Introduction

- What is Machine Learning?

Machine Learning: A Definition

Definition: The field of study that gives computers the ability to learn without being explicitly learned.

(Arthur Samuel-1950)

Machine Learning: A Definition

Definition: A computer program is said to *learn* from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

(Tom Mitchell-1998)

Case study 1: Spam/Not spam emails

- Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam. What is the task T in this setting?
  - **The task T** is classifying emails as spam or not spam
  - **The experience E** is watching you label emails as spam or not spam
  - **The performance P** is the number of emails correctly classified as spam or not spam

Case study 2: Handwriting recognition learning

- **Task T :** recognizing and classifying handwritten words within images
- **Performance measure P :** percent of words correctly classified
- **Training experience E:** a database of handwritten words with given classifications

Machine Learning Approaches

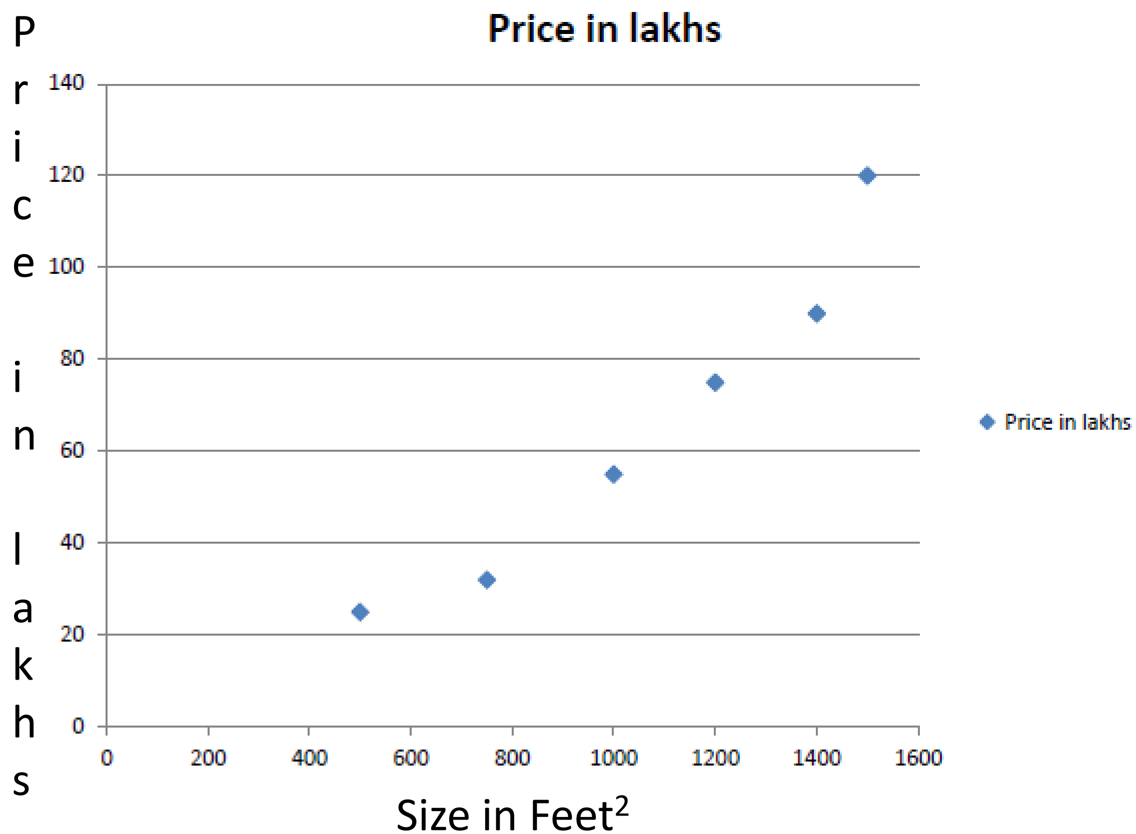
- Supervised learning
- Un-supervised learning

Supervised Learning

- Input and output labels are given.
- Categorized into:
  - regression and
  - Classification
- Regression:
  - map input variables to some continuous function
  - predict results within a continuous output
- Classification:
  - map input variables into discrete categories
  - predict results in a discrete output

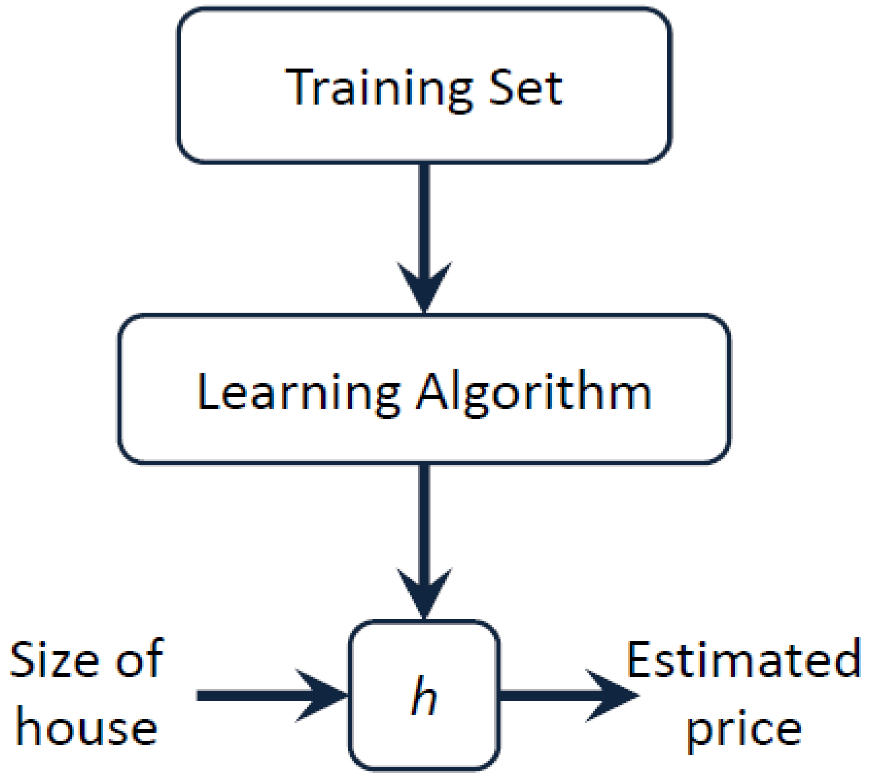
Example 1: Housing price prediction

Size in Feet^2	Price in lakhs
500	25
750	32
1000	55
1200	75
1400	90
1500	120



Supervised Learning : Labeled data is given.

Housing Price Prediction



x

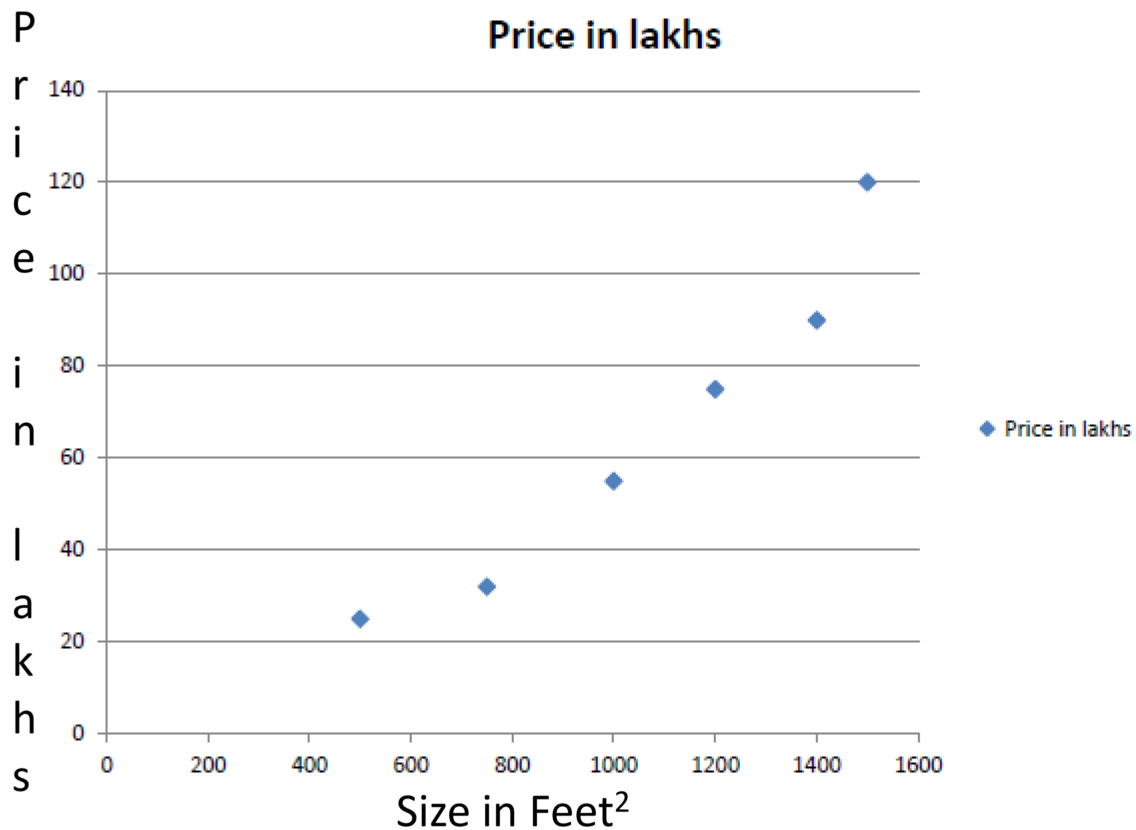
hypothesis

Estimated value of y

h maps from x's to y's

Example: Housing price prediction

Size in Feet^2	Price in lakhs
500	25
750	32
1000	55
1200	75
1400	90
1500	120



Supervised Learning : Labeled data is given.

Regression : Predict continuous valued output(price)

Example2 : Positive/Negative Sentiment Prediction

Doc ID	Sentiment
D1	+ve
D2	+ve
D3	-ve
...	
...	
D1000	-ve

Positive Sentiment features : good, extraordinary, cool, awesome, attractive, special, etc.,

Negative Sentiment features : not good, bad, worse, hate, sad, abused, awkward, dark, etc.,

Example2 : Positive/Negative Sentiment Prediction

Doc ID	Sentiment
D1	+ve
D2	+ve
D3	-ve
…	
…	
D1000	-ve

Positive Sentiment features : good, extraordinary, cool, awesome, attractive, special, etc.,

Negative Sentiment features : not good, bad, worse, hate, sad, abused, awkward, dark, etc.,

Classification : Discrete valued output (+ve or –ve)

You're running a company, and you want to develop learning algorithms to address each of two problems.

Problem 1: You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.
Problem 2: You'd like software to examine individual customer accounts, and for each account decide if it has been hacked/compromised.
Should you treat these as classification or as regression problems?

1. Treat both as classification problems.
2. Treat problem 1 as a classification problem, problem 2 as a regression problem.
3. Treat problem 1 as a regression problem, problem 2 as a classification problem.
4. Treat both as regression problems.

Unsupervised Learning

Unsupervised Learning

- Input is known but output is not known.

Unsupervised Learning

Example 1:Given a collection of text documents, organize them according to content similarity, to produce a topic hierarchy.

Unsupervised Learning

Example 1:Given a collection of text documents, organize them according to content similarity, to produce a topic hierarchy.

Example 2: In marketing, segment customers according to similarities, to do targeted marketing.

Unsupervised Learning

Example 1:Given a collection of text documents, organize them according to content similarity, to produce a topic hierarchy.

Example 2: In marketing, segment customers according to similarities, to do targeted marketing.

Example 3: On social networks, identifying research communities working on same problem.

Of the following examples, which would you address using an unsupervised learning algorithm? (Select all that apply.)

- Given email labeled as spam/not spam, learn a spam filter.
- Given a set of news articles found on the web, group them into set of articles about the same story.
- Given a database of customer data, automatically discover market segments and group customers into different market segments.
- Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not.

Linear Regression

Linear Regression with one variable

- **Single feature(variable)**

Size ($feet^2$)	Price (lakhs)
2104	460
1416	232
1534	315
852	178
…	…

- **Model**

$$h_w(x) = w_0 + w_1 * x_1$$

Linear Regression with one variable

- **Single feature(variable)**

Size (feet^2) **x**	**Price (lakhs)** **y**
2104	460
1416	232
1534	315
852	178
…	…

- **Hypothesis Model**

$$h_w(x) = w_0 + w_1 * x$$

Example:

- Suppose we have the following set of training data:

input x	output y
0	4
1	7
2	7
3	8

- Now we can make a random guess about our $h_w(x)$ function: $w_0=2$ and $w_1=2$. The hypothesis function becomes $h_w(x)=2+2x$.
- For input of 1 , hypothesis, y will be 4

Linear Regression with Multiple variables

- **Multiple features(variables)**

Size ($feet^2$) x1	Number of bedrooms x2	Number of floors x3	Age of home (years) x4	Price (lakhs) Y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

- **Hypothesis Model**

$$h_w(x) = h(x) = w_0 + w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + w_4 * x_4$$

Hypothesis

$$h(x) = w_0 + w_1*x_1 + w_2*x_2 + w_3*x_3 + w_4*x_4$$

Example:

$$h(x) = 80 + 0.1*x1 + 0.01 *x2 + 3*x3 - 2 *w4$$

Now,

Input is a vector of the form

$$X = \begin{matrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{matrix} \in \mathrm{R}^{n+1}$$

W's is a vector of the form

$$W = \begin{matrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ \dots \\ w_n \end{matrix} \in \mathrm{R}^{n+1}$$

Hypothesis

Input is a vector of the form

$$X = \begin{matrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_n \end{matrix} \in \mathbb{R}^{n+1}$$

W's is a vector of the form

$$W = \begin{matrix} w_0 \\ w_1 \\ w_2 \\ \dots \\ w_n \end{matrix} \in \mathbb{R}^{n+1}$$

Now, $h(x) = w_0 + w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + w_4 * x_4$

For convenience of notation, define $x_0=1$

Therefore, $h(x) = W^T X = [w_0 \quad w_1 \quad w_2 \quad \dots \quad w_n] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$

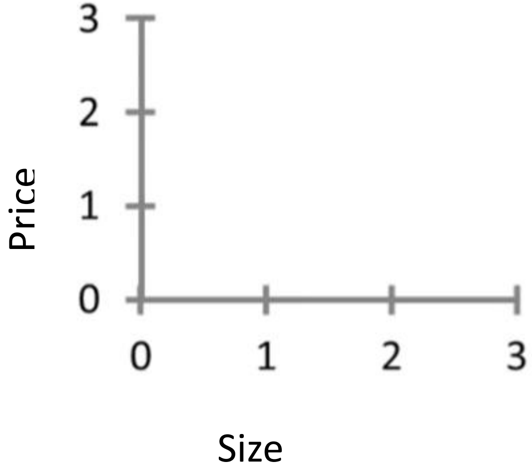
(n+1) x 1 matrix

1x (n+1) matrix

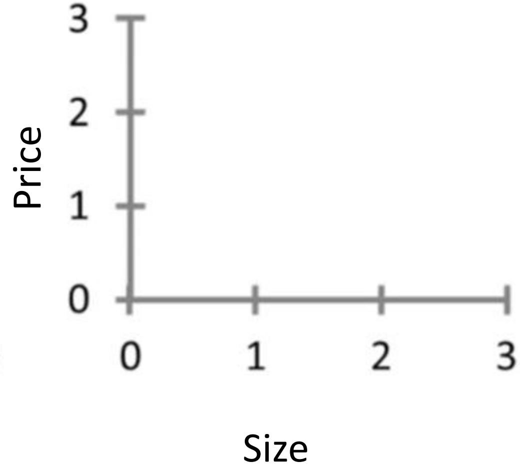
Cost Function

Example: Linear regression (housing prices)

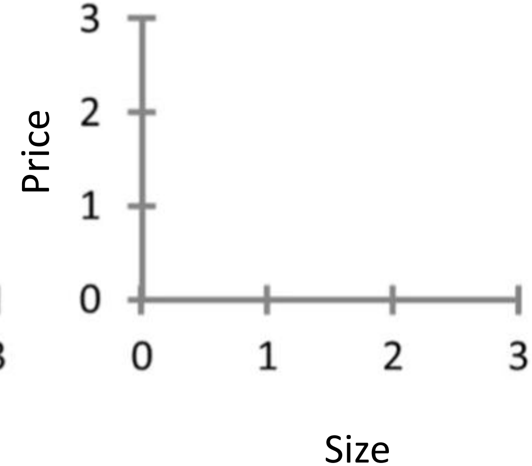
Hypothesis function: $h_w(x) = w_0 + w_1 * x$



$w_0 = 1.5$
$w_1 = 0$



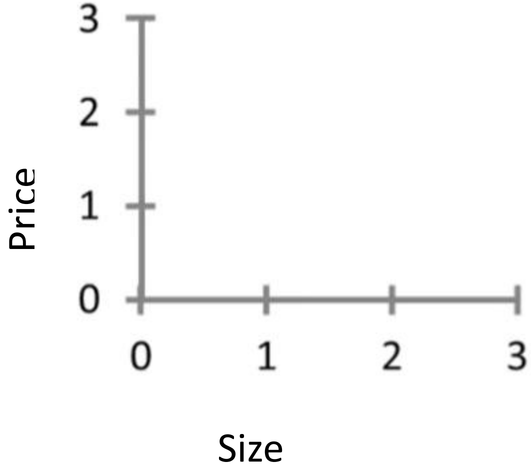
$w_0 = 0$
$w_1 = 0.5$

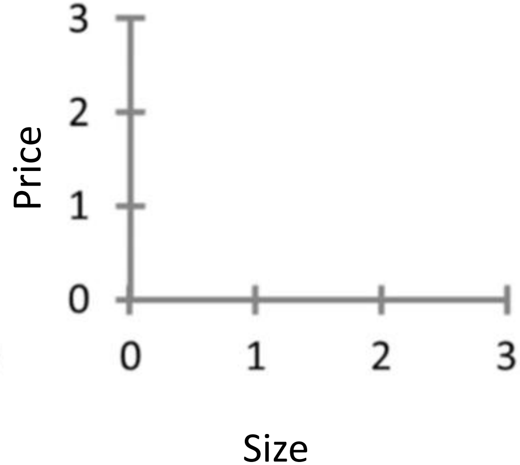


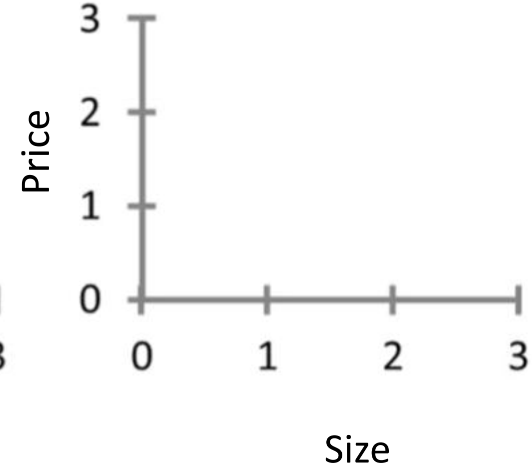
$w_0 = 1$
$w_1 = 0.5$

Example: Linear regression (housing prices)

Hypothesis function: $h_w(x) = w_0 + w_1 * x$



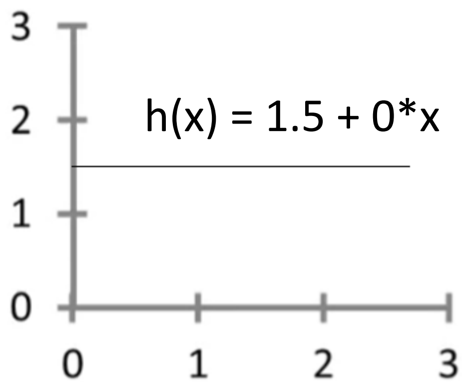


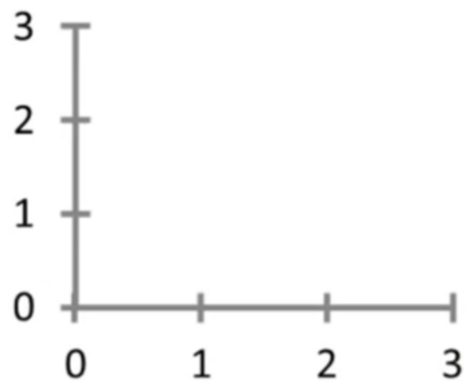


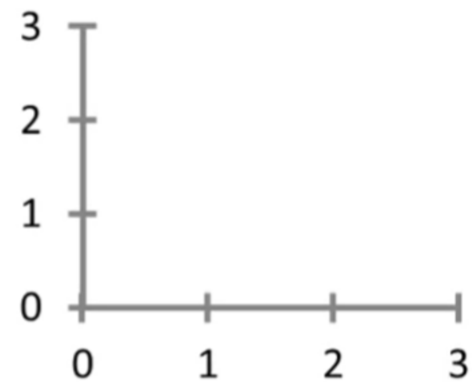
$w_0 = 1.5$
$w_1 = 0$

$w_0 = 0$
$w_1 = 0.5$

$w_0 = 1$
$w_1 = 0.5$

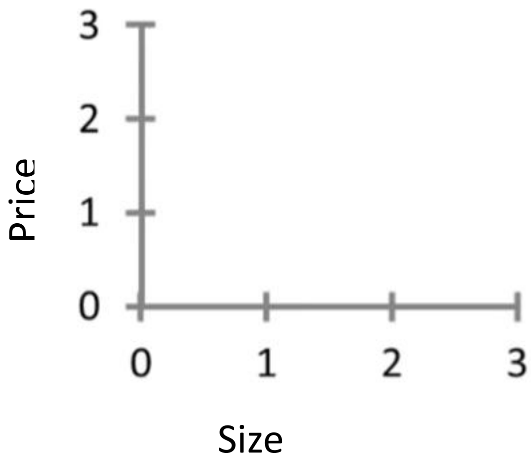


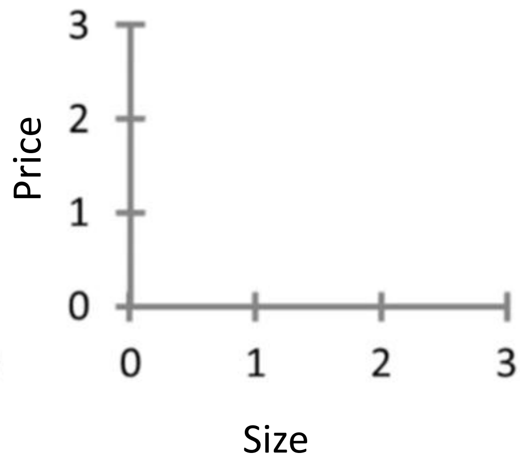


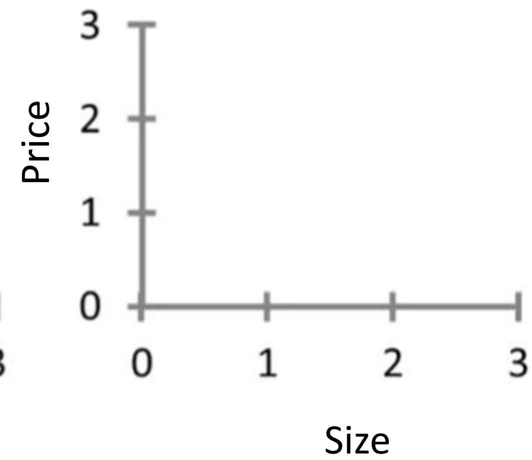


Example: Linear regression (housing prices)

Hypothesis function: $h_w(x) = w_0 + w_1 * x$



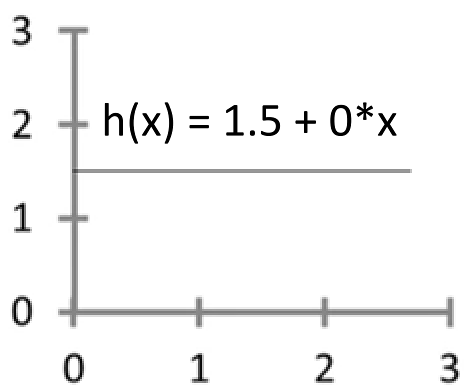


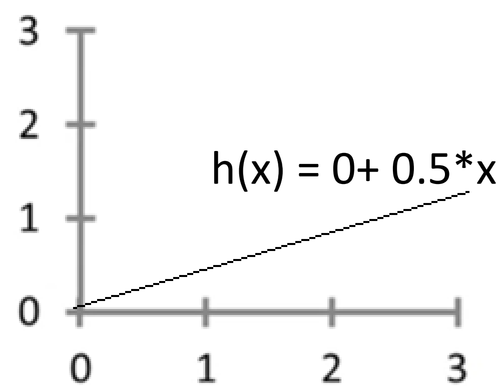


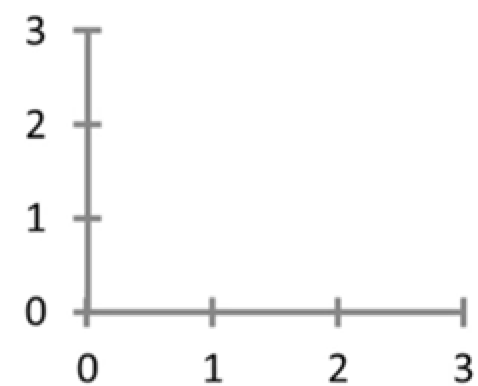
$w_0 = 1.5$
$w_1 = 0$

$w_0 = 0$
$w_1 = 0.5$

$w_0 = 1$
$w_1 = 0.5$

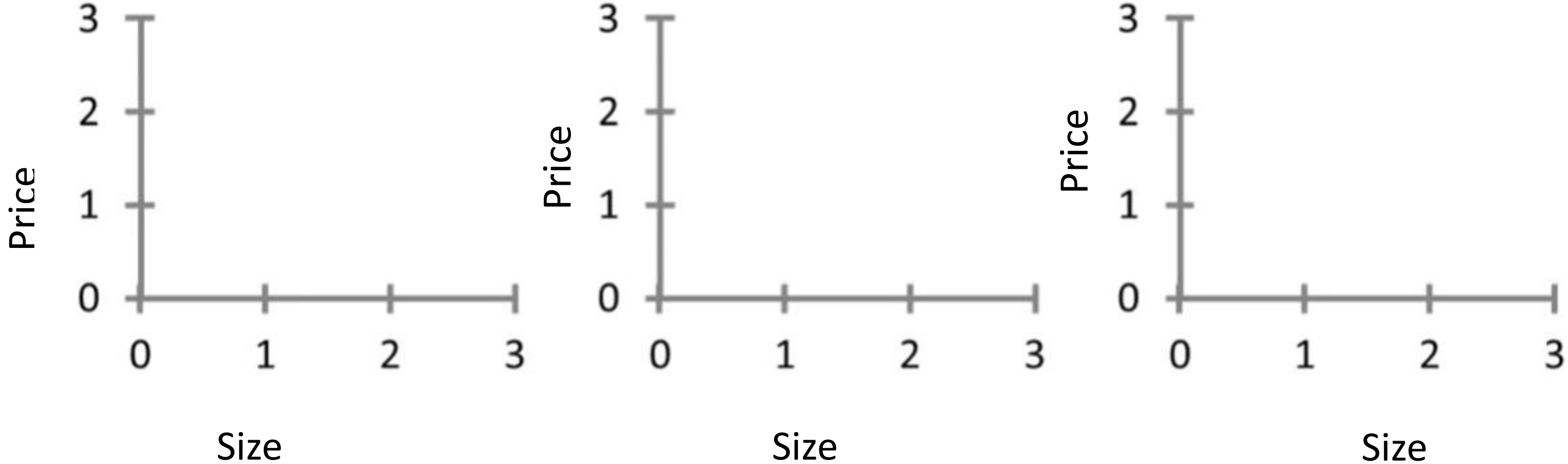






Example: Linear regression (housing prices)

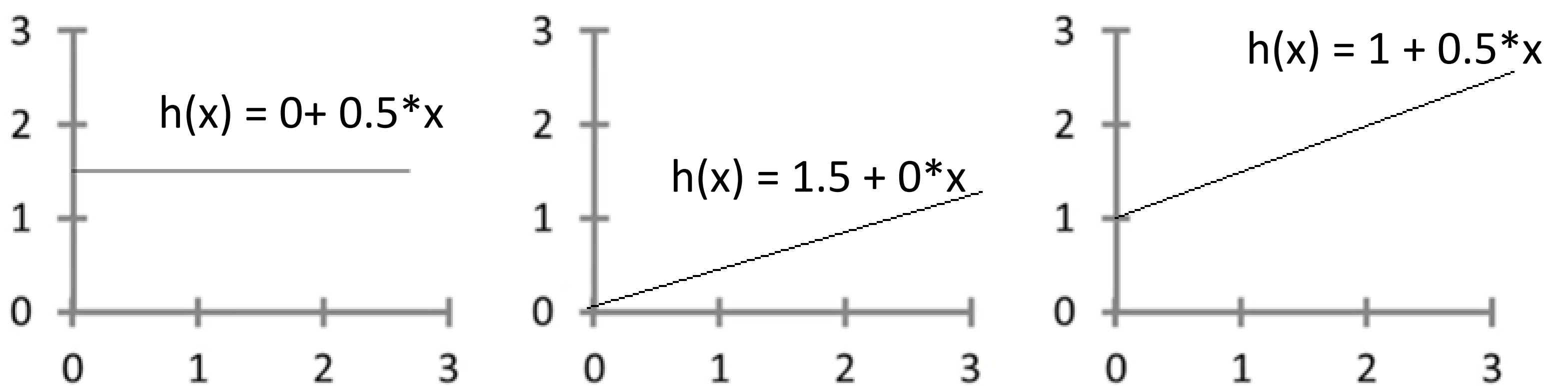
Hypothesis function: $h_w(x) = w_0 + w_1 * x$



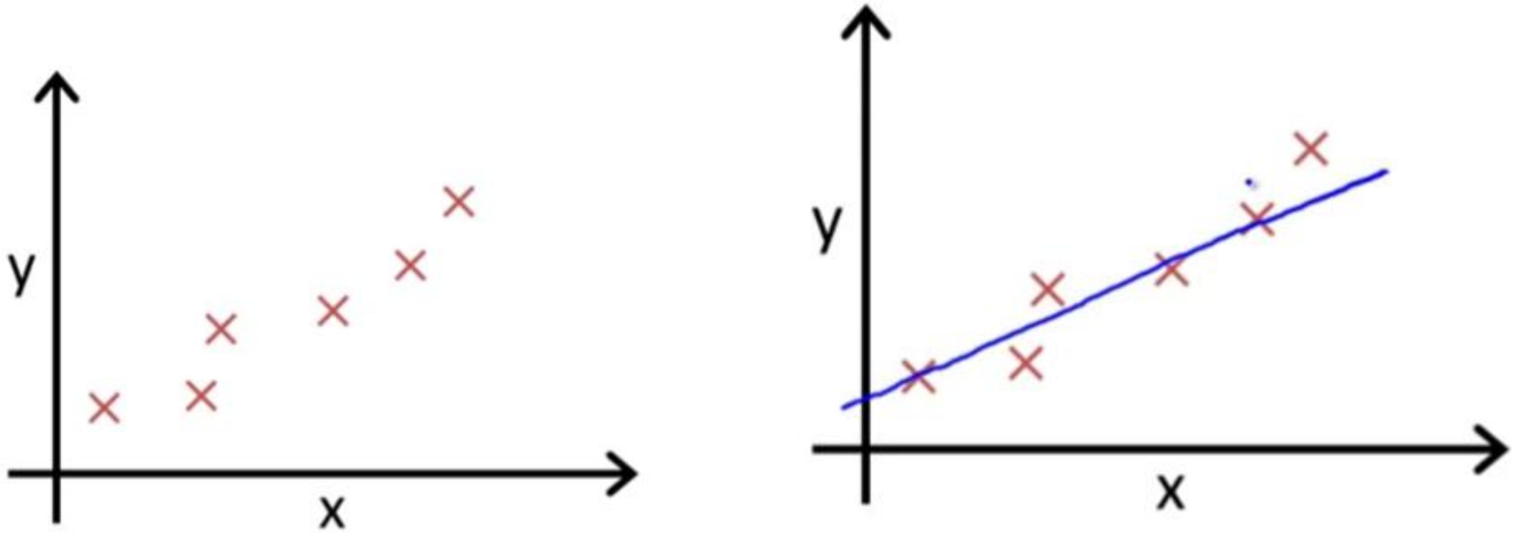
$w_0 = 1.5$
$w_1 = 0$

$w_0 = 0$
$w_1 = 0.5$

$w_0 = 1$
$w_1 = 0.5$



How to choose parameters?



Idea is to choose w_0 , w_1 so that $h_w(x)$ is close to y for training examples (x, y)

$$\underset{w0,\ w1}{\text{minimize}} \frac{1}{2m} \sum_{i=1}^{m} \left(h_w(x_i) - y_i\right)^2$$

where $h_w(x) = w_0 + w_1 * x$

Cost Function

Cost Function: $J(w_0 , w_1)$: This takes an average difference of all the results of the hypothesis with inputs from x's and the actual output y's.

$$J(w_0 , w_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_{\mathrm{w}}(x_i) - y_i\right)^2$$

Minimize the cost function i.e.,

$$\underset{w0,\ w1}{\text{minimize}} \ \frac{1}{2m} \sum_{i=1}^{m} \left(h_{\mathrm{w}}(x_i) - y_i\right)^2$$

Hypothesis:

$h_w(x) = w_0 + w_1 * x$

Parameters:

$w_0 + w_1$

Cost Function:

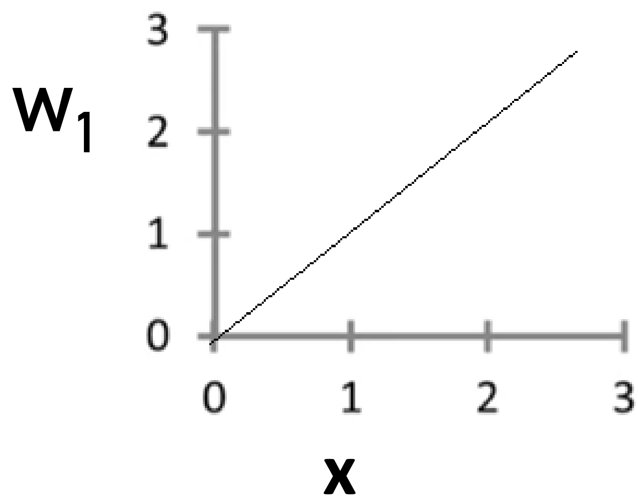
$J(w_0, w_1)$= $\frac{1}{2m}\sum_{i=1}^{m}\left(h_w(x_i) - y_i\right)^2$

Goal: minimize $J(w_0, w_1)$

w0, w1

Simplified Hypothesis

$h_w(x) = w_1 * x$



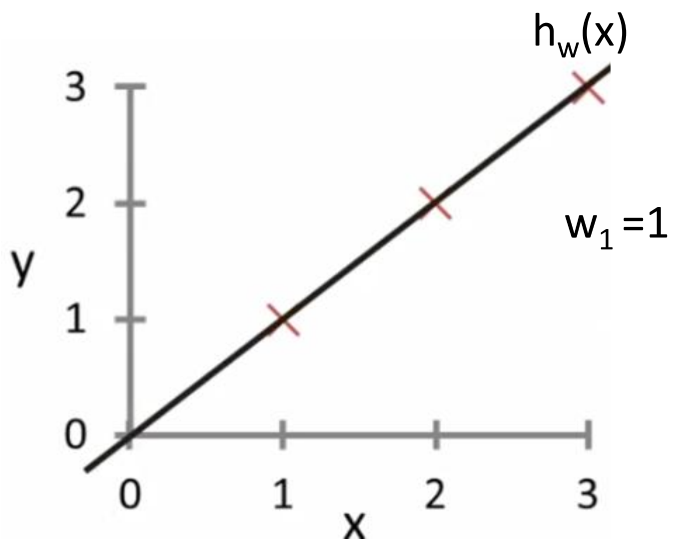
$$J(w_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_w(x_i) - y_i\right)^2$$

minimize
w0, w1

Visualization of Cost Function J

Hypothesis , $h_w(x)$

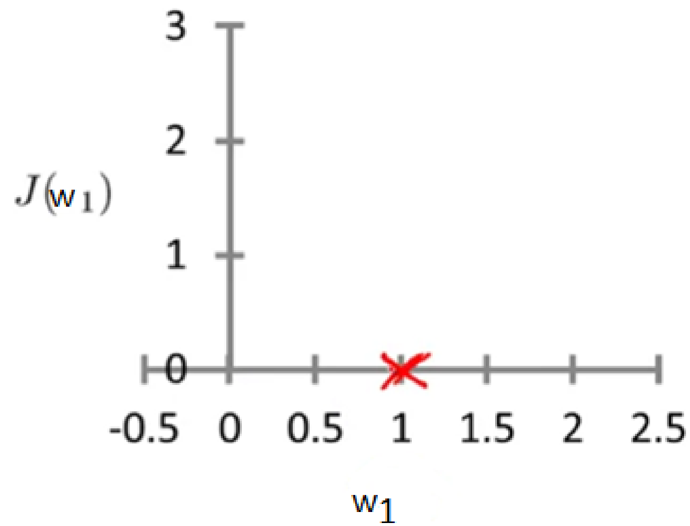
(For fixed w_1, this is function of x)



w_1 =1

Cost function: J(w1)

(function of parameter w1)



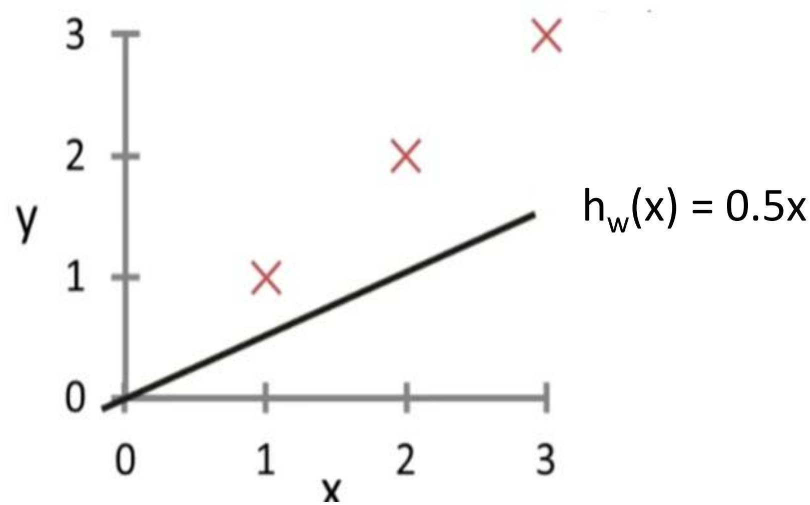
w_1 =1

J(1) =(1/2*3) [((1-1)^2)+((2-2)^2) + ((3-3)^2)]
J(1)= (1/6) [0^2 +0^2 + 0^2]
J(1) =(1/6)[0] = 0

Visualization of Cost Function J

Hypothesis , $h_w(x)$

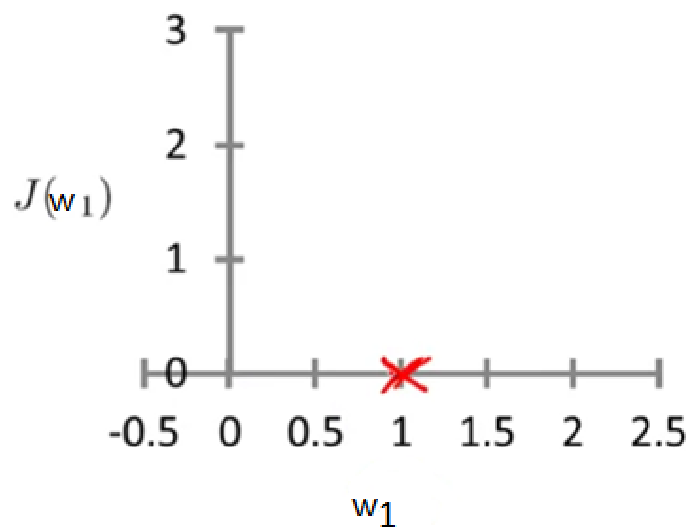
(For fixed w_1, this is function of x)



w_1 =0.5

Cost function: J(w1)

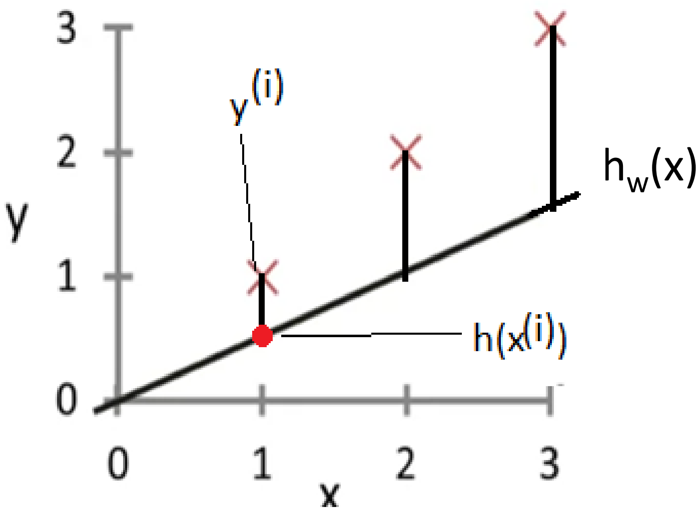
(function of parameter w1)



Visualization of Cost Function J

Hypothesis , $h_w(x)$

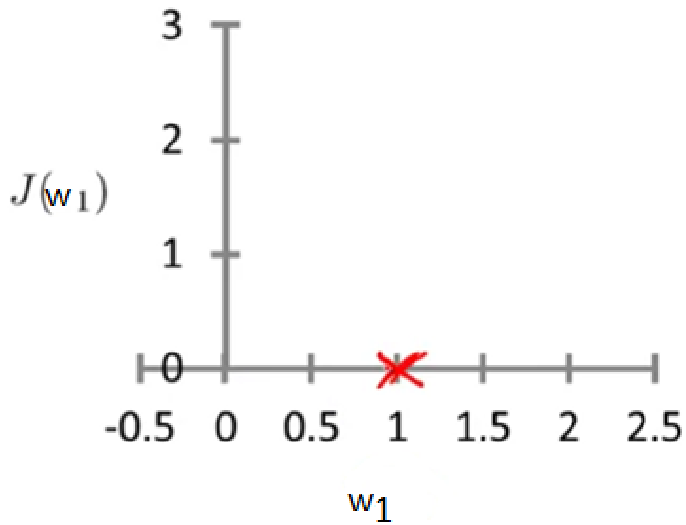
(For fixed w_1, this is function of x)



w_1 =0.5

Cost function: J(w1)

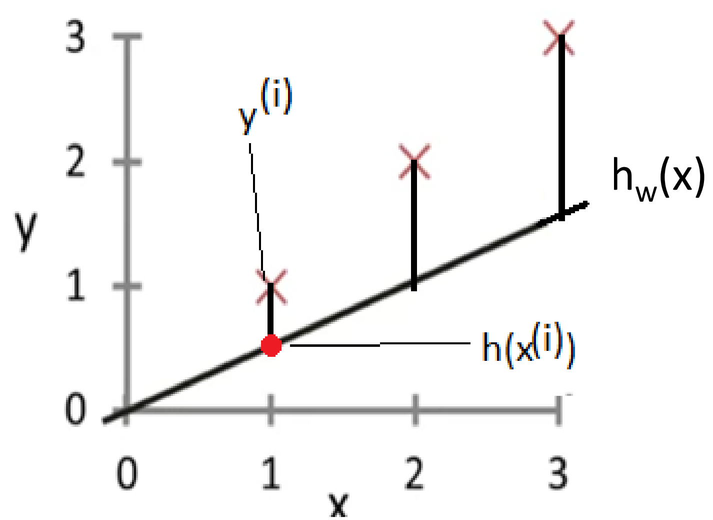
(function of parameter w1)



Visualization of Cost Function J

Hypothesis , $h_w(x)$

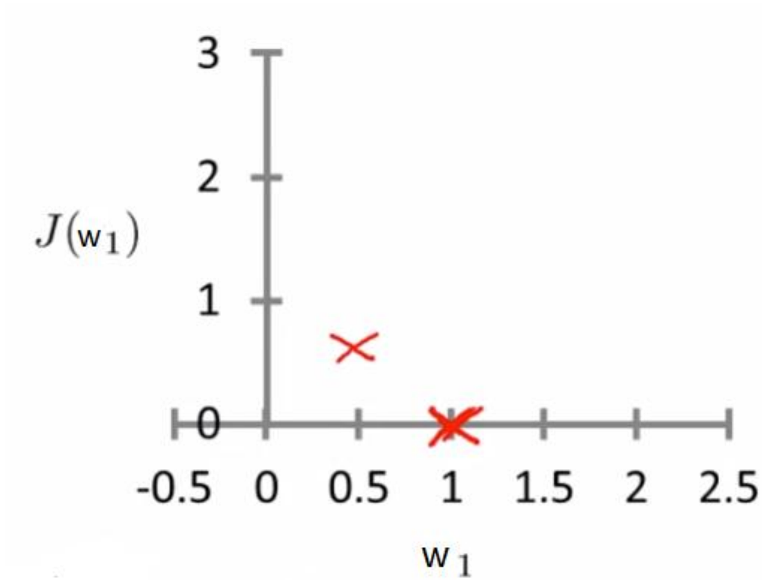
(For fixed w_1, this is function of x)



w_1 =0.5

Cost function: J(w1)

(function of parameter w1)



w_1 =0.5

J(0.5) =(1/2*3) [((0.5-1)^2)+((1-2)^2) + ((1.5-3)^2)]

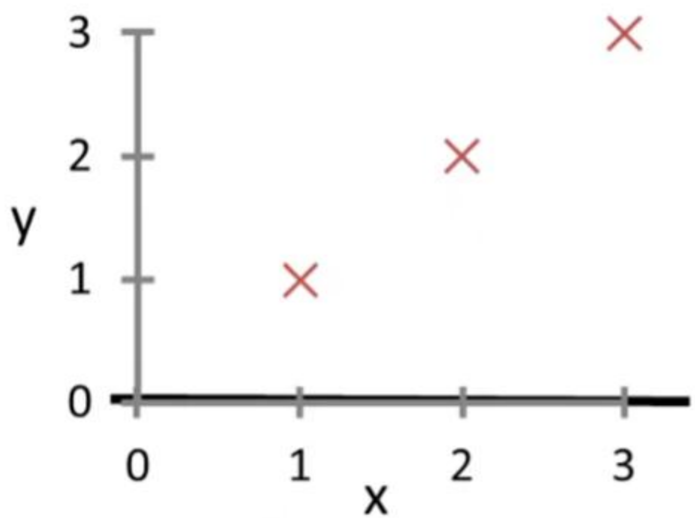
J(0.5)= (1/6) [(-0.5)^2 +(-1)^2 +(-1.5)^2]

J(0.5) =(1/6)[3.5] = 0.58

Visualization of Cost Function J

Hypothesis , $h_w(x)$

(For fixed w_1, this is function of x)

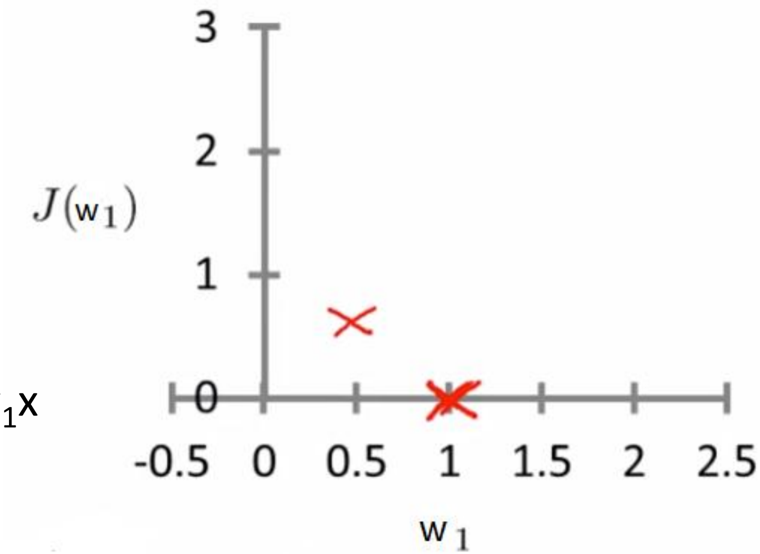


$h_w(x)=w_1x$

w_1 =0

Cost function: J(w1)

(function of parameter w1)



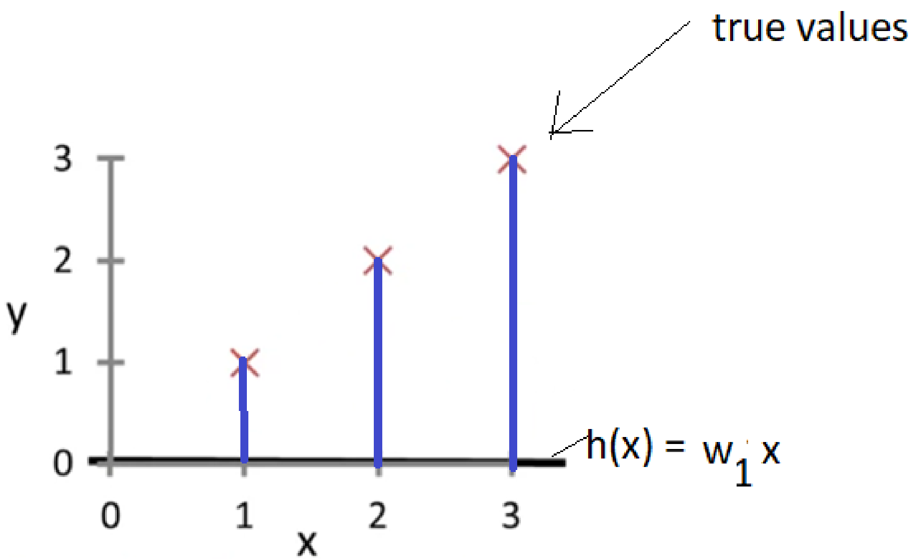
Visualization of Cost Function J

Hypothesis , $h_w(x)$

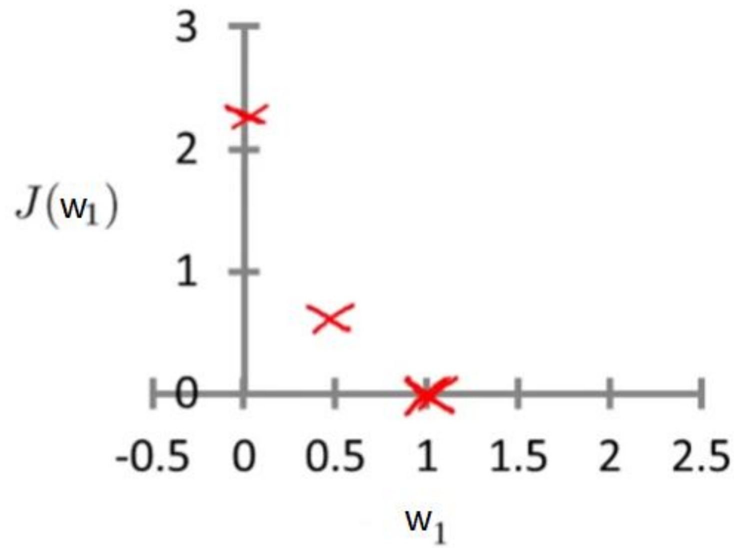
(For fixed w_1, this is function of x)

Cost function: J(w1)

(function of parameter w1)



w_1 =0



w_1 =0

J(0) =(1/2*3) [((-1)^2)+((0-2)^2) + ((0-3)^2)]

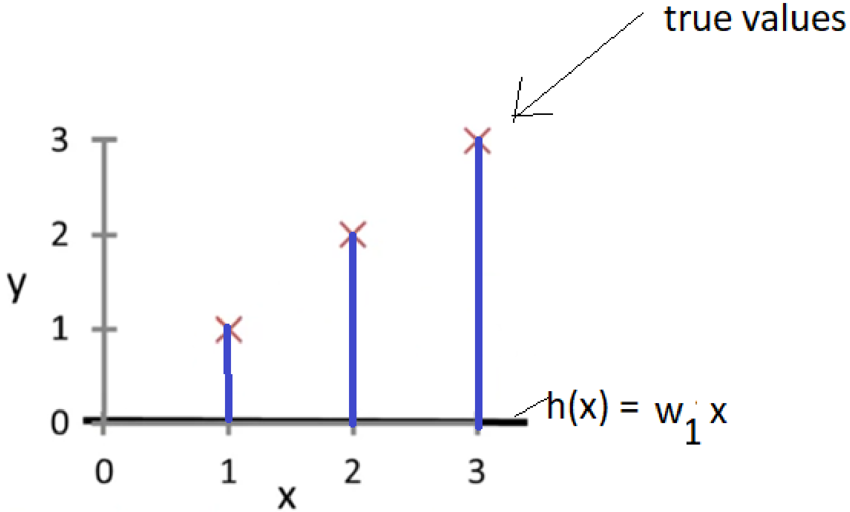
J(0)= (1/6) [1 + 4 + 9]

J(0) =(1/6) [14] = 14/6=2.33

Visualization of Cost Function J

Hypothesis , $h_w(x)$

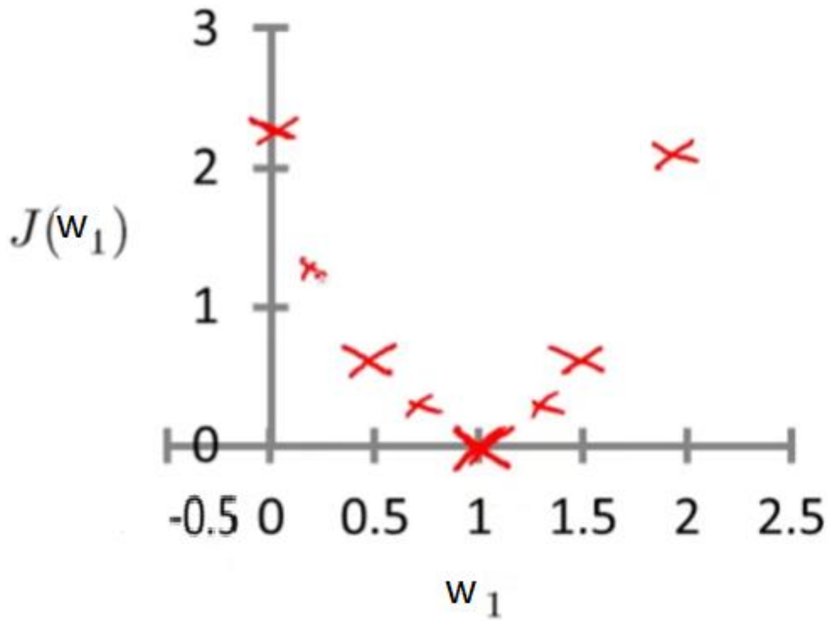
(For fixed w_1, this is function of x)



w_1 =0

Cost function: J(w1)

(function of parameter w1)

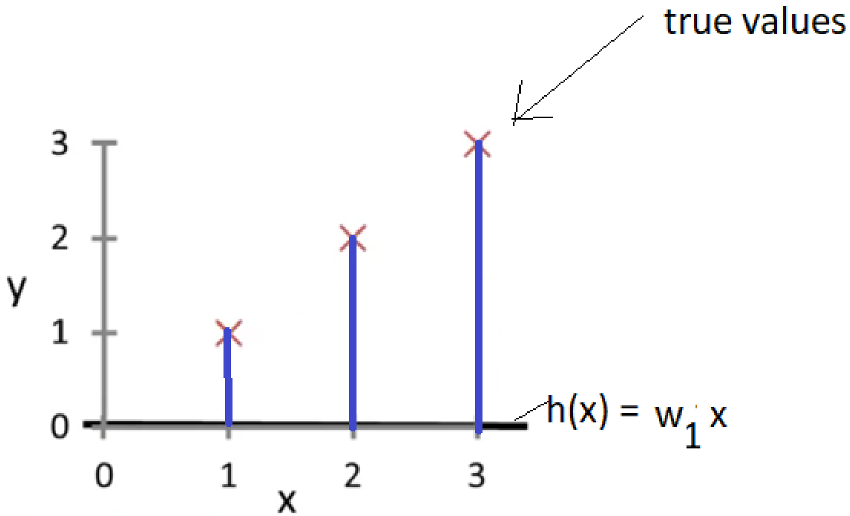


w_1 =1.5,2,2.5,…

Visualization of Cost Function J

Hypothesis , $h_w(x)$

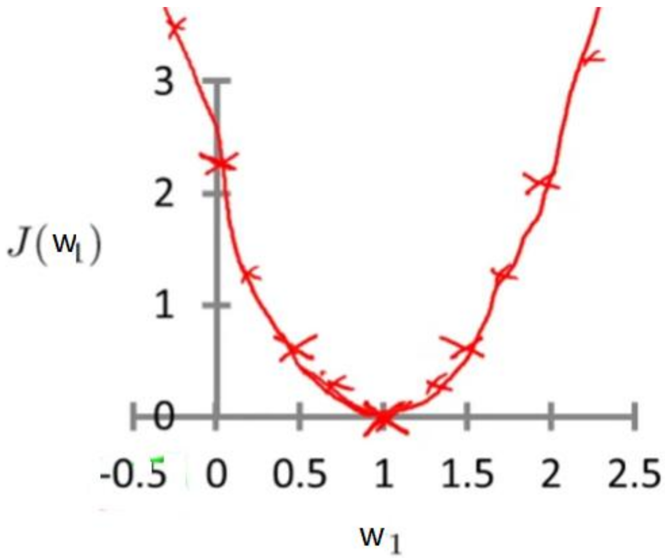
(For fixed w_1, this is function of x)



w_1 =0

Cost function: J(w1)

(function of parameter w1)



w_1 =1.5,2,2.5,…

minimize J_{w1}(W1)

Logistic Regression

- Logistic Regression is classification problem.

Want : 0<=h(x)<=1

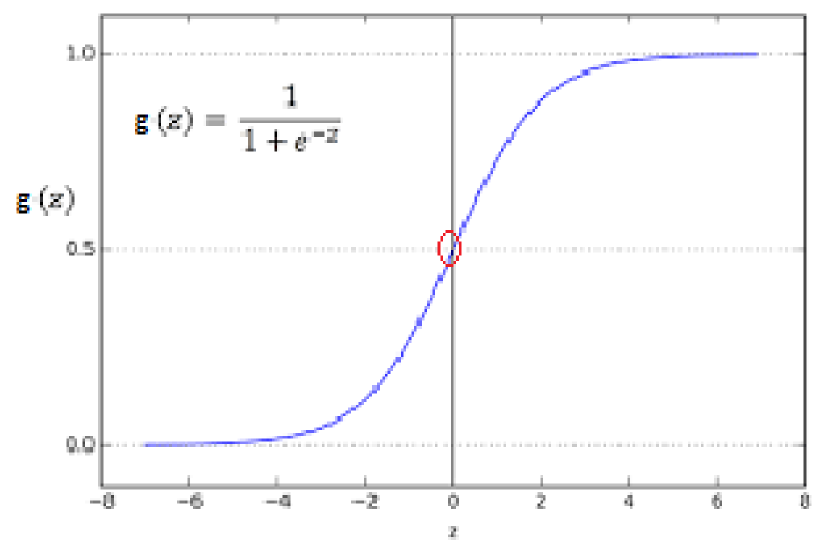
For linear regression: $h(x) = W^TX$

For logistic regression: $h(x) = g(W^TX) =$
$g(z)= 1/1+e^{-z}$ – this is logistic function

Therefore,

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

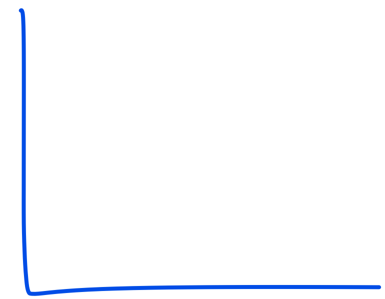
$$h(x) = \frac{1}{1+ e^{-W^TX}}$$



Hypothesis output

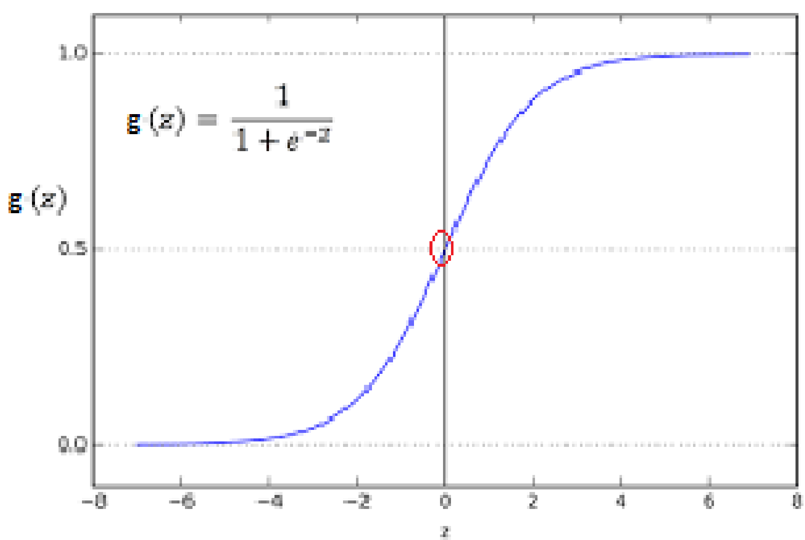
h(x) = estimated probability that y=1 on input x

$$h(x) = \frac{1}{1+e^{-w^T x}} = p(y=1/x,w)$$



Predict “y=1” if h(x) >=0.5

Predict “y=0” if h(x) < 0.5



When z>=0, g(z) >=0.5

i.e., $h(x)=g(w^T x) >= 0.5$ as $z=w^T x$

Loss Function

$$J(w) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Let, cost(h(x),y)= $\frac{1}{2}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2$

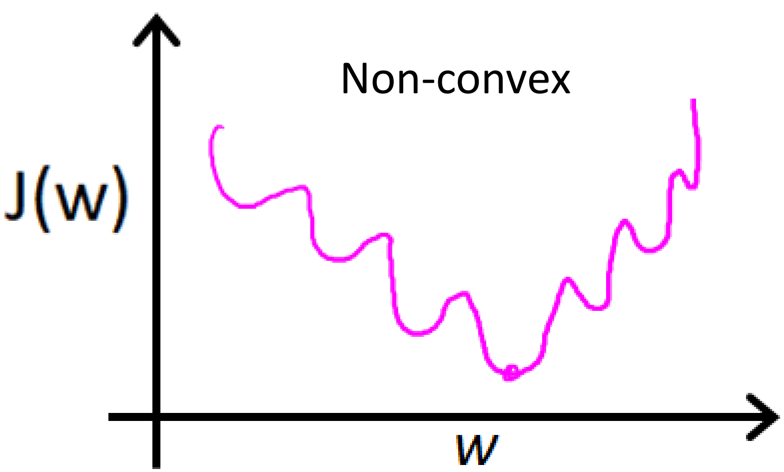
where, $h(x) = \dfrac{1}{1+ e^{-W^T X}}$

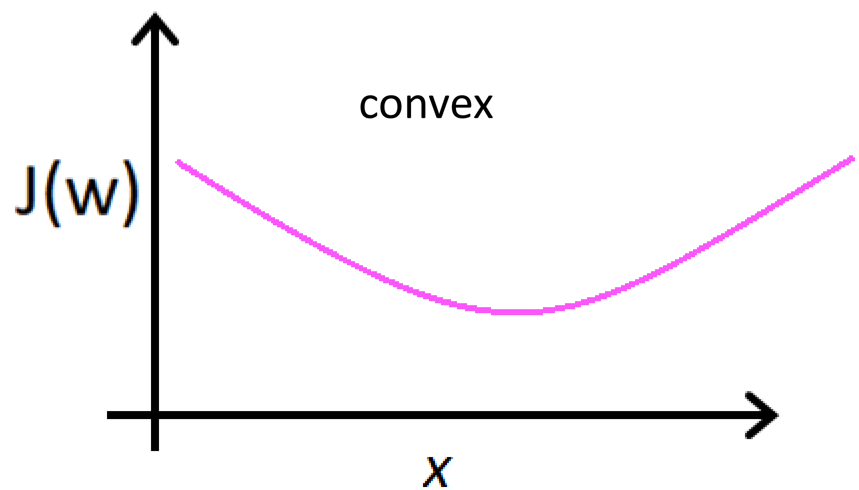
Loss/Objective/Error Function

$$J(w) = \frac{1}{m}\sum_{i=1}^{m}\mathrm{Cost}\left(h_w(x^{(i)}), y^{(i)}\right)$$

$$\mathrm{Cost}(h_w(x), y) = -\log(h_w(x)) \quad \text{if } y = 1$$

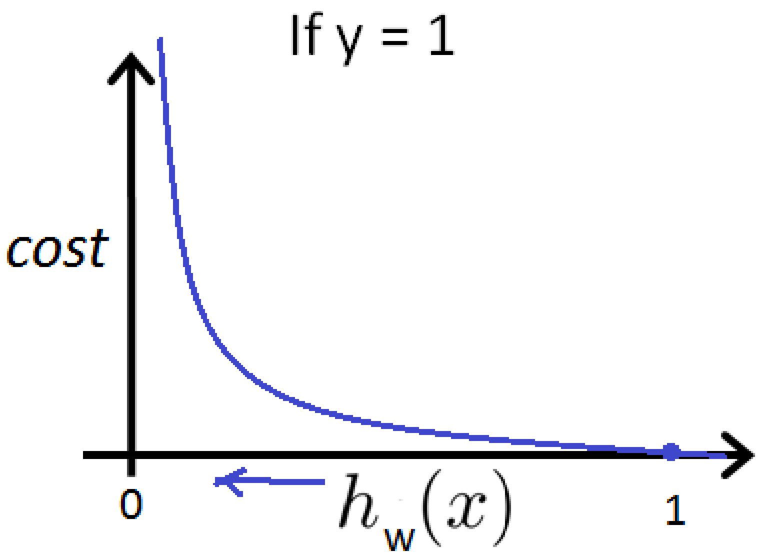
$$\mathrm{Cost}(h_w(x), y) = -\log(1 - h_w(x)) \quad \text{if } y = 0$$





Logistic regression cost function

$$\text{Cost}(h_w(x), y) = \begin{cases} -\log(h_w(x)) & \text{if } y = 1 \\ -\log(1 - h_w(x)) & \text{if } y = 0 \end{cases}$$



$$\text{Cost} = 0 \text{ if } y = 1, h_w(x) = 1$$

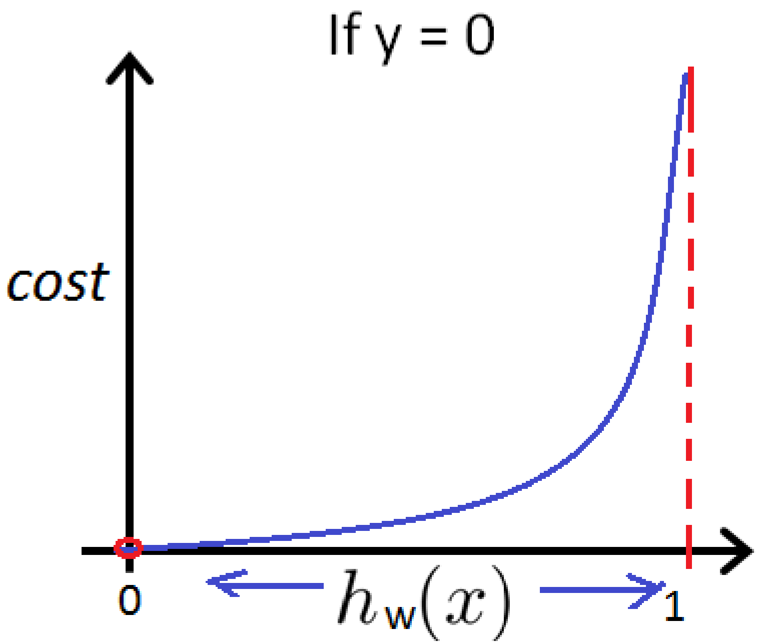
$$\text{But as} \quad h_w(x) \to 0$$

$$Cost \to \infty$$

Captures intuition that if $h_w(x) = 0$, (predict $P(y = 1 | x; \ ) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

Logistic regression cost function

$$\mathrm{Cost}(h_{\mathrm{w}}(x), y) = \begin{cases} -\log(h_{\mathrm{w}}(x)) & \text{if } y = 1 \\ -\log(1 - h_{\mathrm{w}}(x)) & \text{if } y = 0 \end{cases}$$



$$\mathrm{Cost} = \infty \text{ if } y = 0, h_{\mathrm{w}}(x) = 1$$
$$\text{But as } \quad h_{\mathrm{w}}(x) \rightarrow 0$$
$$Cost \rightarrow 0$$

Captures intuition that if $h_{\mathrm{w}}(x) = 1$, (predict $P(y=0|x;\mathrm{w}) = 1$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

Simplified Logistic regression cost function

$$\mathrm{Cost}(h\,(x), y) = \begin{cases} -\log(h\ \ (x)) & \text{if } y = 1 \\ -\log(1 - h\ \ (x)) & \text{if } y = 0 \end{cases}$$

$$\text{J(w)} = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h(x^{(i)}), y^{(i)})$$

$$\mathrm{Cost}(h(x^{(i)}), y^{(i)}) \ = - \ y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log (1 - h\ (x^{(i)}))$$

$$\text{J(w)} = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log (1 - h\ (x^{(i)}))$$

Problem on Linear Regression

- Given the following data set. Using linear regression, estimate the target variable y as a function of the input feature x. The hypothesis is $h_w(x) = w_0 + w_1(x)$

X	Y
2	3
3	4
4	5
5	6
6	7

1. Given w parameter , find the ones which will best fit the data: i) $w_0 = 1, w_1 = 0.5$
   ii) $w_0 = 1, w_1 = 1.5$
   iii) $w_0 = 1.5, w_1 = 1.$
2. Plot the hypothesis for best w parameters and give the value of the cost function for the same, which is mean square error function?
3. Use answer of (1) to evaluate $h_w(x = 8)$

Solution

$w_0 = 1$ and $w_1 = 0.5$				
X	**Y**	**h(X) = w_0+w_1*X**	**h(X)-Y**	**(h(X)-Y)^2**
2	3	1 + 0.5 * 2 = 2	2 -3 = -1	(-1)^2 = 1
3	4	1 + 0.5 * 3 = 2.5	2.5 – 4 = -1.5	(-1.5)^2 = 2.25
4	5	1 + 0.5 * 4= 3	3 – 5 = -2	(-2)^2 = 4
5	6	1 + 0.5 * 5 = 3.5	3.5 – 6 = -2.5	(-2.5)^2 = 6.25
6	7	1 + 0.5 * 6 = 4	4 – 7 = -3	(-3)^2 = 9
				Σ(h(X)-Y)^2 = 22.5

Cost Function: J(w0,w1) = $\frac{1}{2m}\sum_{i=1}^{m}\left(h_{\mathrm{w}}(x_i) - y_i\right)^2$

J(1,0.5) = (1/2*5) *22.5 = (1/10) * 22.5 = 2.25

Solution

$w_0 = 1$ and $w_1 = 1.5$				
X	**Y**	**h(X) = w_0+w_1*X**	**h(X)-Y**	**(h(X)-Y)^2**
2	3	1 + 1.5 * 2 = 4	4 -3 = 1	(1)^2 = 1
3	4	1 + 1.5 * 3 = 5.5	5.5 – 4 = 1.5	(1.5)^2 = 2.25
4	5	1 + 1.5 * 4= 7	7 – 5 = 2	(2)^2 = 4
5	6	1 + 1.5 * 5 = 8.5	8.5 – 6 = 2.5	(2.5)^2 = 6.25
6	7	1 + 1.5 * 6 = 10	10 – 7 = 3	(3)^2 = 9
				Σ(h(X)-Y)^2 = 22.5

Cost Function: J(w0,w1) = $\frac{1}{2m}\sum_{i=1}^{m}\left(h_{\mathrm{w}}(x_i)-y_i\right)^2$

J(1,1.5) = (1/2*5) *22.5 = (1/10) * 22.5 = 2.25

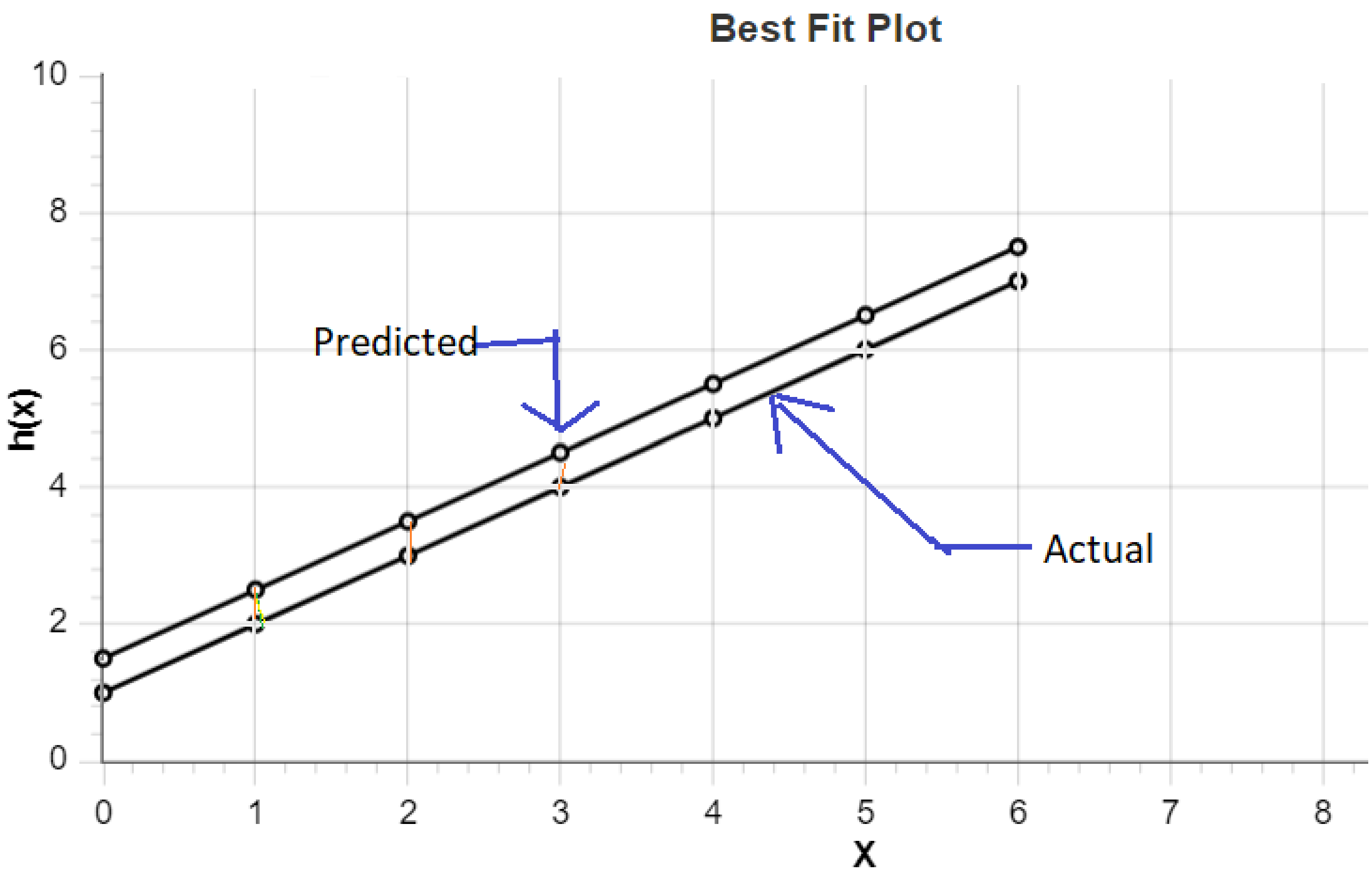
Solution

w_0 = 1.5 and w_1 =1				
X	**Y**	**h(X) = w_0+w_1*X**	**h(X)-Y**	**(h(X)-Y)^2**
2	3	1.5 + 1 * 2 = 3.5	3.5 - 3 = 0.5	(0.5)^2 = 0.25
3	4	1.5 + 1 * 3 = 4.5	4.5 – 4 = 0.5	(0.5)^2 = 0.25
4	5	1.5 + 1 * 4= 5.5	5.5 – 5 = 0.5	(0.5)^2 = 0.25
5	6	1.5 + 1 * 5 = 6.5	6.5 – 6 = 0.5	(0.5)^2 = 0.25
6	7	1.5 + 1 * 6 = 7.5	7.5 – 7 = 0.5	(0.5)^2 = 0.25
				Σ(h(X)-Y)^2 = 1.25

Cost Function: J(w0,w1) = $\frac{1}{2m}\sum_{i=1}^{m}\left(h_{\mathrm{w}}(x_i) - y_i\right)^2$

J(1.5,1) = (1/2*5) *1.25 = (1/10) * 1.25 = 0.125

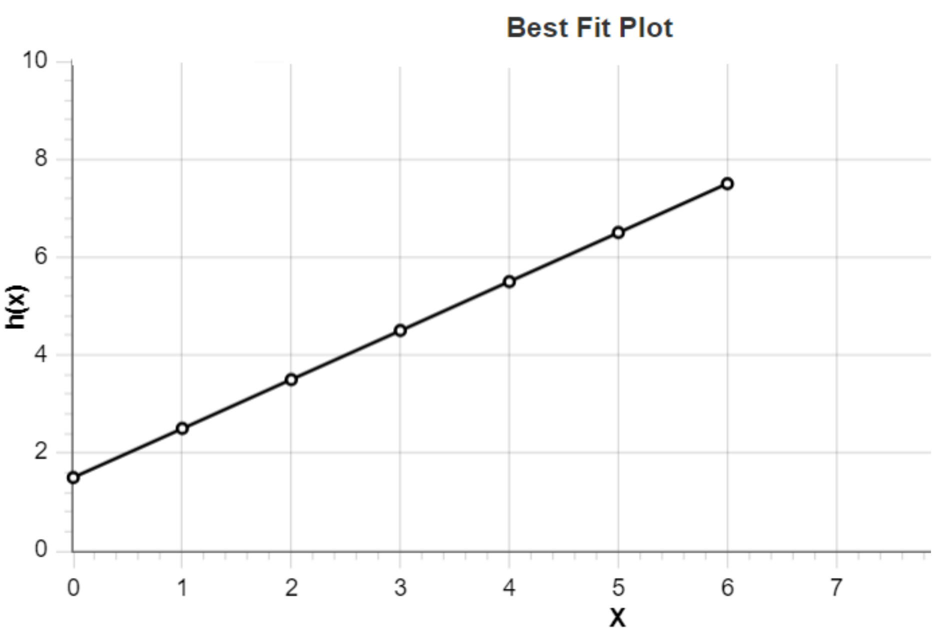
Plot of Difference between true and predicted values for w_0=1.5 and w_1=1



Solution

w_0	w_1	$J(w_0, w_1)$
1	0.5	2.25
1	1.5	2.25
1.5	**1**	**0.125**

Cost is minimum for the weight values w_0=1.5 and w_1=1. these are the parameters which best fit the data. Plot is:



Gradient Descent

Have some function $J(\theta_0, \theta_1)$

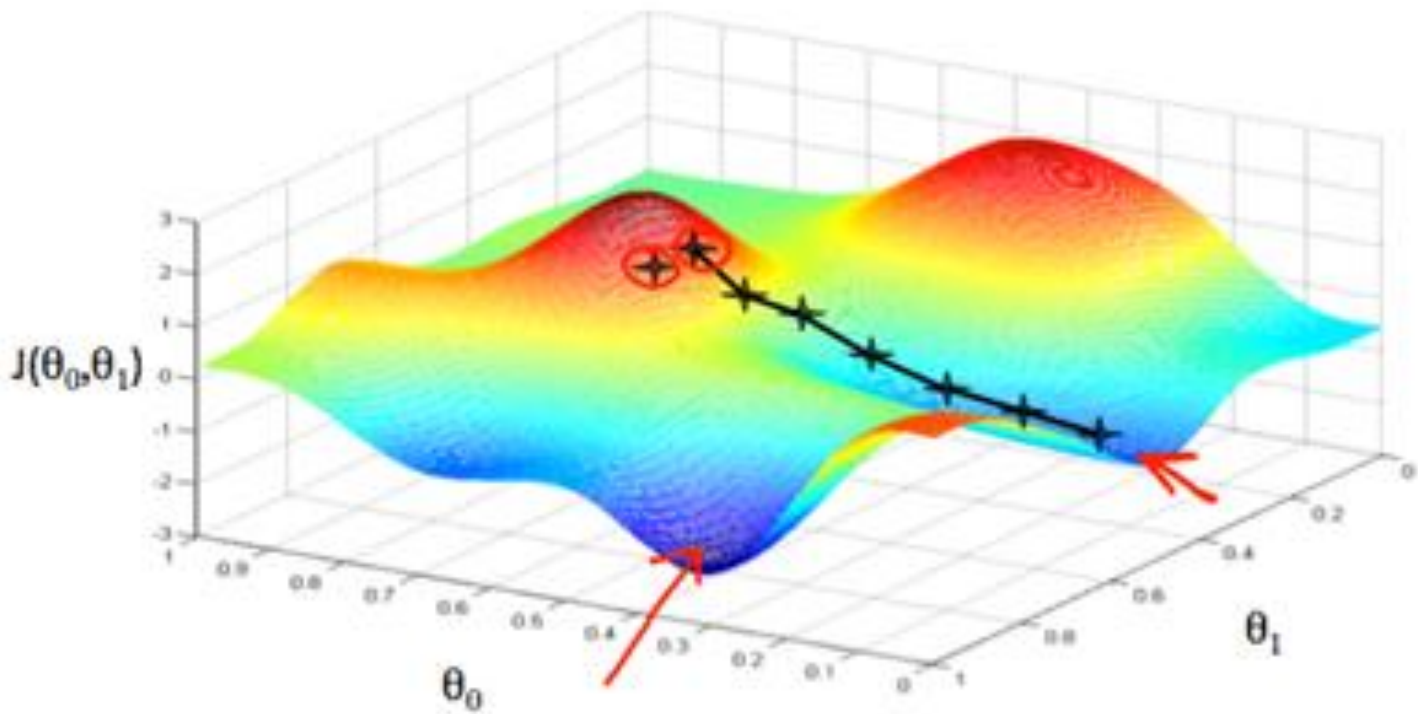
Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

  until we hopefully end up at a minimum

Gradient Descent



Gradient descent algorithm

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad (\text{for } j = 0 \text{ and } j = 1) \\
&\}
\end{aligned}
$$

Correct: Simultaneous update

$$
\begin{aligned}
&\text{temp0} := \theta_0 - \alpha \tfrac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) \\
&\text{temp1} := \theta_1 - \alpha \tfrac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) \\
&\theta_0 := \text{temp0} \\
&\theta_1 := \text{temp1}
\end{aligned}
$$

Incorrect:

$$
\begin{aligned}
&\text{temp0} := \theta_0 - \alpha \tfrac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) \\
&\theta_0 := \text{temp0} \\
&\text{temp1} := \theta_1 - \alpha \tfrac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) \\
&\theta_1 := \text{temp1}
\end{aligned}
$$

Linear regression with one variable
Gradient Descent

Gradient descent algorithm

$$\text{repeat until convergence } \{$$
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$
$$(\text{for } j = 1 \text{ and } j = 0)$$
$$\}$$

Linear Regression Model

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta_0 ,\theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta_0 ,\theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x^{(i)} - y^{(i)})^2$$

$$j = 0 : \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})$$

$$j = 1 : \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})\, x^{(i)}$$

Gradient descent algorithm

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\qquad \left.\begin{aligned}
\theta_0 &:= \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \\
\theta_1 &:= \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)}
\end{aligned}\right\} \\
&\}
\end{aligned}
$$

update θ_0 and θ_1 simultaneously