Linear Regression
Further to More Understanding of ML

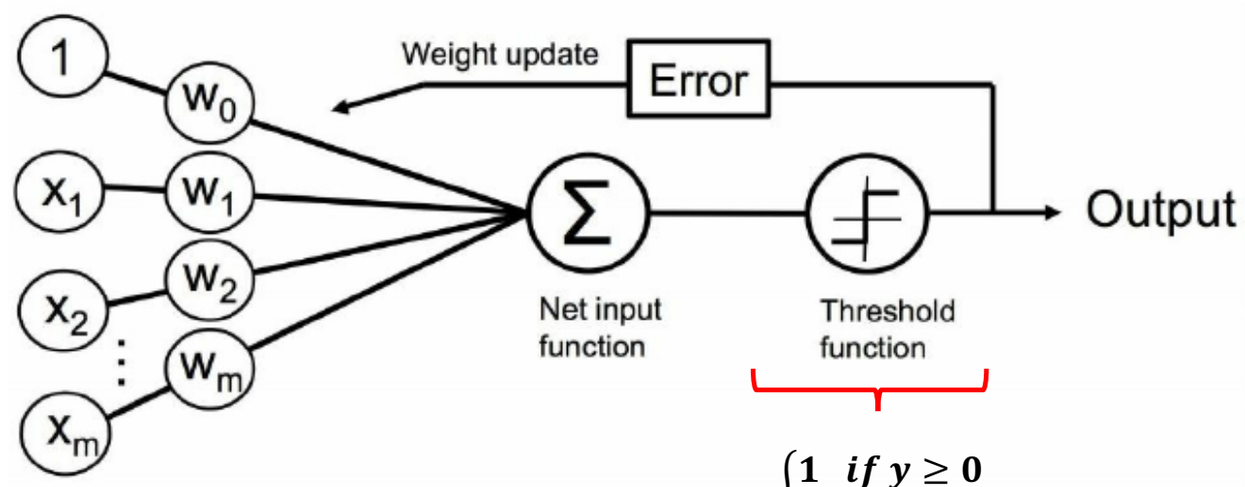
Jin Hyun Kim

In this Lecture

- Regression
- Linear Regression
- Gradient descendent algorithm

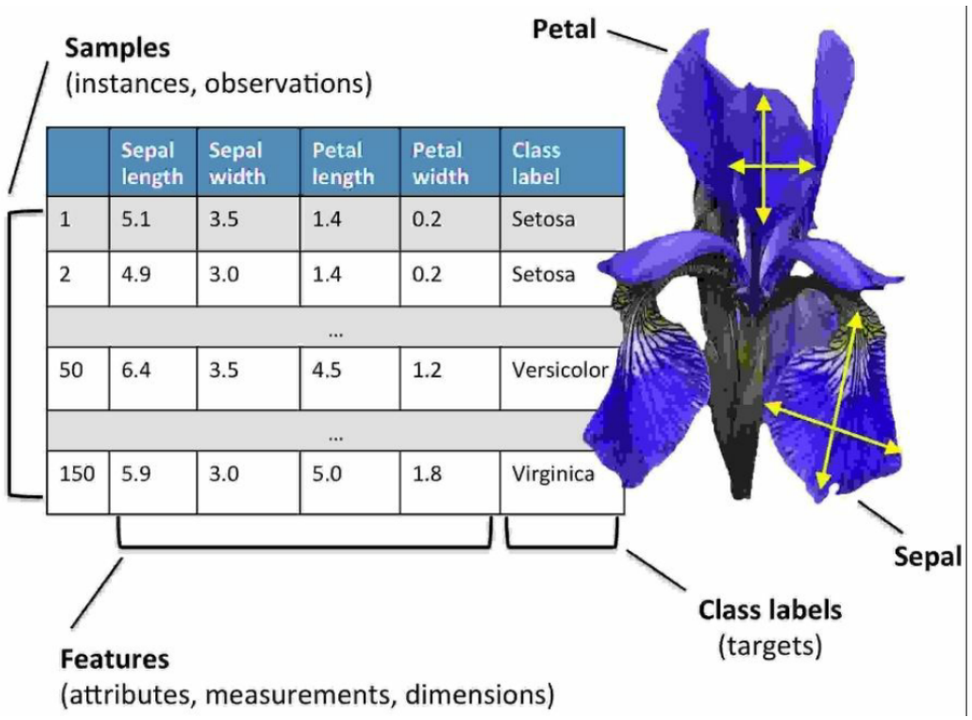
Review

- Perceptron
  - Classification - 0 / 1
  - e.g.
    - Spam Detection: Spam (1) or Ham(0)
    - Facebook feed: show(1) or hide(0)
    - Credit Card Fraudulent Transaction detection: legitimate(0) or fraud (1)

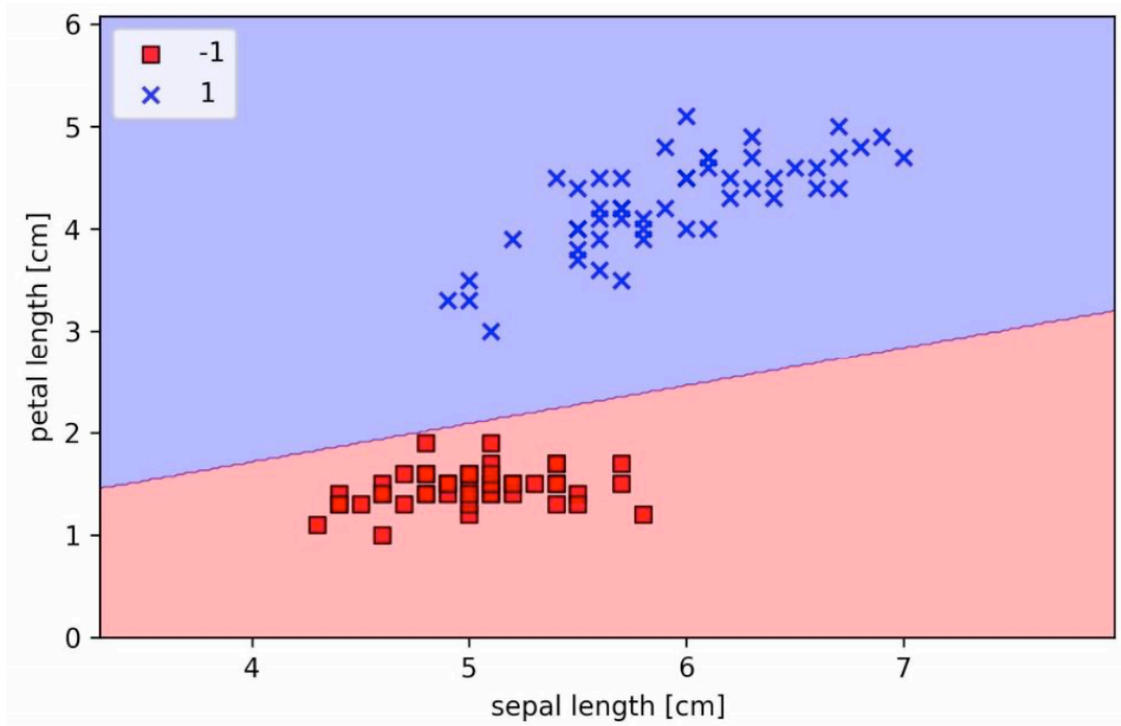


$$y = w_1 \cdot x_1 + \ldots + w_m \cdot x_m = \sum_{i=1}^{m} w_i \cdot x_i,$$

Review - Perceptron Lab



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica



Problem

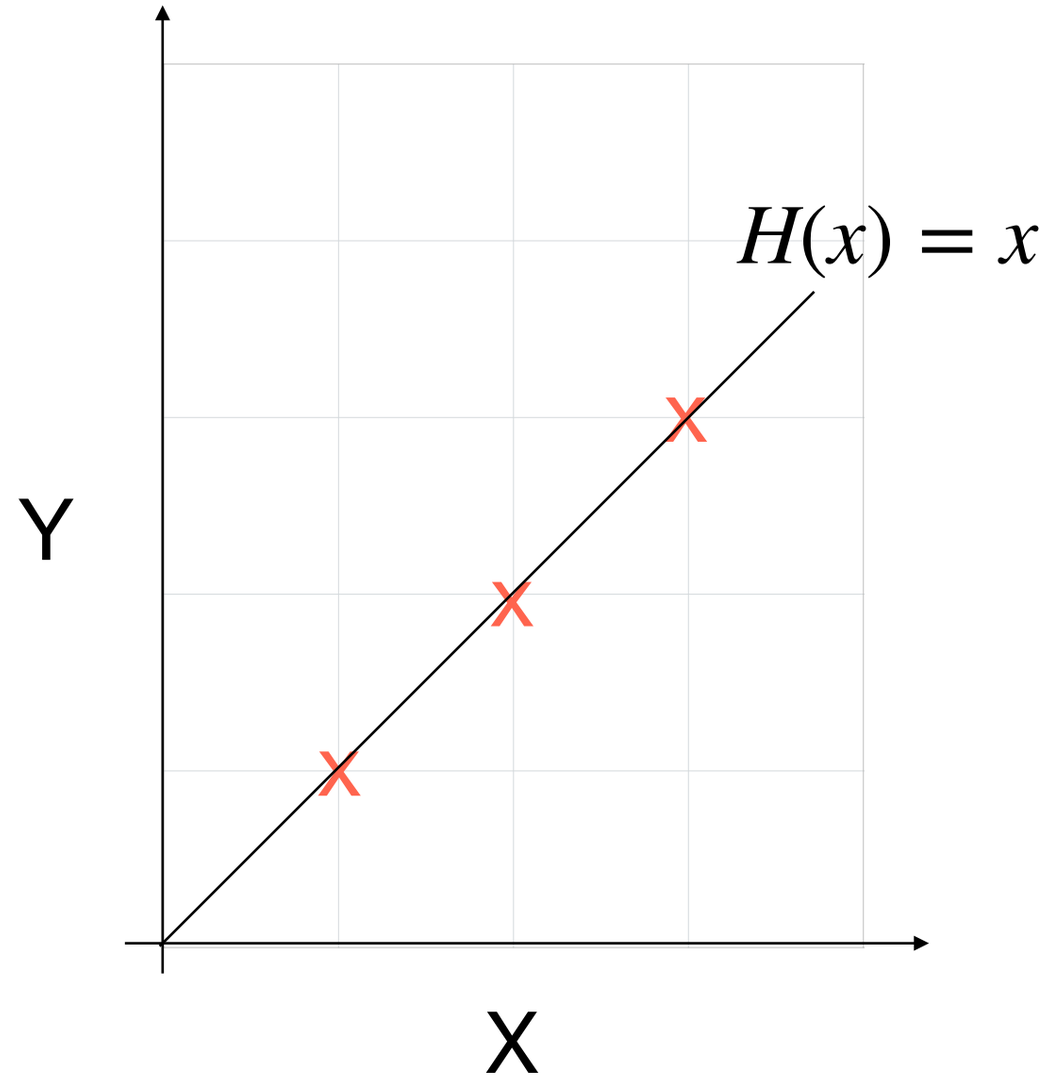
X	Y
1	1
2	2
3	3

키(단위: cm)	몸무게(단위: kg)
174	71
152	55
138	46
128	38
186	88

Regression
(Linear) Hypothesis

X	Y
1	1
2	2
3	3

$$H(x) = Wx + b$$



Linear Regression
Problems

- Given data $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$
  - N is the size of data
  - $x^{(i)}$ is a real number and the D-dimensional vectors of examples $i = 1,...,N$
  - $y^{(i)}$ is a real-valued target
- Find a solution that can predict a target value of y for a given x

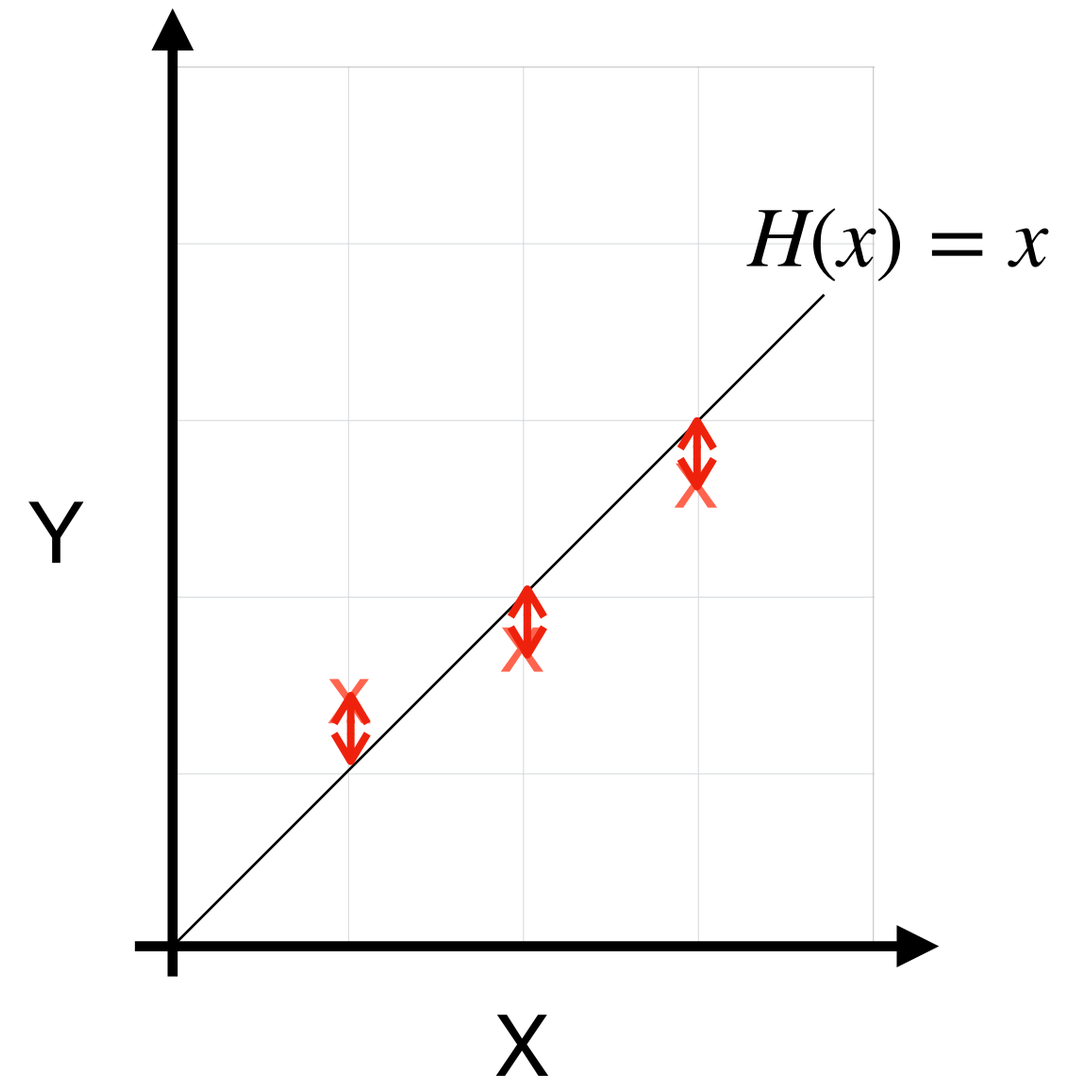
Height (cm)	Weight (kg)
175	71
152	46
138	46
128	38
186	88
190	?

Cost function

$$H(x) = Wx + b$$

$$H(x) - y$$

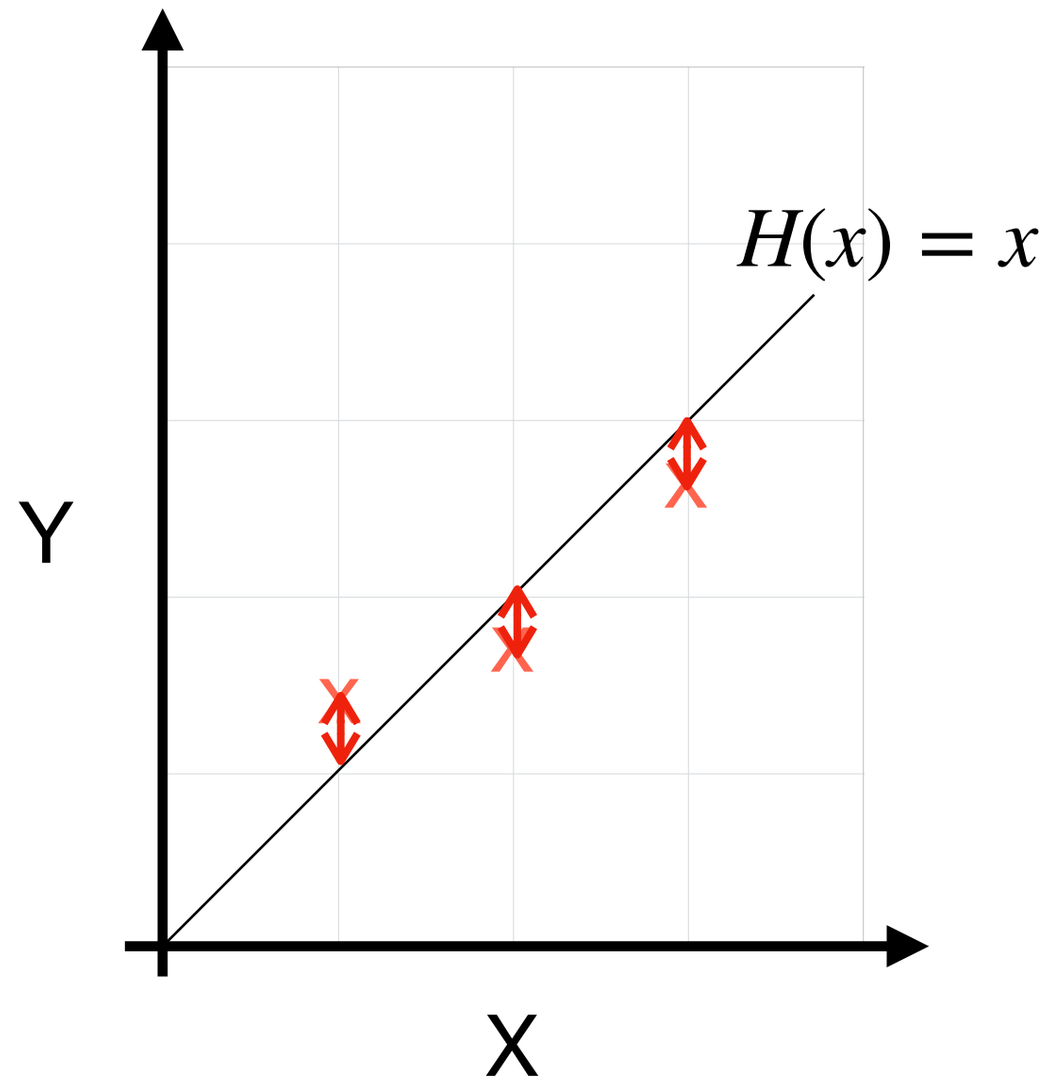
$$(H(x) - y)^2$$



Cost function

$$\frac{(H(x^{(1)}) - y^{(1)})^2 + (H(x^{(2)}) - y^{(2)})^2 + (H(x^{(3)}) - y^{(3)})^2}{3}$$

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

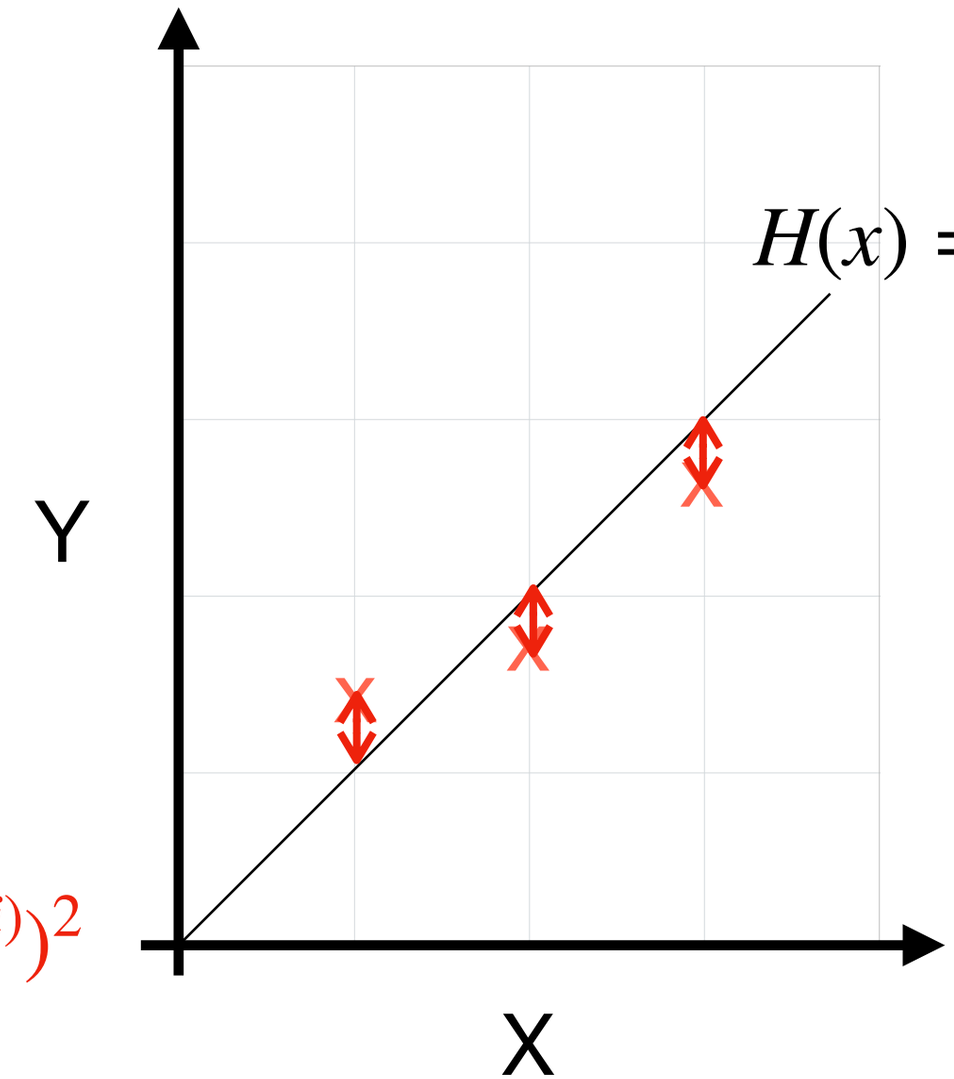


Cost function

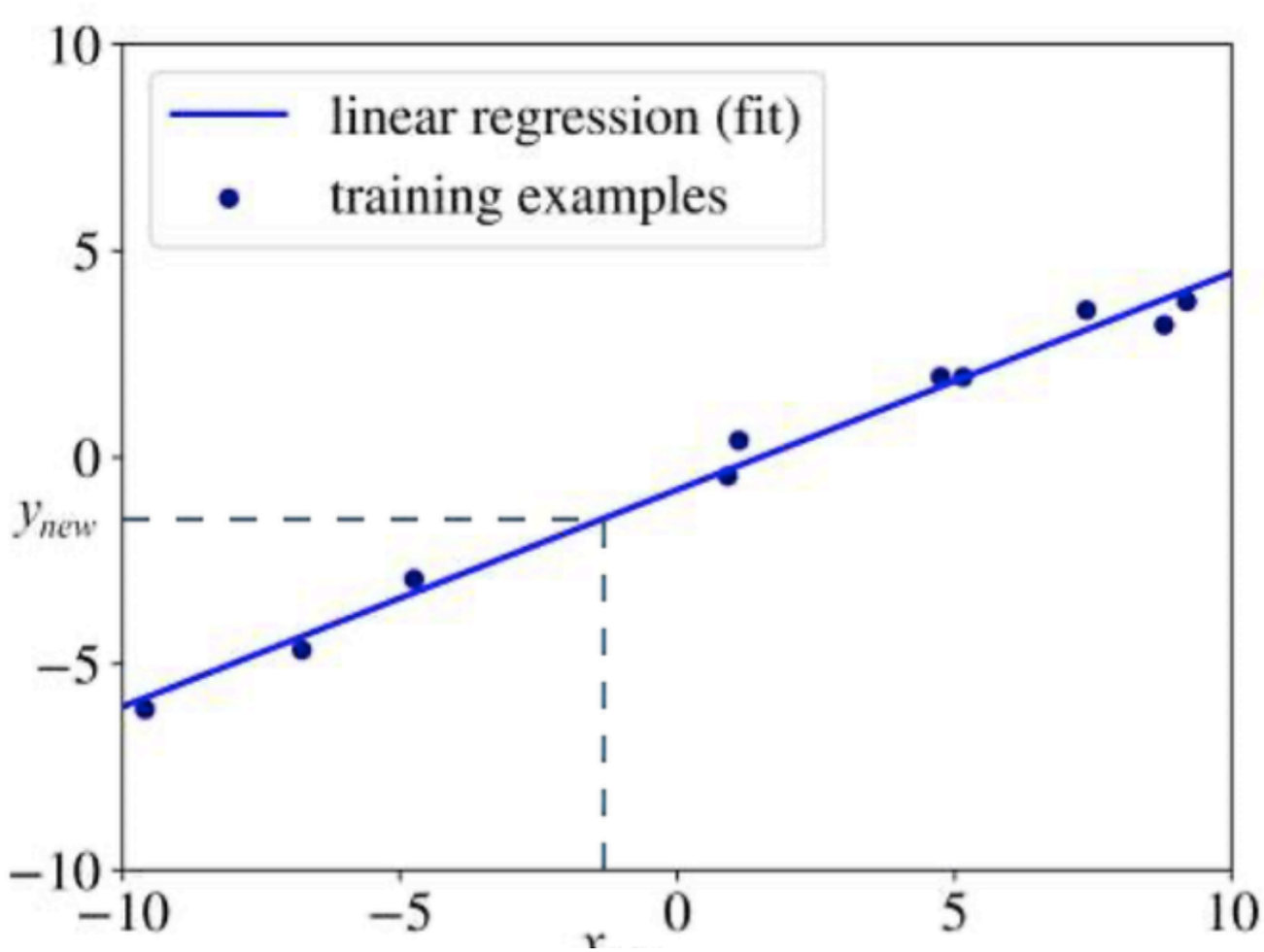
$$cost = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$

$$H(x) = Wx + b$$

$$cost(W, b) = \frac{1}{N}\sum_{i=1}^{N}((Wx^{(i)} + b) - y^{(i)})^2$$



Linear Regression



Linear Regression
Solution

- Find a model minimizing the cost function

$$cost(W, b) = \frac{1}{N} \sum_{i=1}^{N} ((Wx^{(i)} + b) - y^{(i)})^2$$

Linear Regression

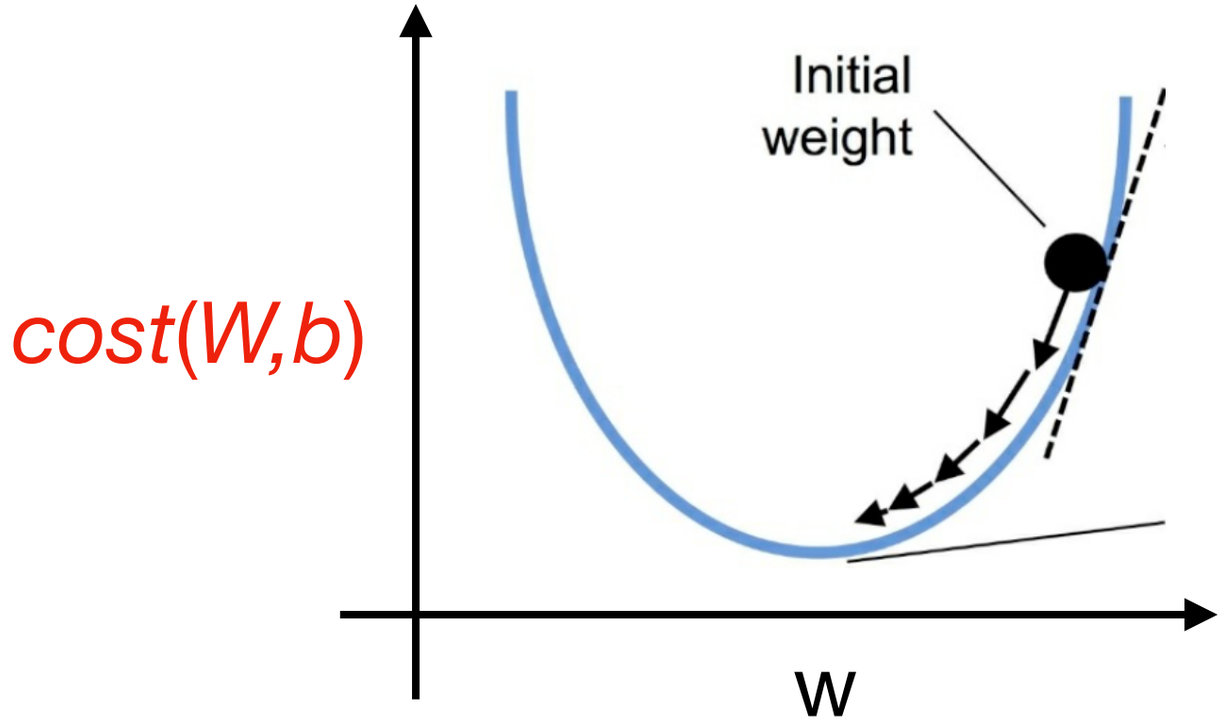
Objective

- Goal: Find a solution that minimizes the cost function

$$\min_{w,b} cost(W, b)$$

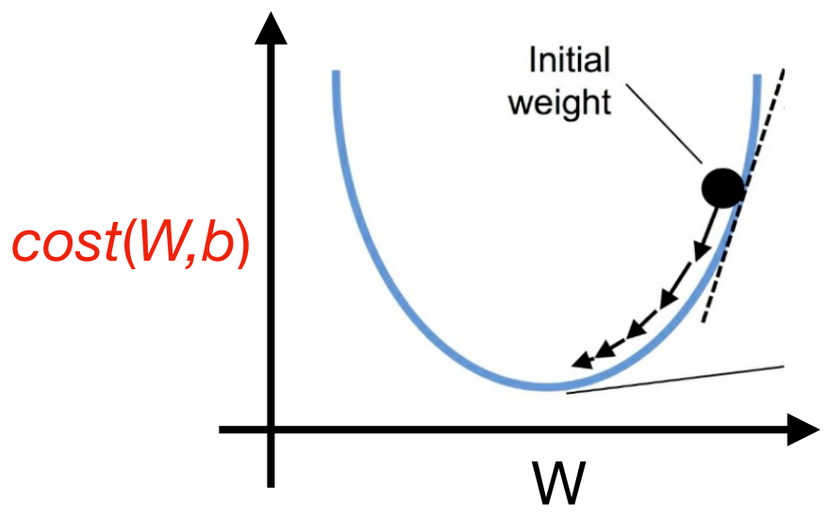
Gradient Descendent Algorithm

- Find *W* and *b* where the derivative of the cost function is 0



Gradient Descendent Algorithm

- Start with initial guesses
    - Start with random value
    - Keeping changing W and b a little bit to try and reduce cost(W, b)
- Each time you change the parameters, select the gradient which reduces cost(W, b) the most possible
- Repeat until it converges to a local minimum



Gradient Descendent Algorithm

$$cost(W,b) = \frac{1}{N}\sum_{i=1}^{N}((Wx^{(i)}+b)-y^{(i)})^2$$

$$\frac{\partial}{\partial W}cost(W,b) = \frac{2}{N}\sum_{i=1}^{N}x^{(i)}((Wx^{(i)}+b)-y^{(i)})$$

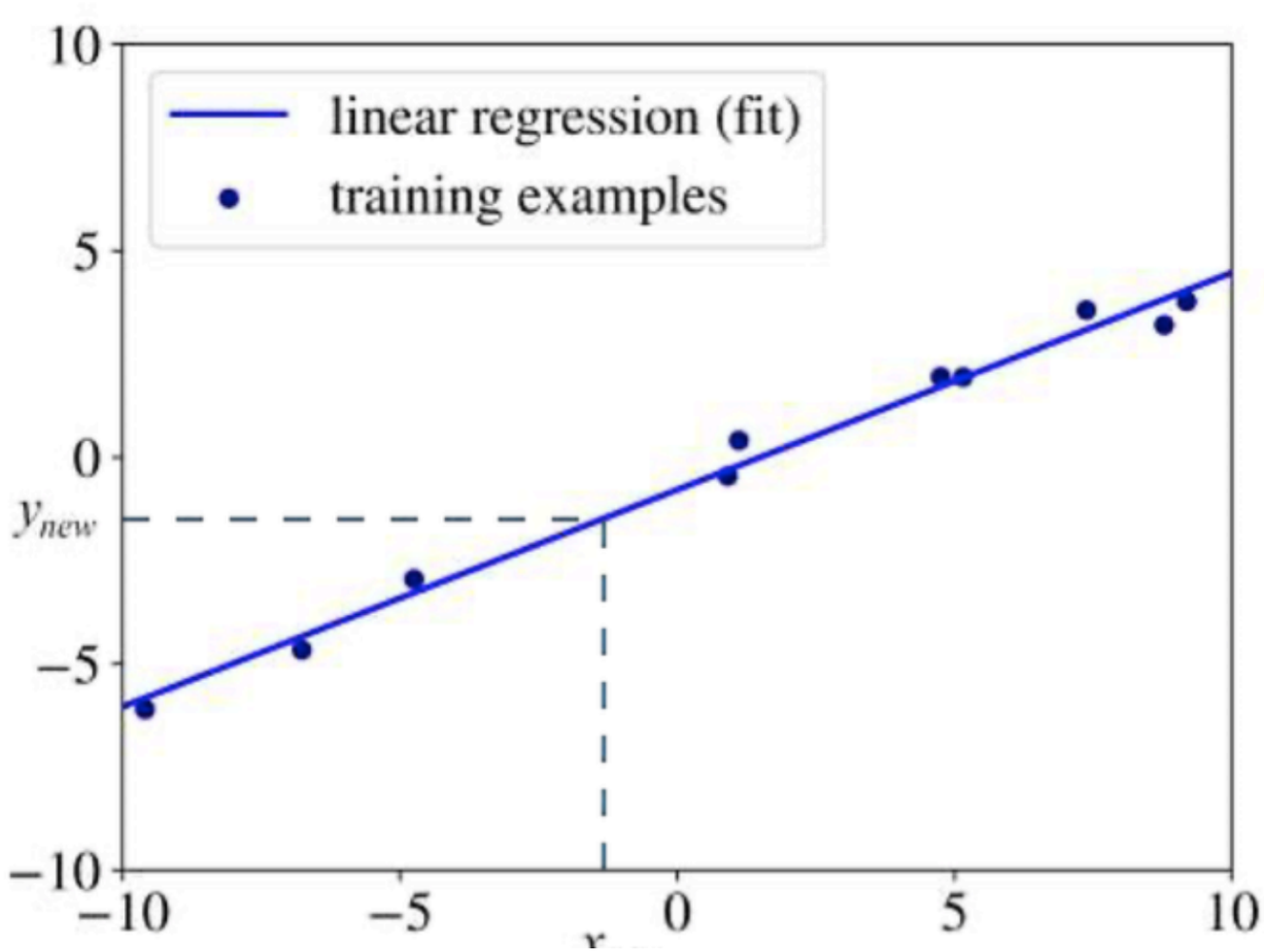
$$\frac{\partial}{\partial b}cost(W,b) = \frac{2}{N}\sum_{i=1}^{N}((Wx^{(i)}+b)-y^{(i)})$$

Gradient Descendent Algorithm

$$W = W - \alpha \frac{\partial}{\partial W} cost(W, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} cost(W, b)$$

Linear Regression



Next

- Linear Regression Lab