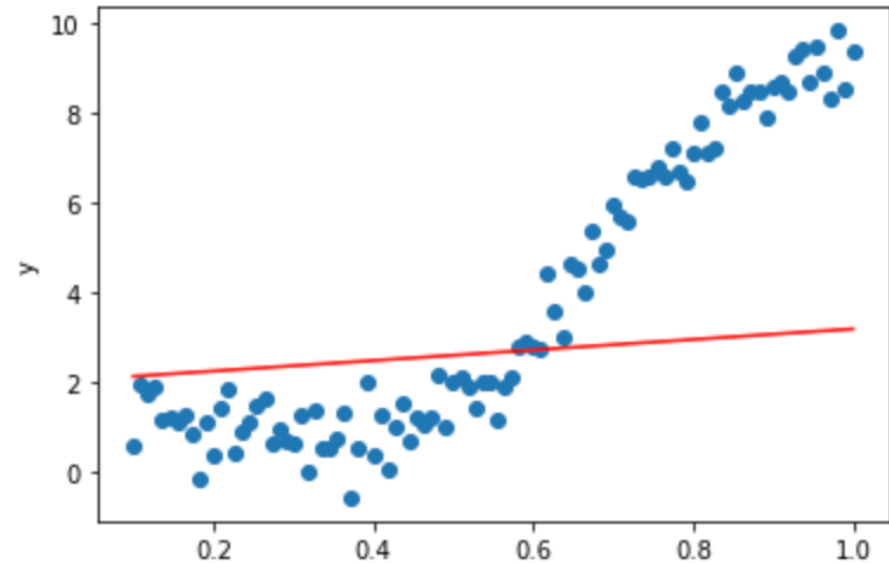


iter: 50 cost: 2.0982659858775956

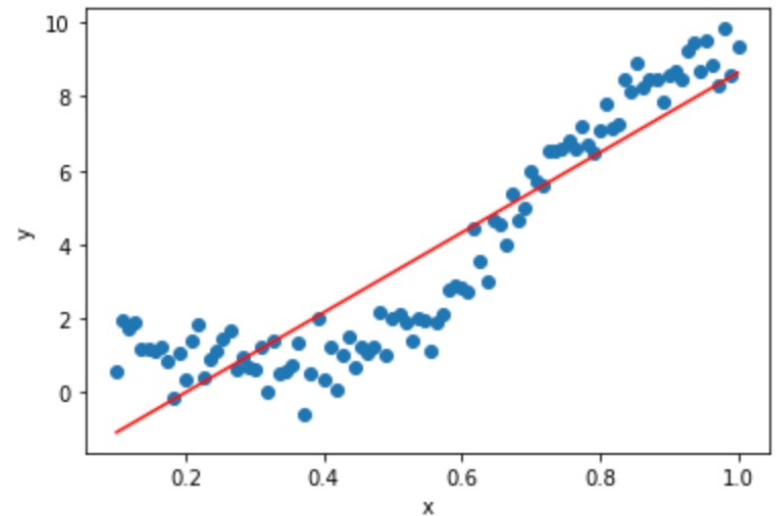


Alymzhan Toleu

iter: 0 cost: 24.973176452747104



iter: 138 cost: 1.6546338133174023



Converged.

Multivariate linear regression

Size in feet ² (x)	Price (\$) in 1000's (y)
2104	460
1416	232
1534	315
852	178

$$f_{\theta}(x) = w_1 x_1 + b$$

linear regression with one variable

Size (feet ²)	Number of bedrooms	Number of floors	Age of home (years)	Price (\$1000)
x_1	x_2	x_3	x_4	y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...

linear regression with multiple variables

n – is number of features

$$f_{\theta}(x) = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + \dots + w_n x_n + b$$

$$\textcolor{red}{\lceil} \quad x_0 = 1$$

Hypothesis: $f_{\theta}(x) = \theta^T x = w_0 x_0 + w_1 x_1 + \dots + w_n x_n$

Parameters: $\theta = \{w_0, \dots, w_n\}$

Cost Function: $E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$

Goal: minimize $E(\theta)$

- Repeat until convergence

$$w_0 := w_0 - \alpha \frac{1}{m} \sum_{i=1}^m [f_{\theta}(x^{(i)}) - y^{(i)}] x_0^{(i)}$$

$$w_1 := w_1 - \alpha \frac{1}{m} \sum_{i=1}^m [f_{\theta}(x^{(i)}) - y^{(i)}] x_1^{(i)}$$

update θ simultaneously

$$w_2 := w_2 - \alpha \frac{1}{m} \sum_{i=1}^m [f_{\theta}(x^{(i)}) - y^{(i)}] x_2^{(i)}$$

.....

Feature scaling

Size (feet ²)	Number of bedrooms	Number of floors	Age of home (years)	Price (\$1000)
x_1	x_2	x_3	x_4	y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...

Get every feature into approximately a $-1 \leq x_i \leq 1$ range.

Normalization:

$$x_i = \frac{x_i - \mu_i}{S_i}$$

Standard deviation:

$$S_i = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu_i)^2}$$

For example:


$$x_1 = \frac{\text{size} - 1000}{2000}$$

$$x_2 = \frac{\text{\#bedrooms} - 2}{4}$$


...

Multivariate linear regression: Vectorization

	Size (feet ²)	Number of bedrooms	Number of floors	Age of home (years)	Price (\$1000)
x_0	x_1	x_2	x_3	x_4	y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178
...



$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}$$



$$y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

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

Normal equation

$(X^T X)^{-1}$ is inverse of matrix $X^T X$.

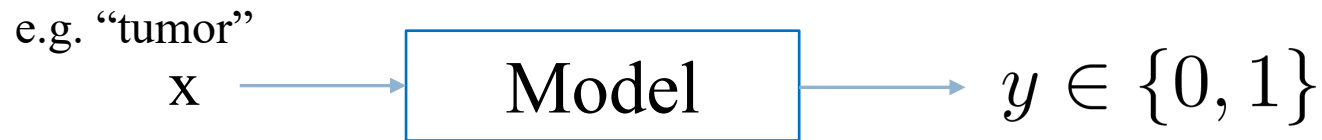
Logistic Regression

Binary Classification

Email: Spam / Not Spam?

Watermelon: Good / not?

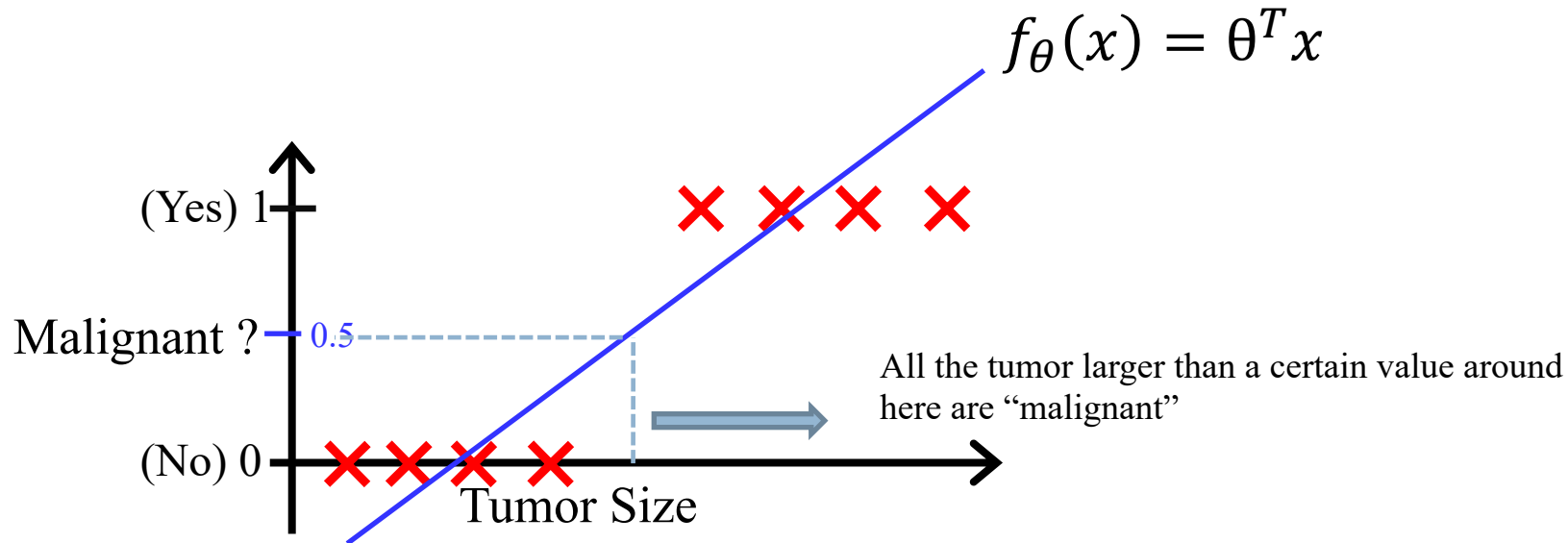
Tumor: Malignant / Benign ?



0: "Negative Class" (e.g., benign tumor)

1: "Positive Class" (e.g., malignant tumor)

Classification with linear regression?



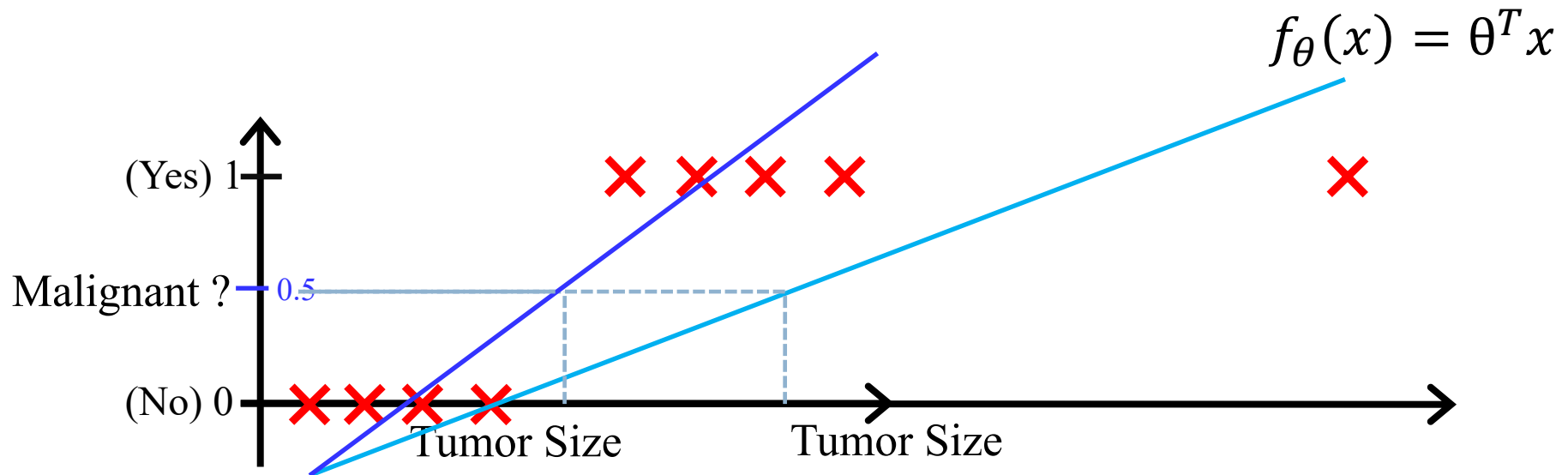
Hypothesis with linear regression : $f_{\theta}(x) = \theta^T x$

Threshold = 0.5

If $f_{\theta}(x) \geq 0.5$, output "Yes"

If $f_{\theta}(x) \leq 0.5$, output "No"

Classification with linear regression?



Hypothesis with linear regression : $f_{\theta}(x) = \theta^T x$

Threshold = 0.5

If $f_{\theta}(x) \geq 0.5$, output “Yes”

If $f_{\theta}(x) \leq 0.5$, output “No”

Add extra samples, **linear regression** may give you a worse hypothesis.
Applying **linear regression** to a **classification** is not a good idea.

Logistic regression model

Linear regression model: $f_{\theta}(x) = \theta^T x$

Linear regression can be: $f_{\theta}(x) > 1$ or $f_{\theta}(x) < 0$

Want: $0 \leq f_{\theta}(x) \leq 1$

$$f_{\theta}(x) = g(\theta^T x)$$



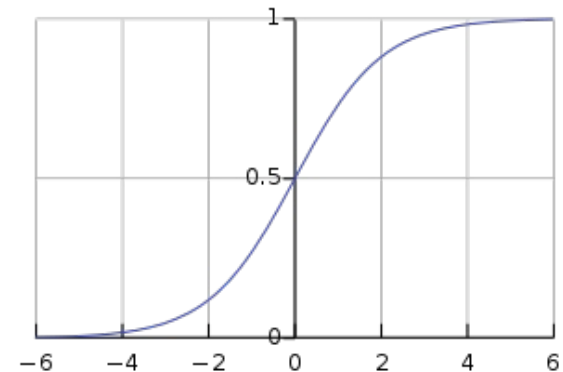
$$f_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

where,

$$g(z) = \frac{1}{1 + e^{-z}}$$




Sigmoid/Logistic
function



Training set:

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$$



$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} \quad x_0 = 1, y \in \{0, 1\}$$

Hypothesis: $f_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$ where, $\theta^T x = w_0 x_0 + w_1 x_1 + \dots + w_n x_n$

What cost function we choose?

How to choose parameters θ ?

Cost function

- Choose cost function same with linear regression

Cost Function:
$$E(\theta) = \frac{1}{2m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)})^2$$

But the hypothesis looks like:
$$f_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Problem: it is **non-convex function**, there are many local minums. Gradient descent algorithm not guranteed to find a good local minma.

Want: the cost function should be **convex**.

Hypothesis: $f_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$ where, $\theta^T x = w_0 x_0 + w_1 x_1 + \dots + w_n x_n$

Parameters: $\theta = \{w_0, \dots, w_n\}$ $y \in \{0, 1\}$ Real-value from data set

Cost function: $E(\theta) = -\frac{1}{m} [\sum_{i=1}^m y^{(i)} \log f_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log(1 - f_{\theta}(x^{(i)}))]$

if $y^{(i)} = 0$:

$$E(\theta) = -\frac{1}{m} [\sum_{i=1}^m (1 - y^{(i)}) \log(1 - f_{\theta}(x^{(i)}))]$$



Cross entropy loss function

if $y^{(i)} = 1$:

$$E(\theta) = -\frac{1}{m} [\sum_{i=1}^m y^{(i)} \log f_{\theta}(x^{(i)})]$$

Goal: minimize $E(\theta)$

Logistic regression: Gradient Descent

Cost function: $E(\theta) = -\frac{1}{m} [\sum_{i=1}^m y^{(i)} \log f_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log(1 - f_{\theta}(x^{(i)}))]$

Goal: minimize $E(\theta)$

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} E(\theta)$$

(simultaneously update all θ)

Learning rate

Gradient

}

Logistic regression: Gradient Descent

Cost function: $E(\theta) = -\frac{1}{m} [\sum_{i=1}^m y^{(i)} \log f_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log(1 - f_{\theta}(x^{(i)}))]$

Goal: minimize $E(\theta)$

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m [f_{\theta}(x^{(i)}) - y^{(i)}] x^{(i)}$$

$$f_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

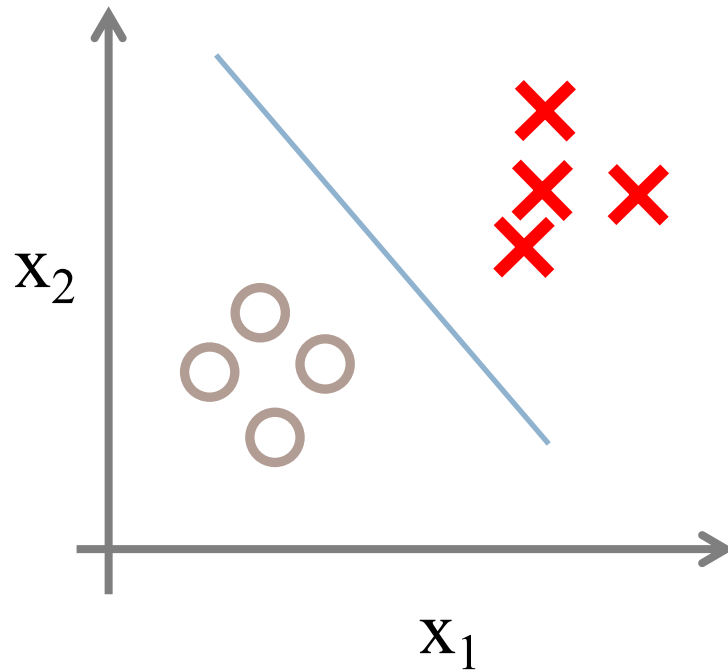
}

(simultaneously update all θ)

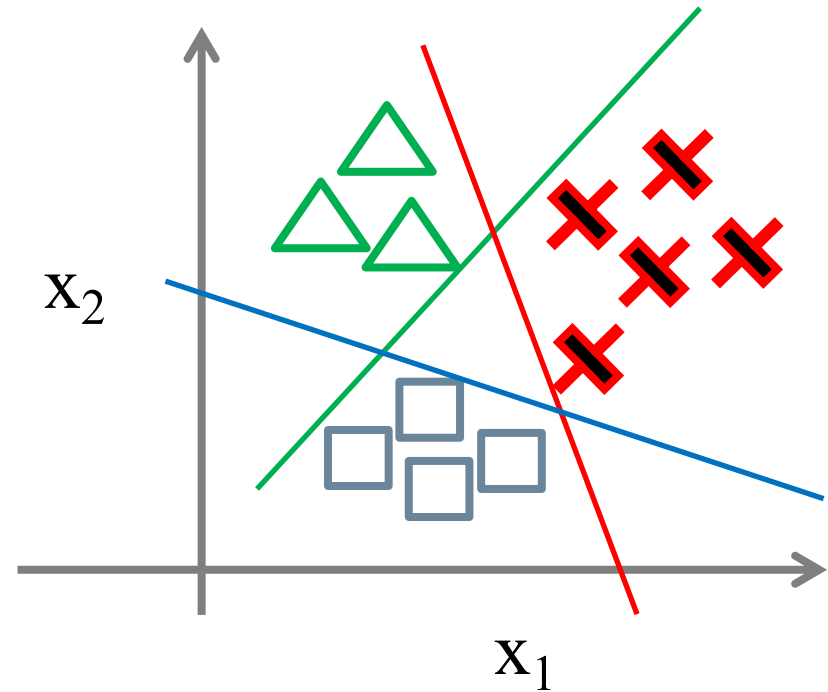
Similar with linear regression.

Multiclass classification

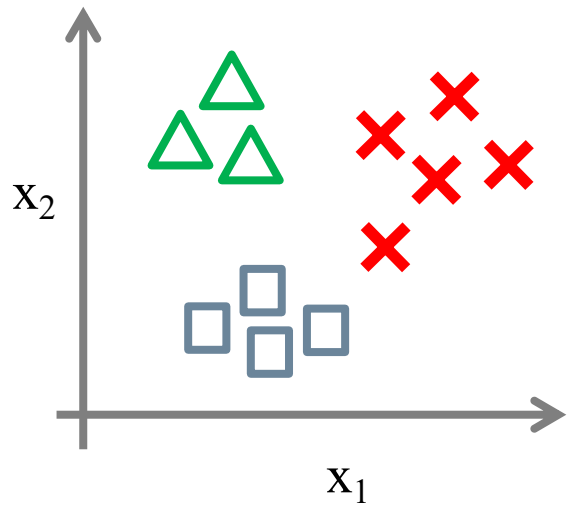
Binary classification:



Multi-class classification:



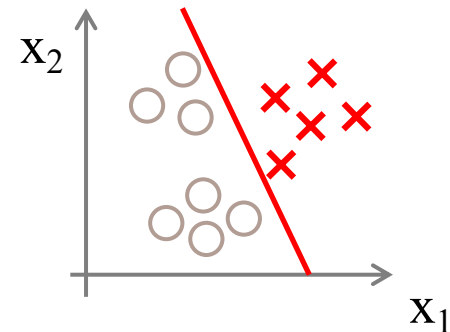
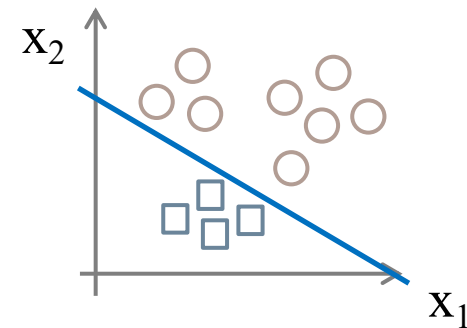
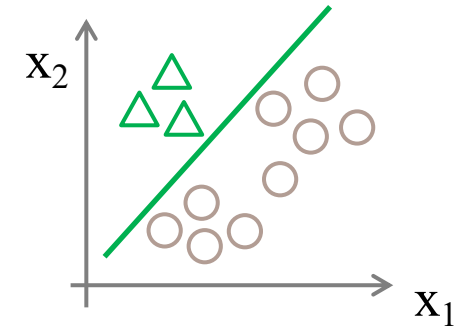
One-vs-rest



Class 1: 

Class 2: 

Class 3: 



- Thank you!