Applied Artificial Intelligence

Session 10: Linear Models

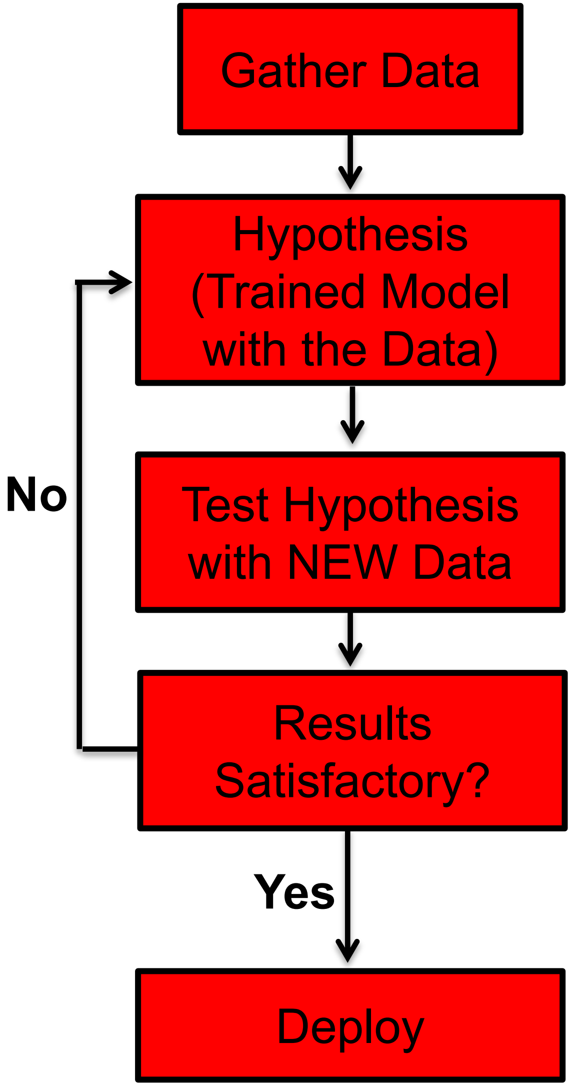
Fall 2018

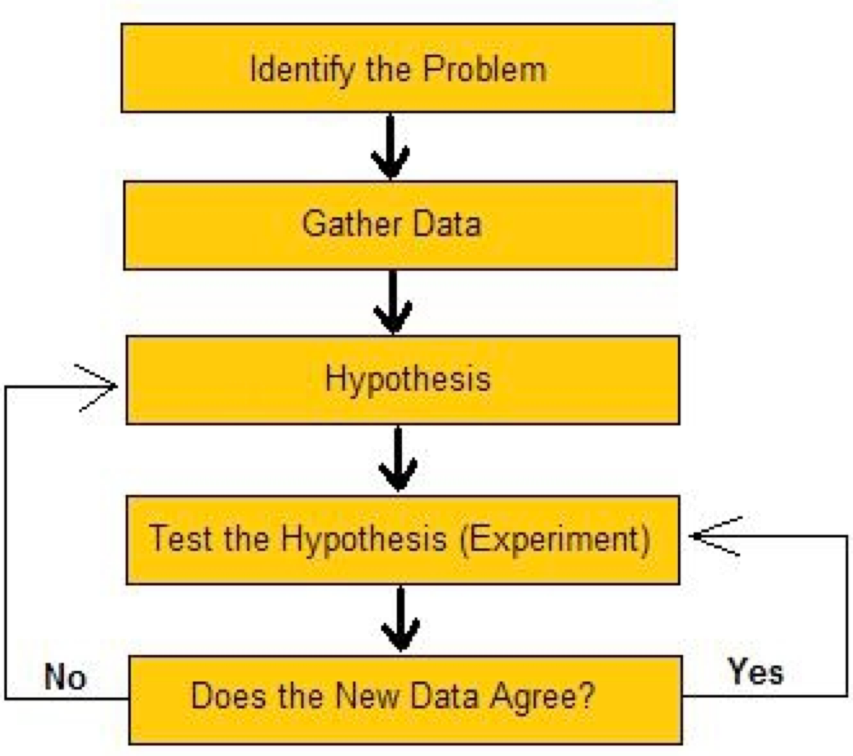
NC State University

Lecturer: Dr. Behnam Kia

Course Website: https://appliedai.wordpress.ncsu.edu/

Machine Learning Flowchart (Which Follows Scientific Method)





Scientific Method

Regression Problem and Regression Models

- In Regression problems the task is to approximate a mapping function (*h*) from input variables (*x*) to continuous output variables, (also called real-valued outputs) , (also called numeric outputs).

- We design Regression Models that learn from the training data to perform this task.

  Training data: (x_1,y_1), (x_2,y_2), (x_3,y_3),…, (x_n,y_n)

Regression Problem and Regression Models

- In Regression problems the task is to approximate a mapping function (*h*) from input variables (*x*) to continuous output variables, (also called real-valued outputs) , (also called numeric outputs).

- In this session we work on linear regression models:

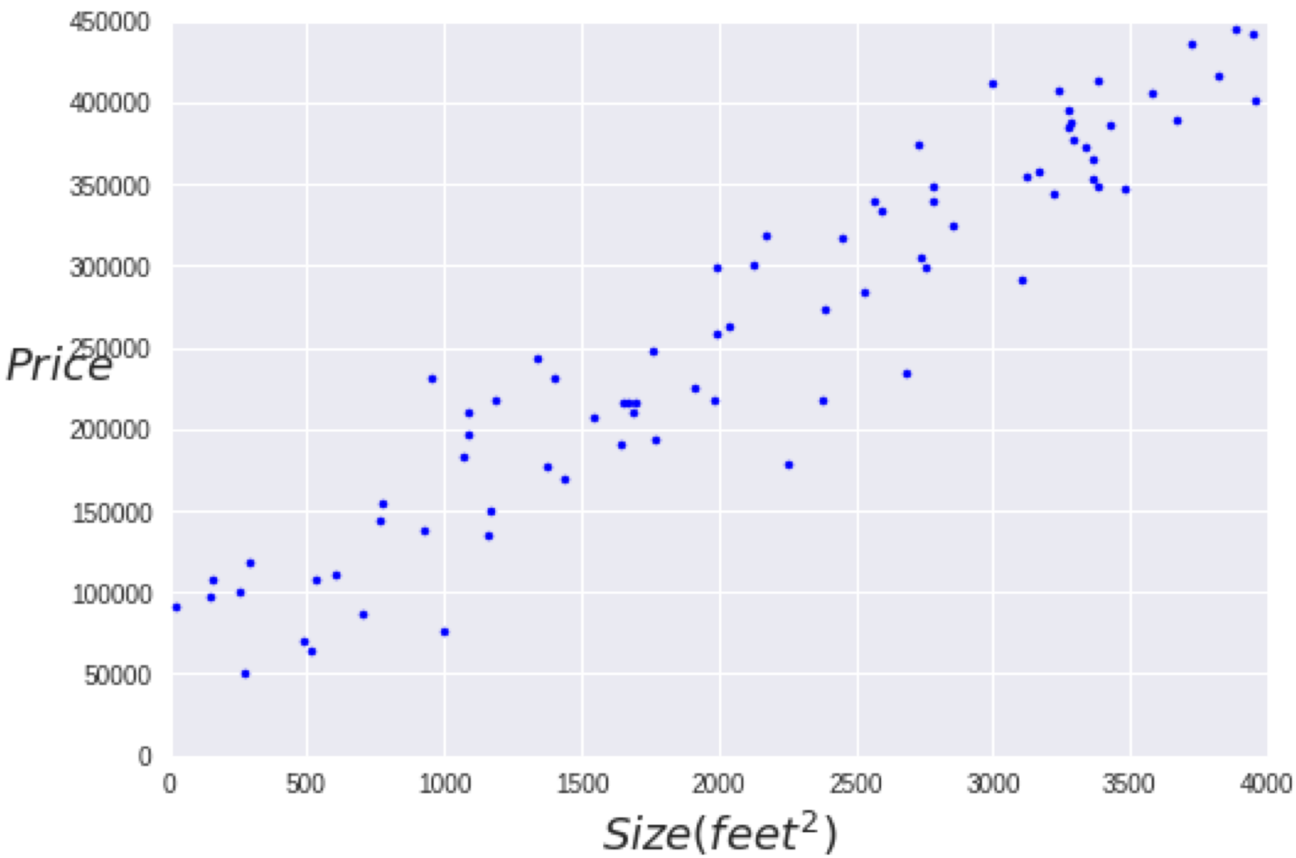
$$\hat{y} = h(X) = W^T X$$

Training data: (x_1,y_1), (x_2,y_2), (x_3,y_3),…, (x_n,y_n)

House Prices
Goal: Come Up with a Linear Regression Model that Explains the Training Data, and Predicts the Price of Other Houses as Well.

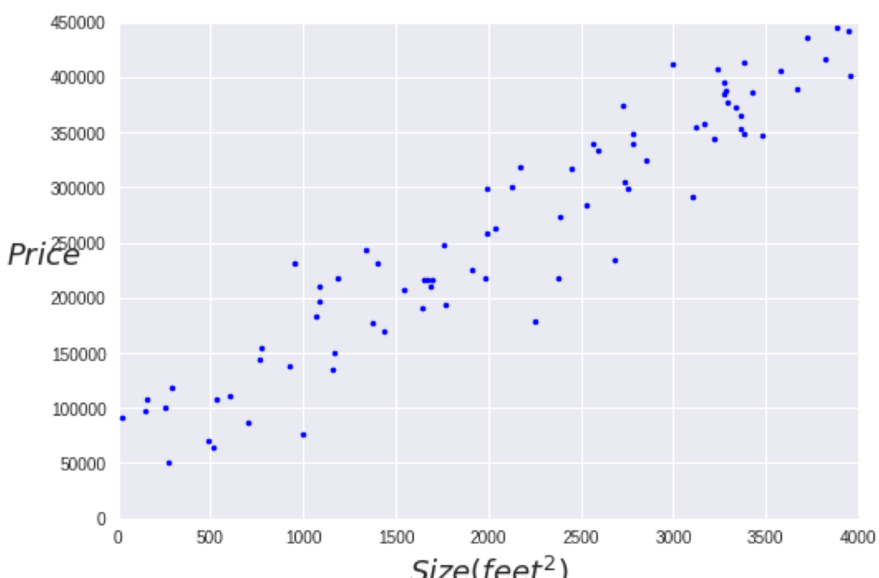
(x ft^2,$ y K)

(3883 ft^2,$432K)
(1668 ft^2,$218K)
(3577 ft^2,$366K)
(1668 ft^2,$218K)
(765 ft^2,$123K)
(3822 ft^2,$493K)
.
.
.



House Prices
Goal: Come Up with a Linear Regression Model that Explains the Training Data, and Predicts the Price of Other Houses as Well.

Input: x (the size of house)
Target: y (the price of the house), $y \in R$
Linear Model: $\hat{y} = w_0 + w_1 x_1$



House Prices
Goal: Obtaining a Linear Regression Model

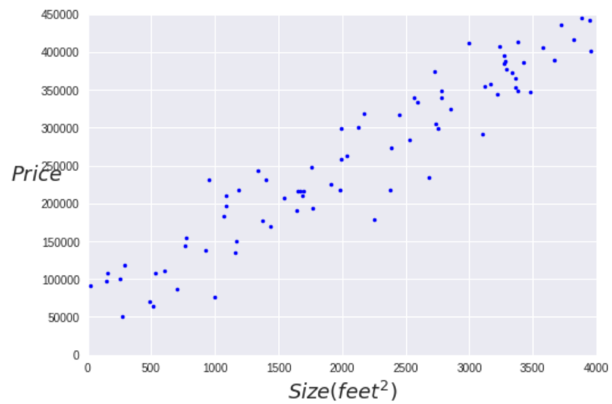
Input: x (the size of house)

Target: y (the price of the house), y$\in R$

Linear Model: $\hat{y} = w_0 + w_1 x_1$

Bias value

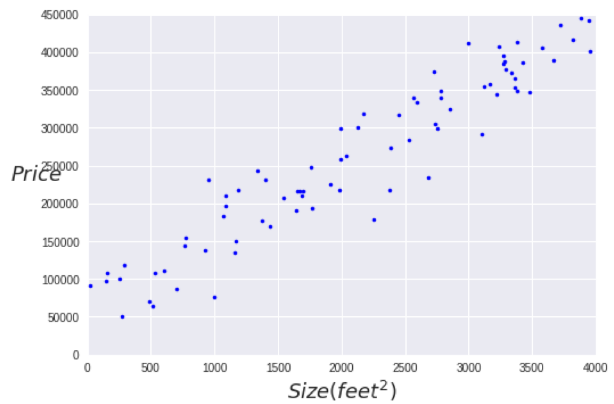
Coefficient



House Prices
Goal: Obtaining a Linear Regression Model

Input: x (the size of house)
Target: y (the price of the house), $y \in R$
Linear Model: $\hat{y} = w_0 x_0 + w_1 x_1$

Constant feature x_0=1



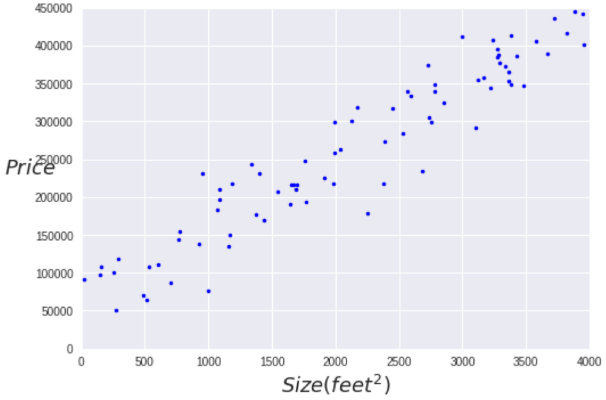
House Prices
Goal: Obtaining a Linear Regression Model

Input: x (the size of house)

Target: y (the price of the house), y$\in R$

Linear Model: $\hat{y} = w_0 x_0 + w_1 x_1$

$$\hat{y} = W^T . x_p$$



where:

$$W^T = [w_0, w_1]$$

$$x_p = \begin{bmatrix} 1 \\ x_1 \end{bmatrix}$$

House Prices
Goal: Obtaining a Linear Regression Model

Input: x (the size of house)

Target: y (the price of the house), y∈ R

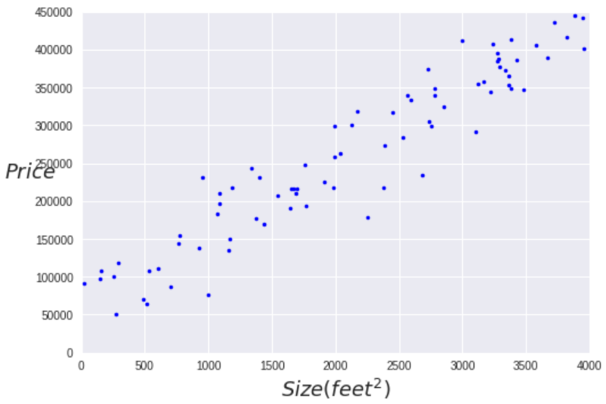
Linear Model: $\hat{y} = w_0 x_0 + w_1 x_1$

$$\hat{y} = W^T . x_p$$

where:

W^T=$[w_0, w_1]$?

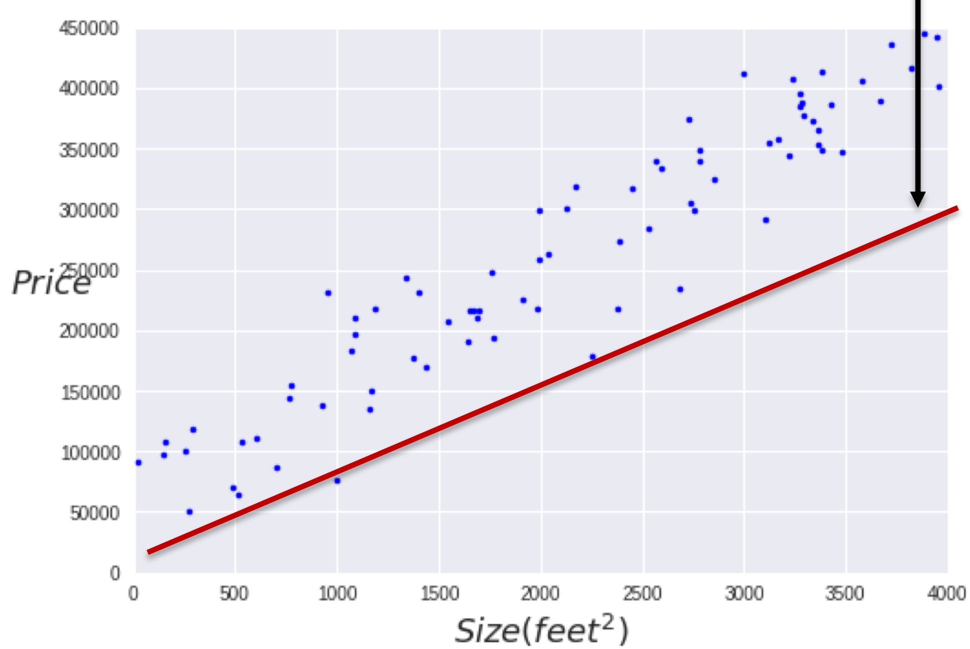
$$x_p = \begin{bmatrix} 1 \\ x_1 \end{bmatrix}$$



House Prices
Cost Function: How well or poorly a model (Hypothesis) explains the training data

(x ft^2,$ y K)
(3883 ft^2,$432K)
(1668 ft^2,$218K)
(3577 ft^2,$366K)
(765 ft^2,$123K)
(3822 ft^2,$493K)
(1668 ft2, $218K)
.
.
.

W^T=[w_0, w_1] ?

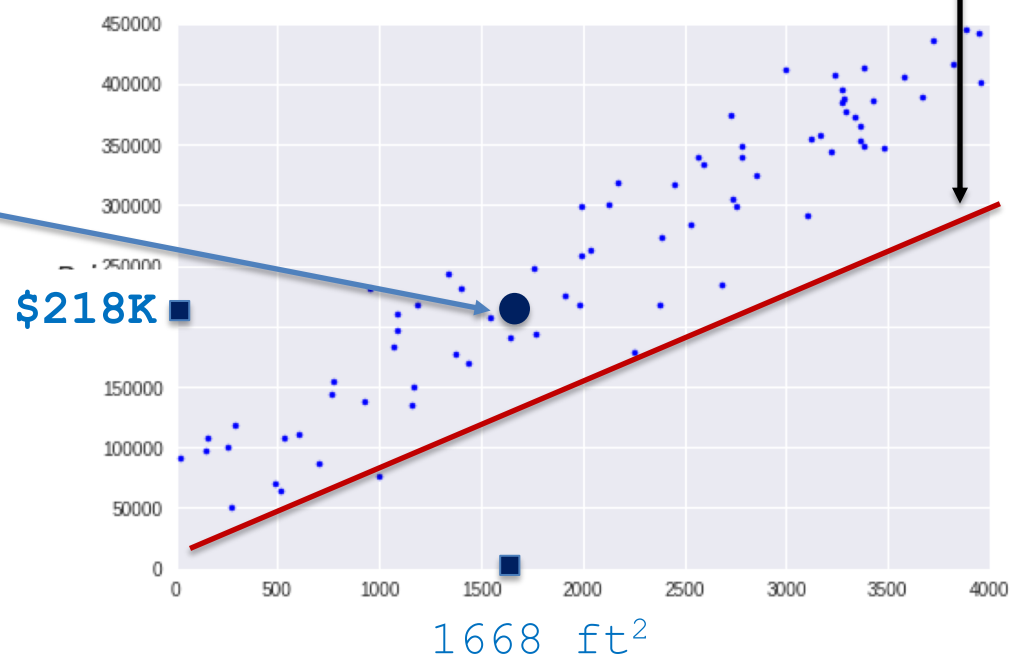


House Prices
Cost Function: How well or poorly a model (Hypothesis) explains the training data

(x ft², $ y K)
(3883 ft², $432K)
(1668 ft², $218K)
(3577 ft², $366K)
(765 ft², $123K)
(3822 ft², $493K)
(1668 ft², $218K)

.
.
.

$W^T = [w_0, w_1]$ **?**



House Prices
Cost Function: How well or poorly a model (Hypothesis) explains the training data

(x ft², $ y K)

(3883 ft², $432K)

(1668 ft², $218K)

(3577 ft², $366K)

(765 ft², $123K)

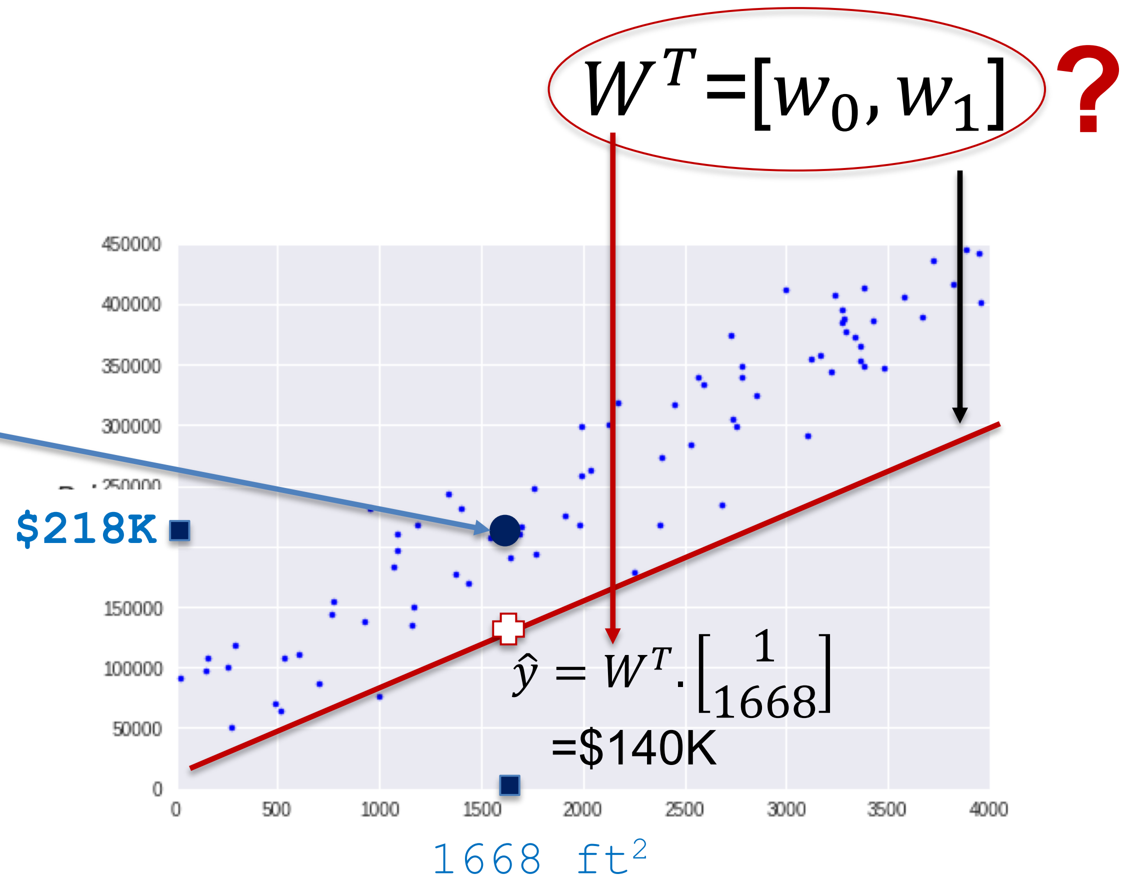
(3822 ft², $493K)

(1668 ft², $218K)

.

.

.



House Prices
Cost Function: How well or poorly a model (Hypothesis) explains the training data

(x ft^2, $ y K)

(3883 ft^2, $432K)

(1668 ft^2, $218K)

(3577 ft^2, $366K)

(765 ft^2, $123K)

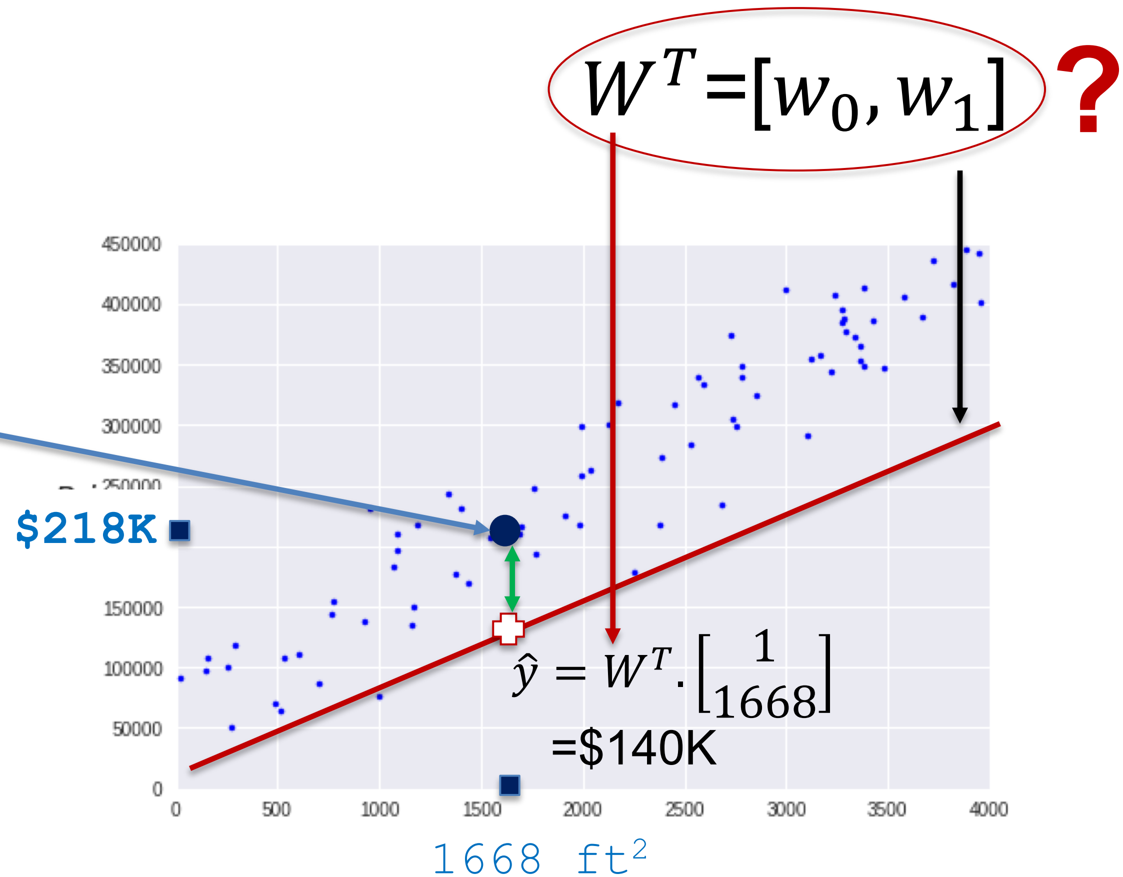
(3822 ft^2, $493K)

(1668 ft^2, $218K)

.

.

.



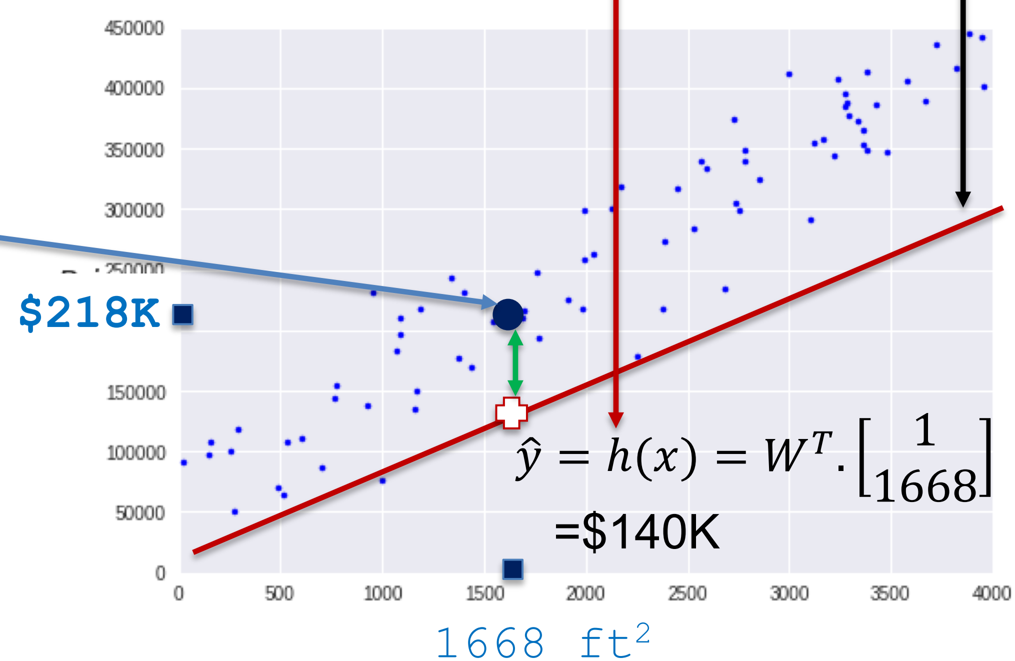
House Prices
Cost Function: How well or poorly a model (Hypothesis) explains the training data

(x ft², $ y K)
(3883 ft², $432K)
(1668 ft², $218K)
(3577 ft², $366K)
(765 ft², $123K)
(3822 ft², $493K)
(1668 ft², $218K)

Cost Function for input x^i:

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

$$W^T=[w_0, w_1]\ ?$$

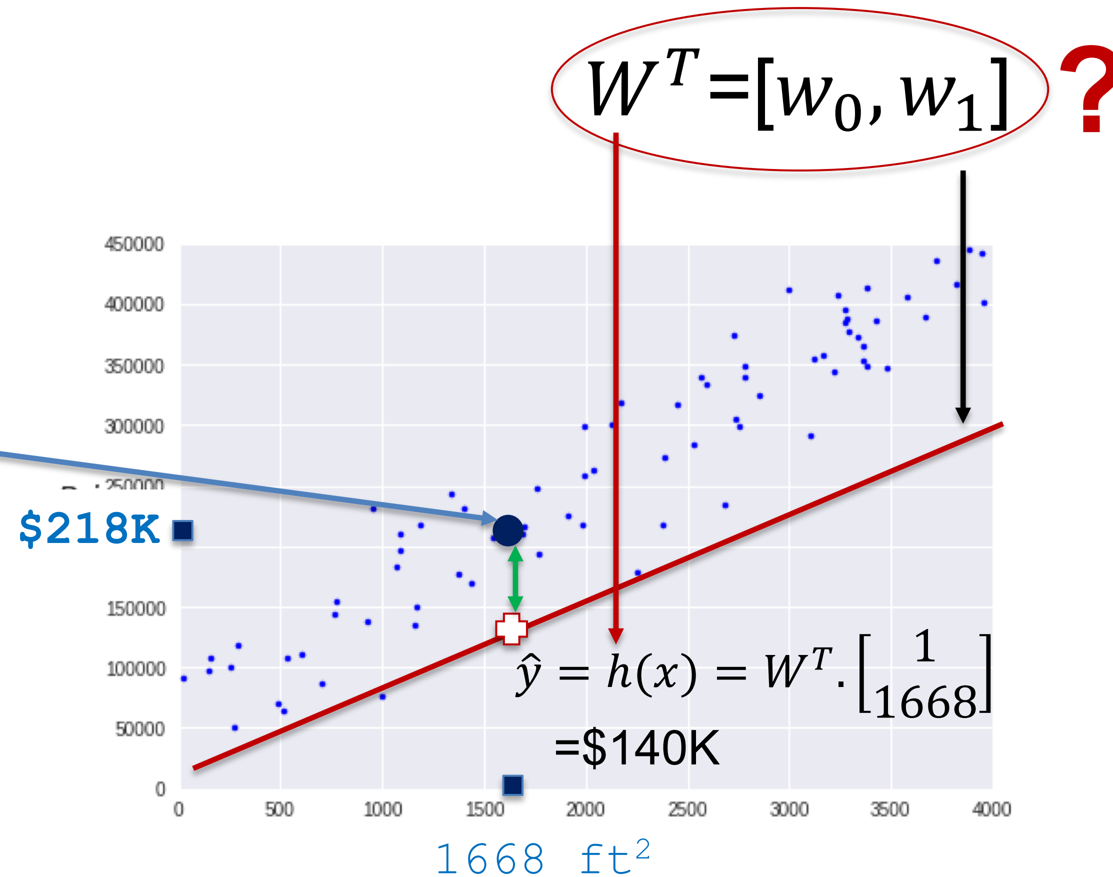


House Prices
Cost Function: How well or poorly a model (Hypothesis) explains the training data

(x ft², $ y K)
(3883 ft², $432K)
(1668 ft², $218K)
(3577 ft², $366K)
(765 ft², $123K)
(3822 ft², $493K)
(1668 ft², $218K)

Cost Function for entire training set:

$$J(W)=\sum_i \left(h(\hat{x}^i) - y^i\right)^2$$



House Prices
Cost Function: How well or poorly a model (Hypothesis) explains the training data

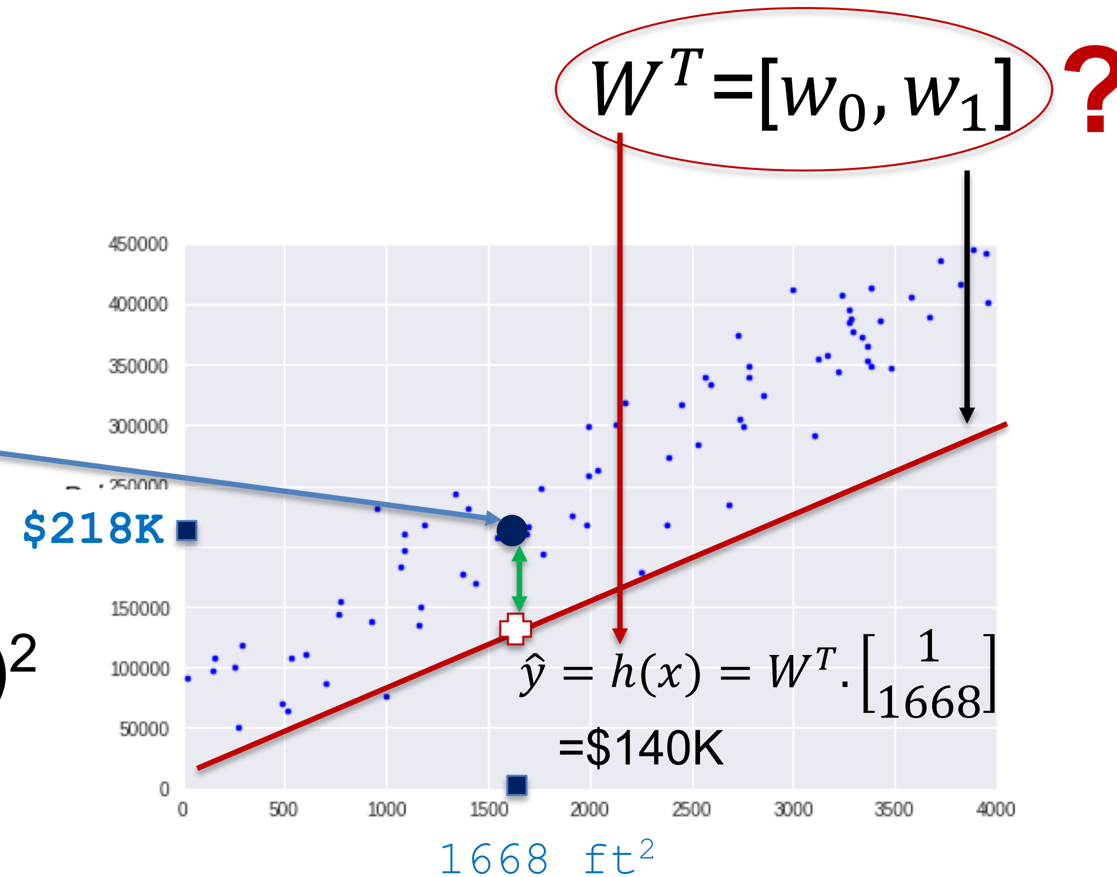
(x ft², $ y K)

(3883 ft², $432K)
(1668 ft², $218K)
(3577 ft², $366K)
(765 ft², $123K)
(3822 ft², $493K)
(1668 ft², $218K)

Cost Function for entire training set:

$$J(W)=\frac{1}{n}\sum_i \left(h(\hat{x}^i)-y^i\right)^2$$

Size of training data



House Prices
Cost Function: How well or poorly a model (Hypothesis) explains the training data

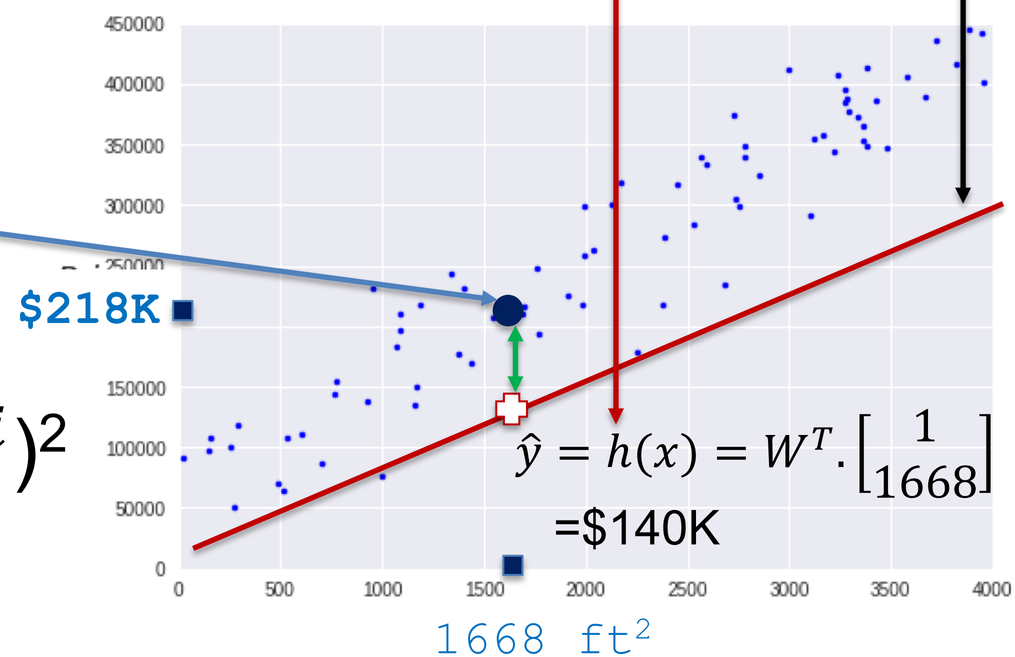
(x ft², $ y K)
(3883 ft², $432K)
(1668 ft², $218K)
(3577 ft², $366K)
(765 ft², $123K)
(3822 ft², $493K)
(1668 ft², $218K)

W^T=$[w_0, w_1]$ **?**

Cost Function for entire training set:

$$J(W) = \frac{1}{2n} \sum_i \left(h(\hat{x}^i) - y^i\right)^2$$

Size of training data



House Prices
Cost Function: How well or poorly a model (Hypothesis) explains the training data

Cost Function for entire training set:

$$J(W) = \frac{1}{2n} \sum_i \left(h(\hat{x}^i) - y^i\right)^2$$

Therefore the training problem is reduced to an optimization problem to find parameters W that minimizes the cost function

$$\begin{aligned} W_{optimal} &= \underset{W}{\operatorname{argmin}}\; J(W) \\ &= \underset{W}{\operatorname{argmin}} \frac{1}{2n} \sum_i \left(h(\hat{x}^i) - y^i\right)^2 \\ &= \underset{W}{\operatorname{argmin}} \frac{1}{2n} \sum_i \left(W^T . x_p^i - y^i\right)^2 \end{aligned}$$

How to find the minimum point of cost function?

$$W_{optimal} = \underset{W}{\operatorname{argmin}}\ \mathrm{J(W)}$$

?

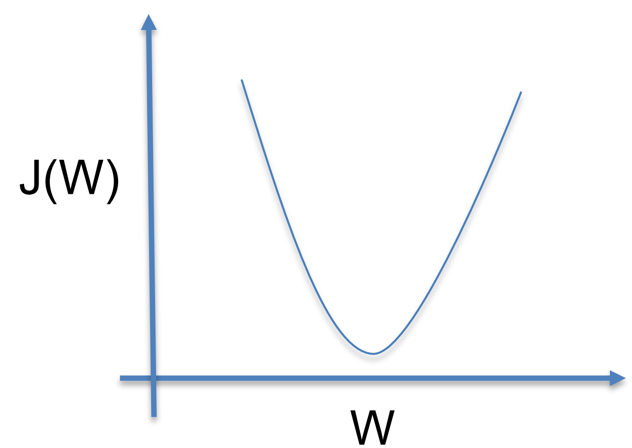
$$\mathrm{J(W)} = \frac{1}{2n}\sum_i (W^T . x_p^i - y^i\ )^2$$

How to find the minimum point of cost function? Analytical Method

$$W_{optimal} = \underset{W}{\operatorname{argmin}}\ \mathrm{J(W)}$$

$$\mathrm{J(W)} = \frac{1}{2n} \sum_i (W^T . x_p^i - y^i\ )^2$$

$$\frac{\partial J(W)}{\partial W} = 0$$



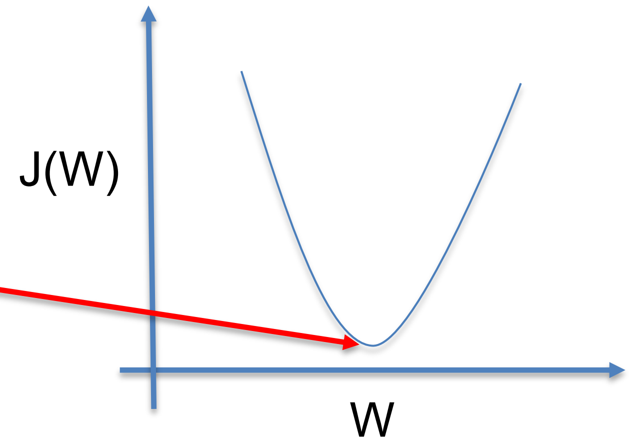
Solving this will give us the optimal W

How to find the minimum point of cost function? Analytical Method

$$W_{optimal} = \underset{W}{\operatorname{argmin}} \; \mathrm{J(W)}$$

$$\mathrm{J(W)} = \frac{1}{2n} \sum_i (W^T . x_p^i - y^i)^2$$

$$\frac{\partial J(W)}{\partial W} = 0$$



Solving this will give us the optimal W

How to find the minimum point of cost function? Analytical Method

$$W_{optimal} = \underset{W}{\operatorname{argmin}}\ \mathrm{J(W)}$$

$$\mathrm{J(W)} = \frac{1}{2n}\sum_i (W^T . x_p^i - y^i\ )^2$$

$$\frac{\partial J(W)}{\partial W} = 0 \Longrightarrow \mathrm{W_{optimal}} = (X_p^T . X_p)^{-1} .\ X_p^T .\mathrm{y}$$

How to find the minimum point of cost function? Analytical Method

$$W_{optimal} = \underset{W}{\operatorname{argmin}}\ \mathrm{J(W)}$$

$$\mathrm{J(W)} = \frac{1}{2n}\sum_i (W^T . x_p^i - y^i)^2$$

(n,1+1)

$$X_p = \begin{array}{c} x_p^1 \\ \begin{bmatrix} 1 & 3883 \\ 1 & 1668 \\ 1 & 3577 \\ \vdots & \vdots \\ 1 & 1668 \end{bmatrix} \\ x_p^n \end{array}$$

$$\frac{\partial J(W)}{\partial W} = 0 \Rightarrow W_{optimal} = (X_p^T . X_p)^{-1} . X_p^T . y$$

(n,1)

$$y = \begin{bmatrix} 432K \\ 218K \\ 366K \\ \vdots \\ 218K \end{bmatrix} \begin{array}{c} y^1 \\ \\ \\ \\ y^n \end{array}$$

How to find the minimum point of cost function? Analytical Method

$$W_{optimal} = \underset{W}{\operatorname{argmin}} \ \mathrm{J(W)}$$

$$\mathrm{J(W)} = \frac{1}{2n} \sum_i (W^T . x_p^i - y^i\ )^2$$

(n,1+1)

$$X_p = \begin{array}{c} \begin{bmatrix} 1 & 3883 \\ 1 & 1668 \\ 1 & 3577 \\ \vdots & \vdots \\ 1 & 1668 \end{bmatrix} \end{array} \begin{array}{l} x_p^1 \\ \\ \\ \\ x_p^n \end{array}$$

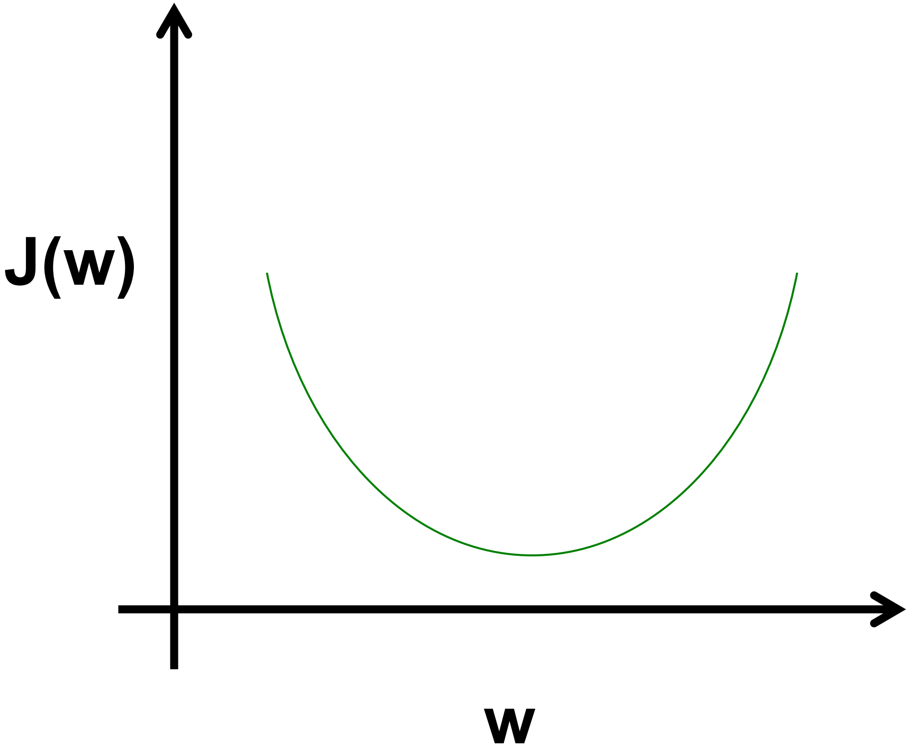
(n,1)

$$y = \begin{bmatrix} 432K \\ 218K \\ 366K \\ \vdots \\ 218K \end{bmatrix} \begin{array}{l} y^1 \\ \\ \\ \\ y^n \end{array}$$

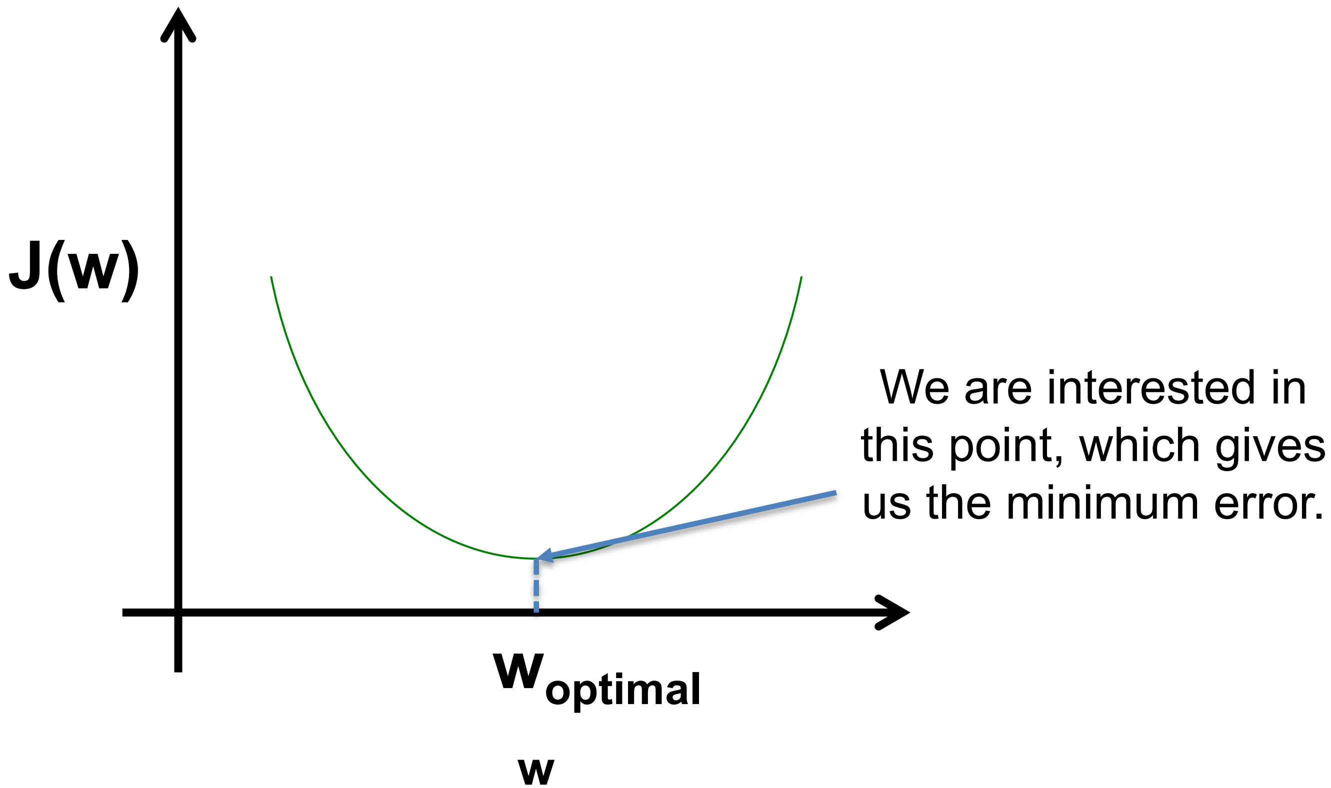
$$\frac{\partial J(W)}{\partial W} = 0 \Rightarrow \mathrm{W_{optimal}} = (X_p^T . X_p)^{-1} . \ X_p^T . \mathrm{y}$$

The Elements of Statistical Learning, T. Hastie, R. Tibshirani, J. Friedman
Page 12, and pages 44-45

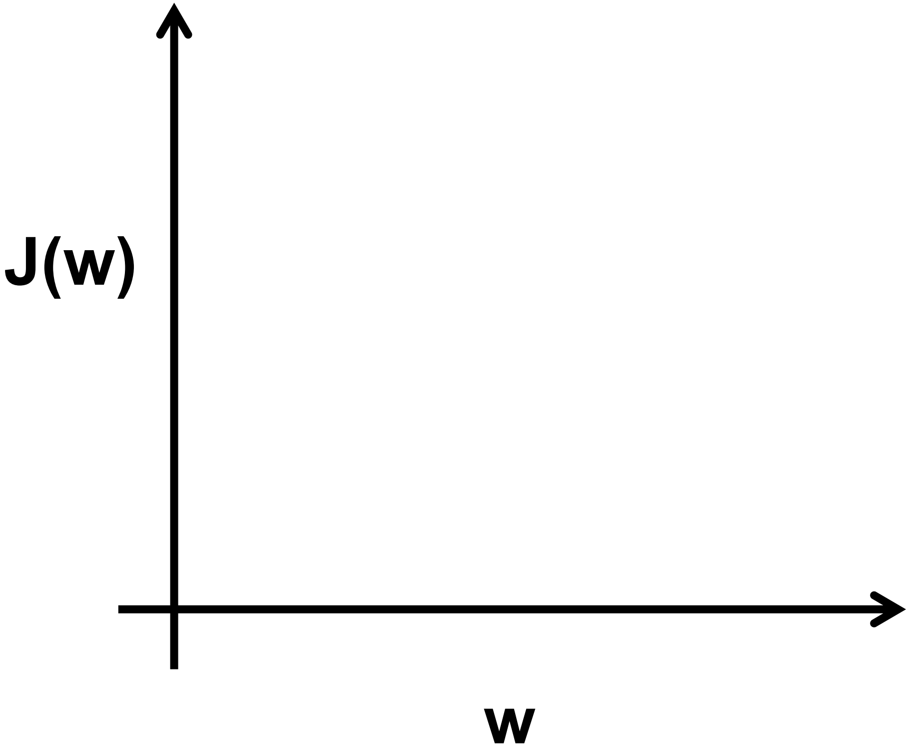
Gradient Descent



Gradient Descent

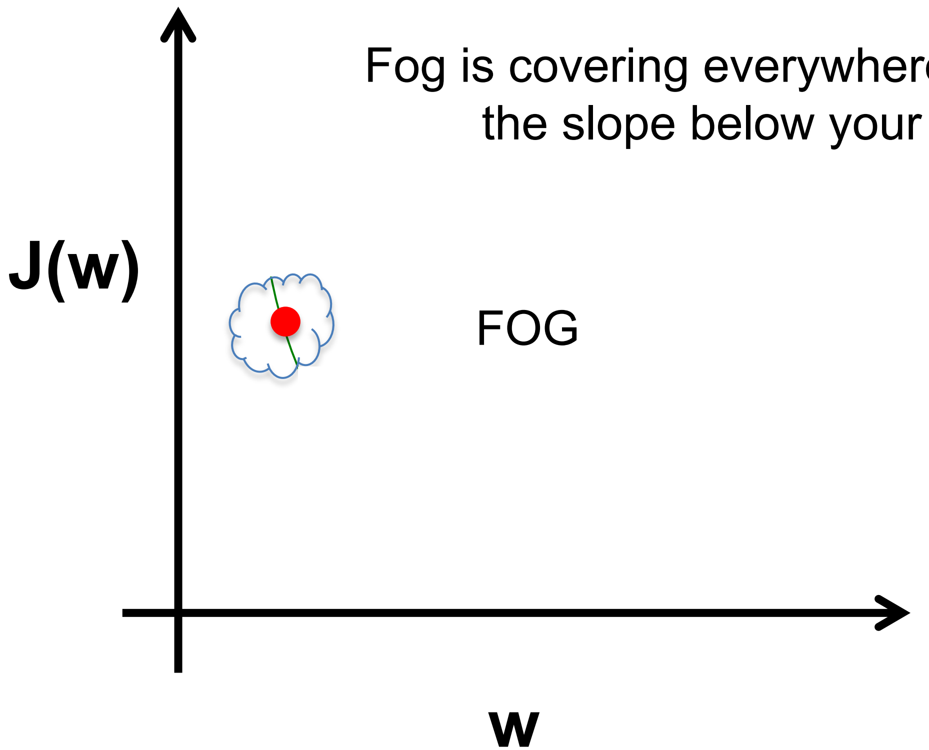


Gradient Descent

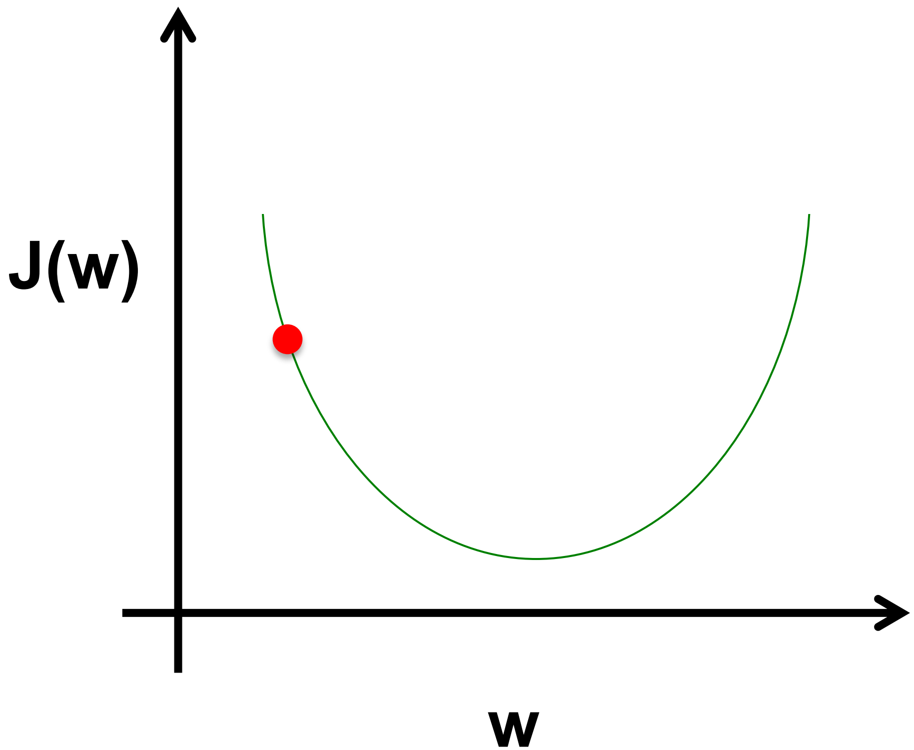


Gradient Descent

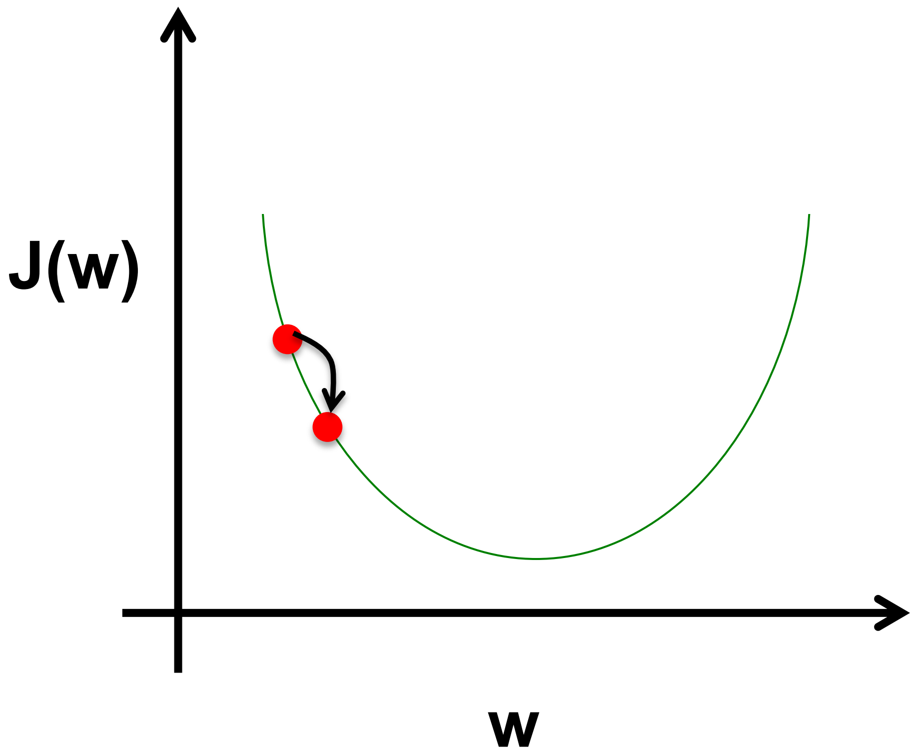
Fog is covering everywhere, you see the slope below your feet.



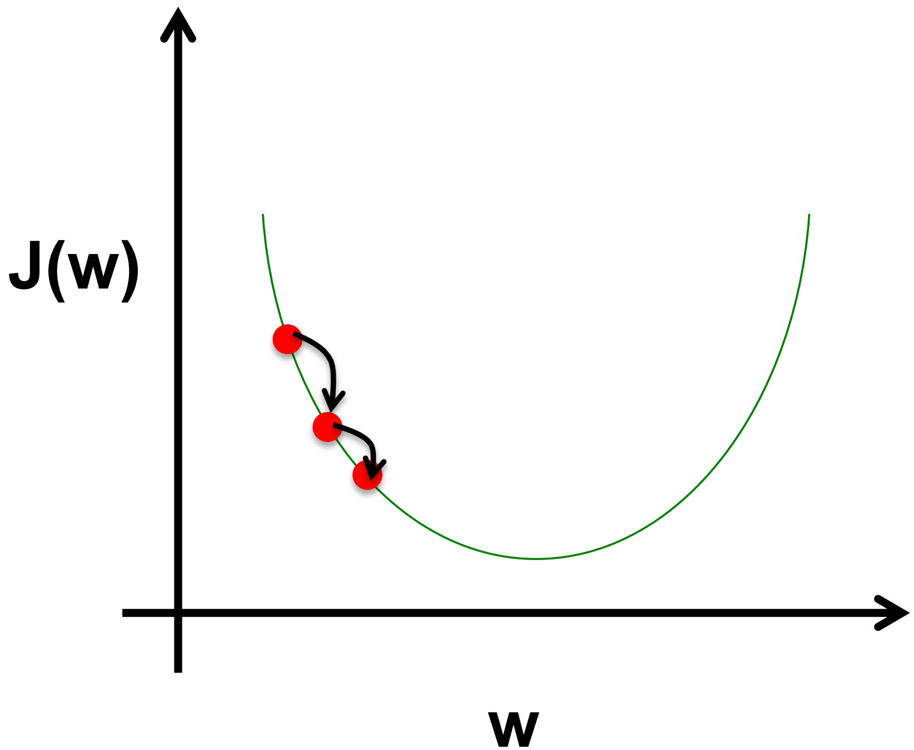
Gradient Descent



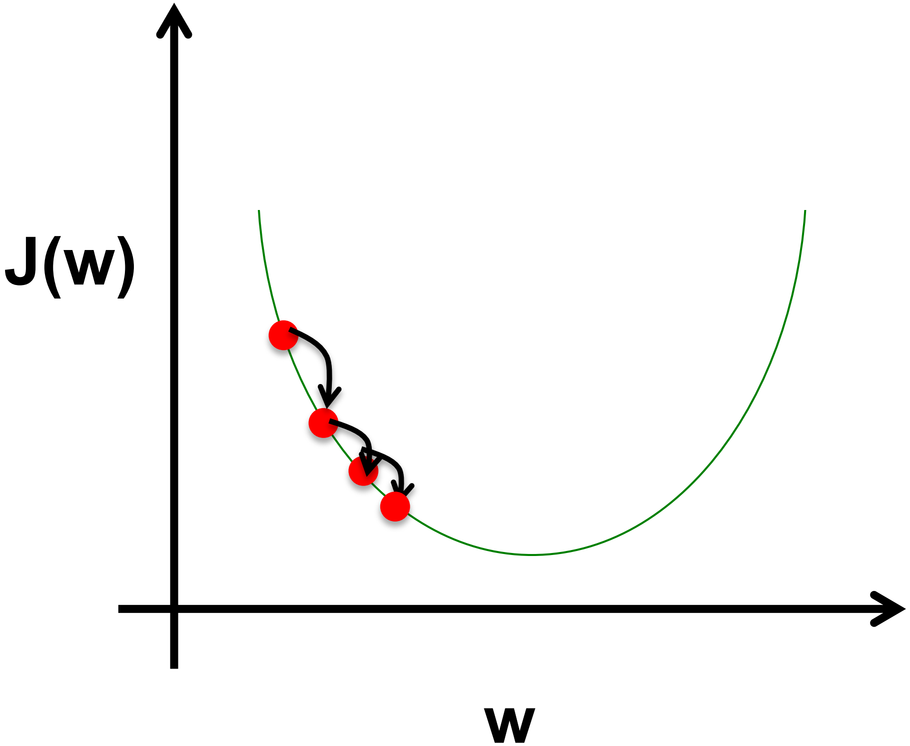
Gradient Descent



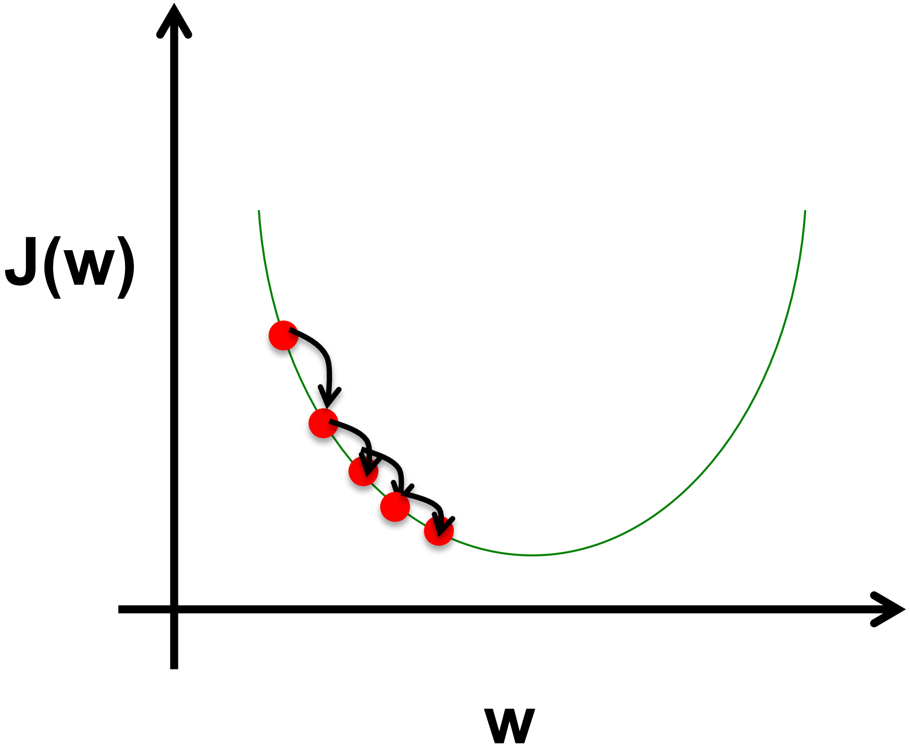
Gradient Descent



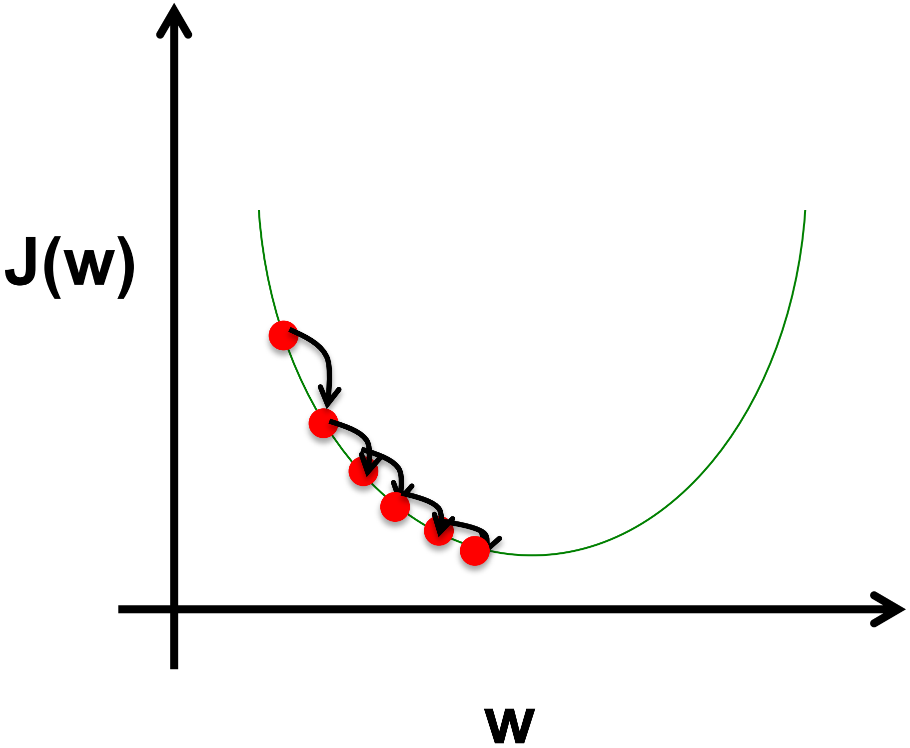
Gradient Descent



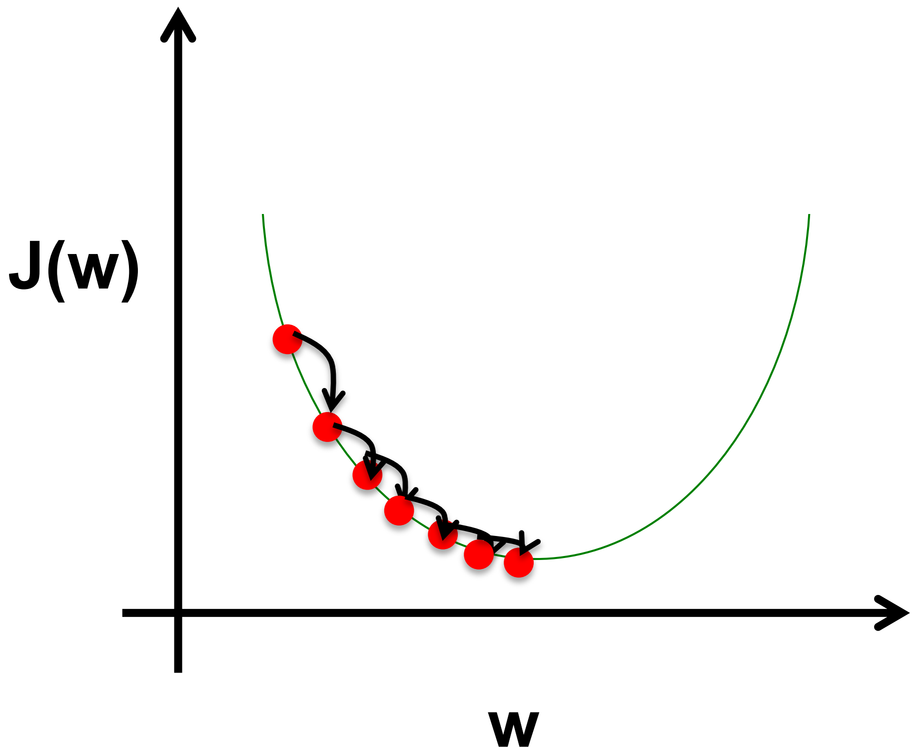
Gradient Descent



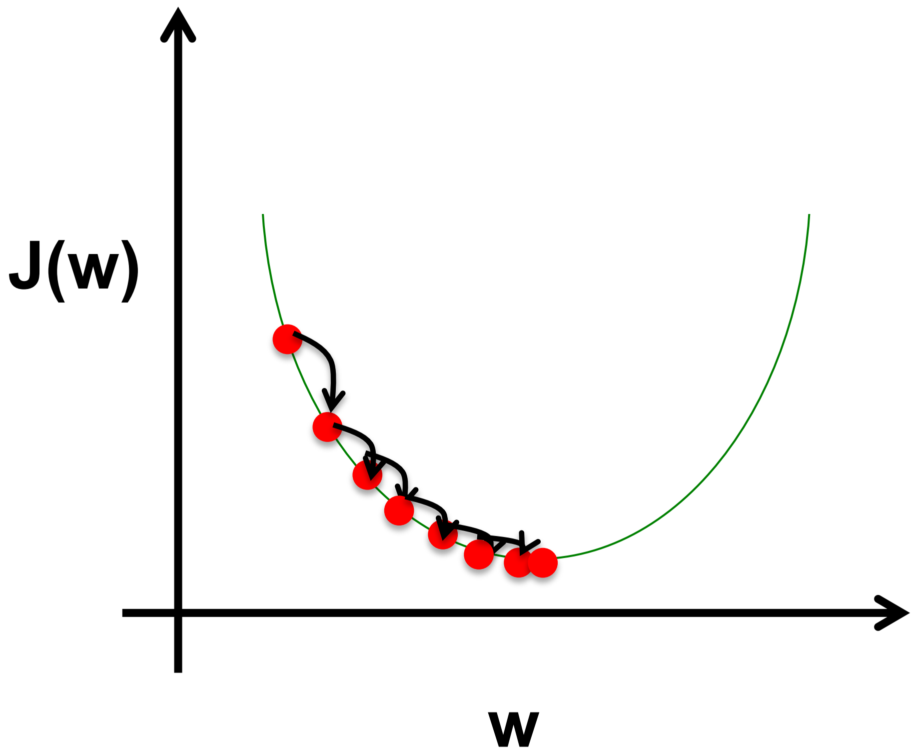
Gradient Descent



Gradient Descent



Gradient Descent

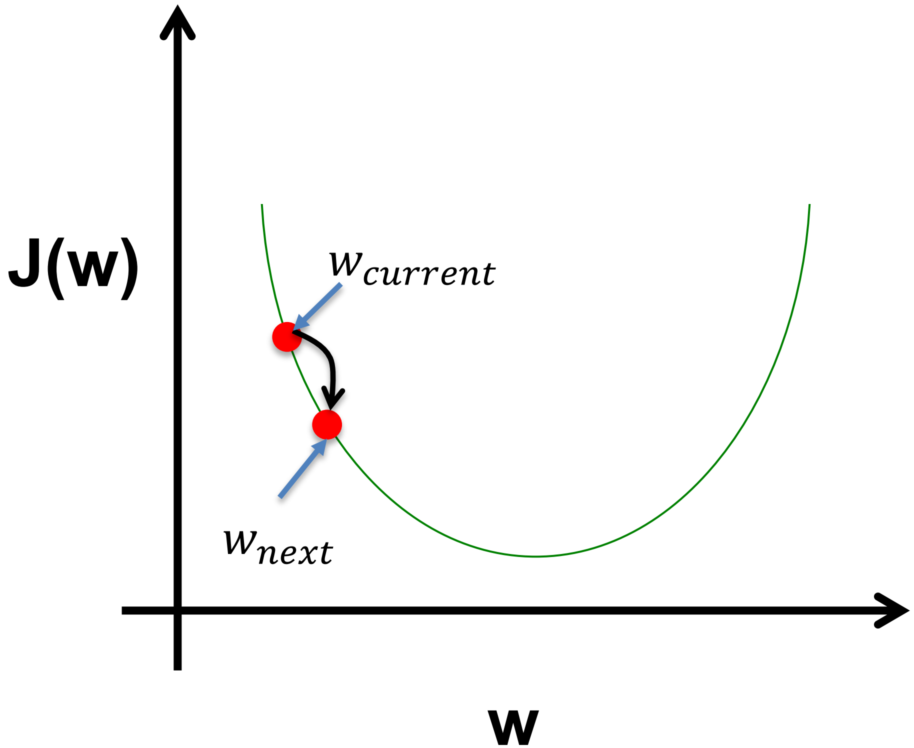


How to Formulate Gradient Descent?

For simplicity we start from 1-D *w*, then will extend the concepts to 2-D *w*'s.

$$w_{next} = w_{current} - \alpha \frac{\partial J(w)}{\partial w}$$

α is learning rate.

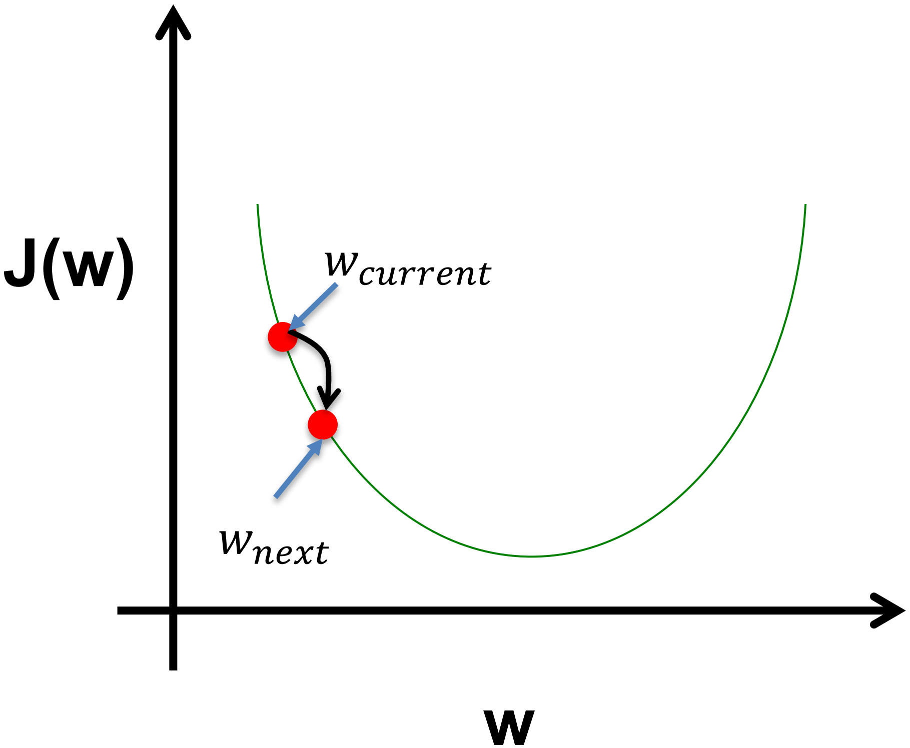


Why this Formula for Gradient Descent works?

For simplicity we start from 1-D *w*, then will extend the concepts to 2-D *w*'s.

$$w_{next} = w_{current} - \alpha \frac{\partial J(w)}{\partial w}$$

α is learning rate.



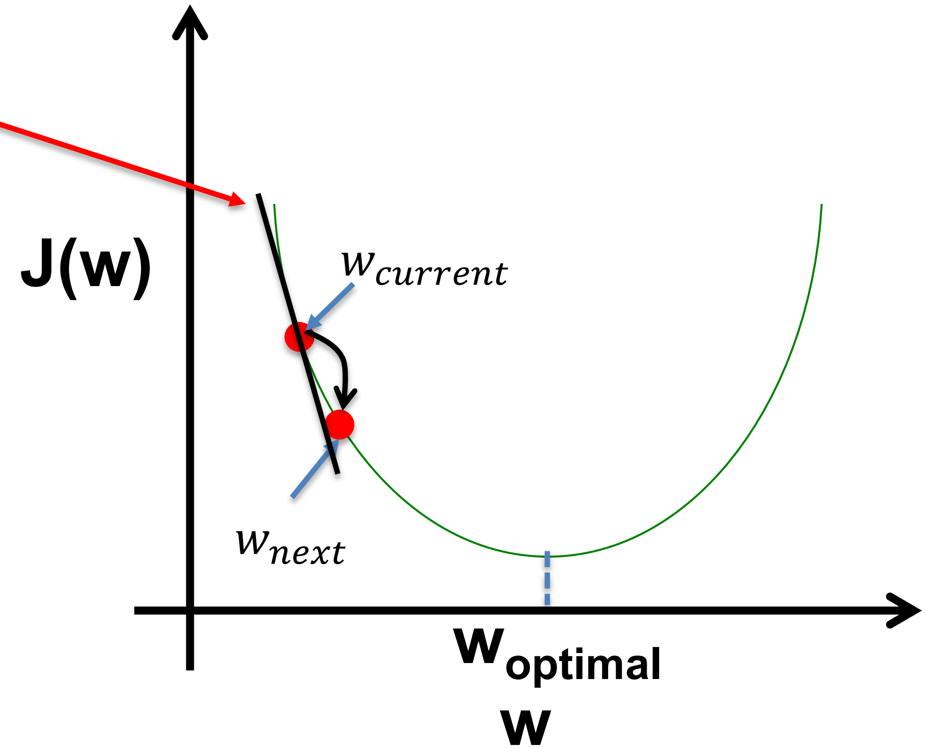
Why this Formula for Gradient Descent works?

For simplicity we start from 1-D *w*, then will extend the concepts to 2-D *w*'s.

$$w_{next} = w_{current} - \alpha \frac{\partial J(w)}{\partial w}$$

α is learning rate.

$0 < \alpha$



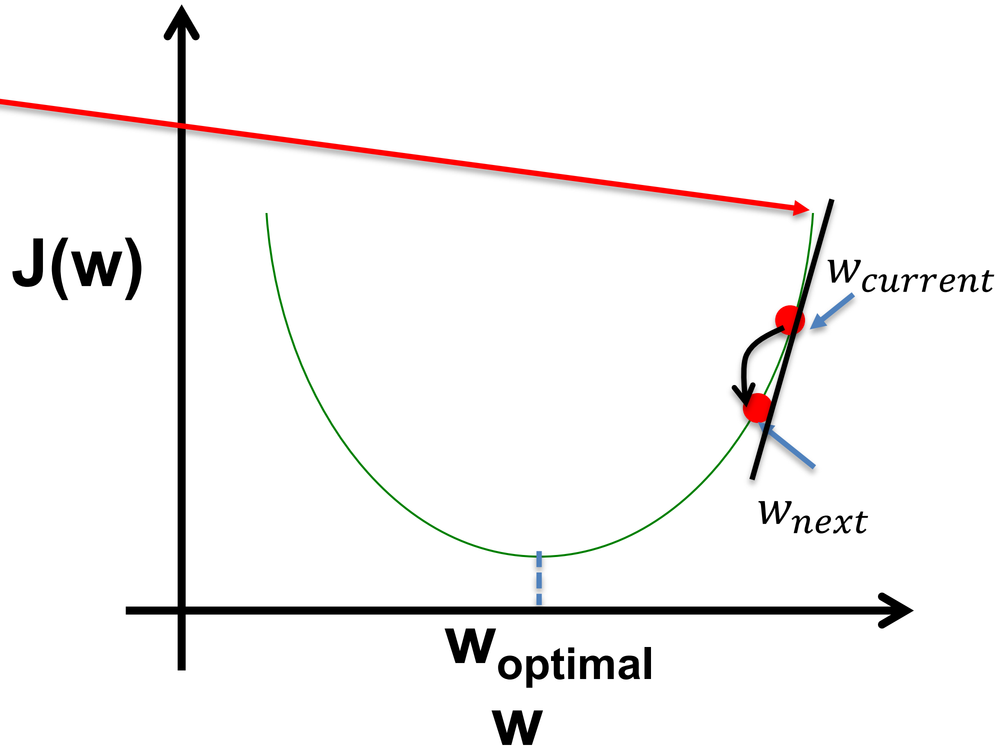
Why this Formula for Gradient Descent works?

For simplicity we start from 1-D *w*, then will extend the concepts to 2-D *w*'s.

$$w_{next} = w_{current} - \alpha \frac{\partial J(w)}{\partial w}$$

α is learning rate.

$0 < \alpha$



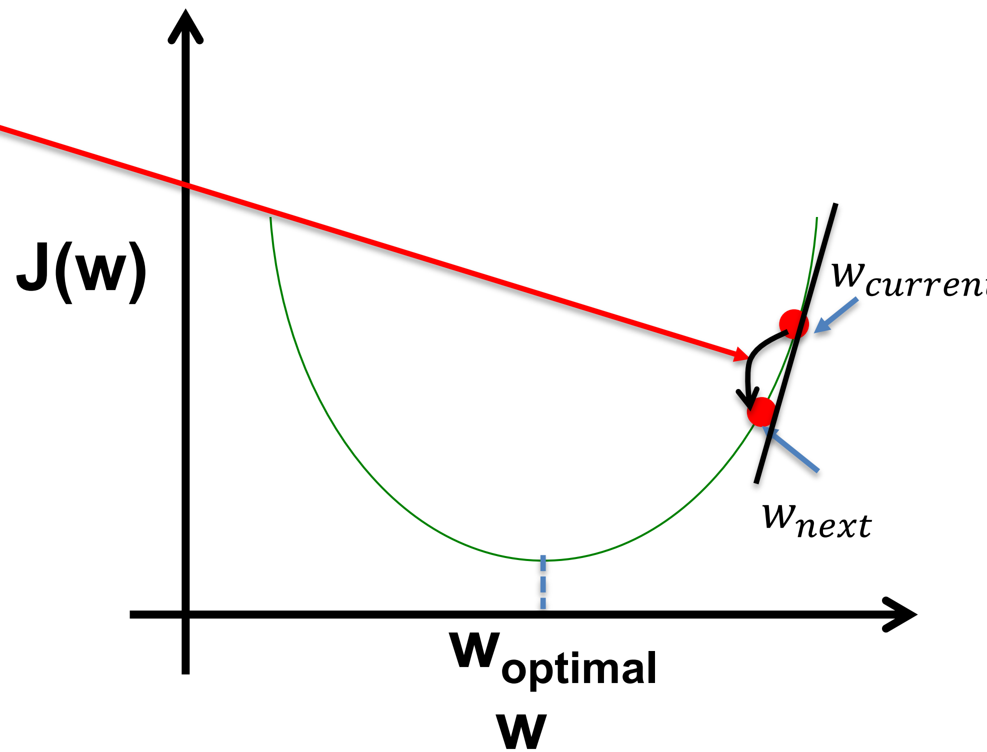
Why this Formula for Gradient Descent works?

For simplicity we start from 1-D w, then will extend the concepts to 2-D w's.

$$w_{next} = w_{current} - \alpha \frac{\partial J(w)}{\partial w}$$

α is learning rate.

$0 < \alpha$



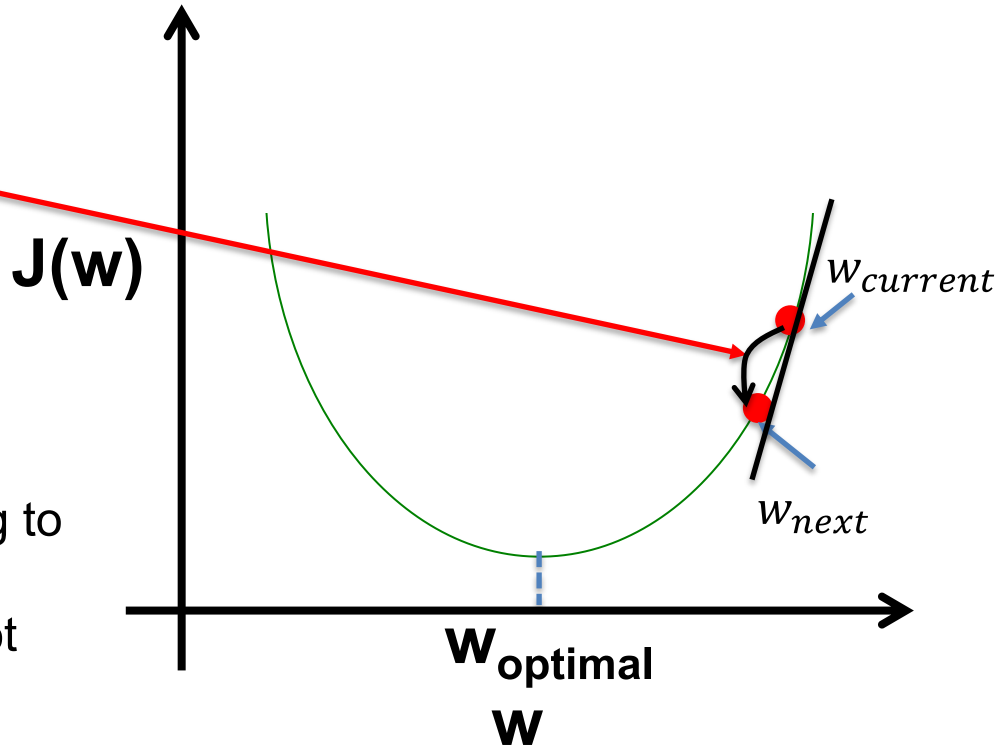
Why this Formula for Gradient Descent works?

For simplicity we start from 1-D *w*, then will extend the concepts to 2-D *w*'s.

$$w_{next} = w_{current} - \alpha \frac{\partial J(w)}{\partial w}$$

α is learning rate.

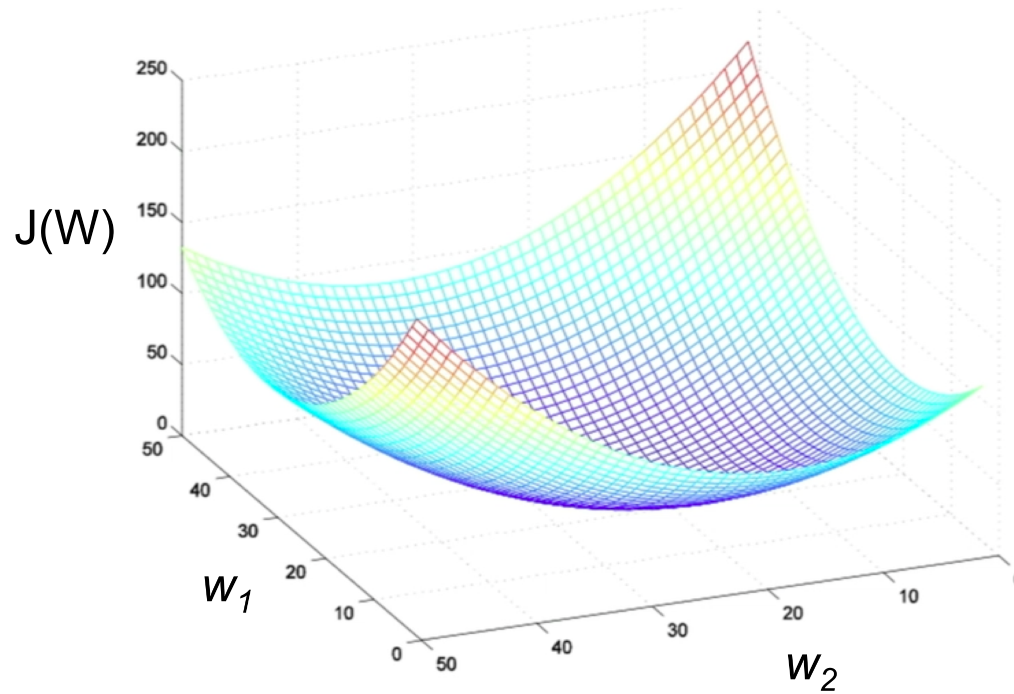
$0 < \alpha$



Set α too small and it will take too long to reach minimum point.
Use a large α and you might overshoot the minimum point.

Gradient Descent in 2D

$$W_{next} = W_{current} - \alpha \nabla_W J(W)$$



Gradient Descent in 2D

$$W_{next} = W_{current} - \alpha \nabla_W J(W)$$

$$\mathrm{J(W)} = \frac{1}{2n} \sum_i (W^T . x_p^i - y^i)^2$$

$$\frac{\partial J(W)}{\partial w_j} = \frac{1}{n} \sum_i (W^T . x_p^i - y^i) X_{pj}^i$$

$$\nabla_W J(W) = \frac{1}{n} X_p^T . (X_p^T . W - y)$$