MIDS W207
Applied Machine Learning

Week 6
Live Session Slides

Gradient Descent

Gradient Descent is an optimization algorithm for finding a local minimum of a differentiable function.

Gradient descent is simply used in machine learning to find the values of a function's parameters (coefficients) that minimize a cost function as far as possible.

It's based on a convex function and tweaks its parameters iteratively to minimize a given function to its local minimum.

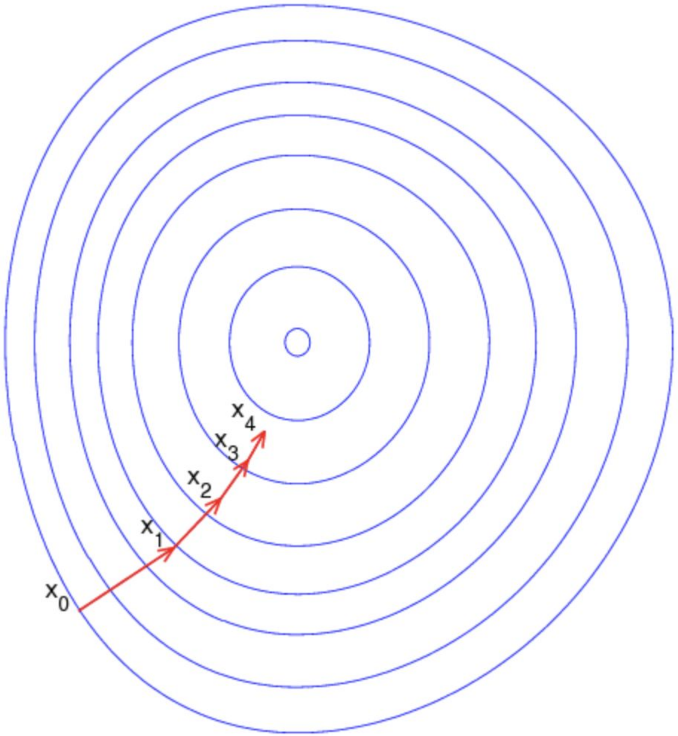
Gradient Descent

> "A gradient measures how much the output of a function changes if you change the inputs a little bit." —Lex Fridman (MIT)

A gradient is a derivative of a function that has more than one input variable.

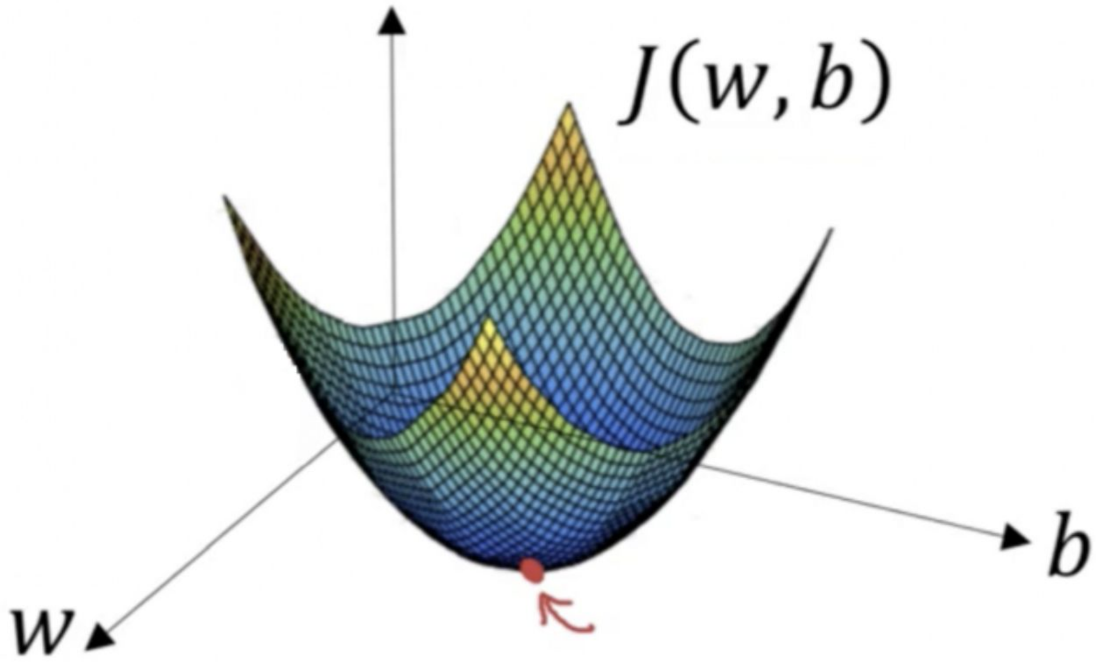
Known as the slope of a function in mathematical terms, the gradient simply measures the change in all weights with regard to the change in error.

Gradient Descent

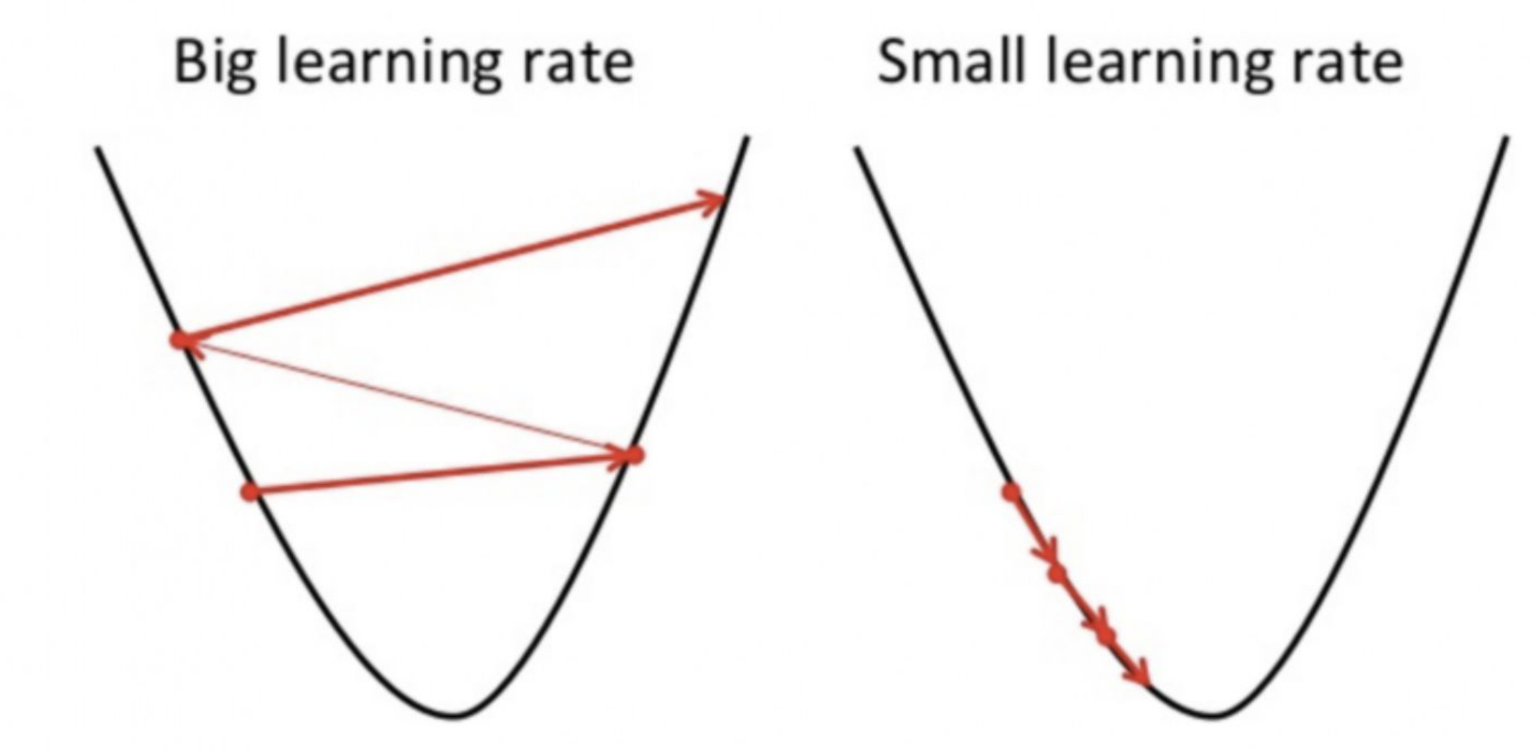


$$\mathbf{b} = \mathbf{a} - \gamma \nabla \mathbf{f}(\mathbf{a})$$

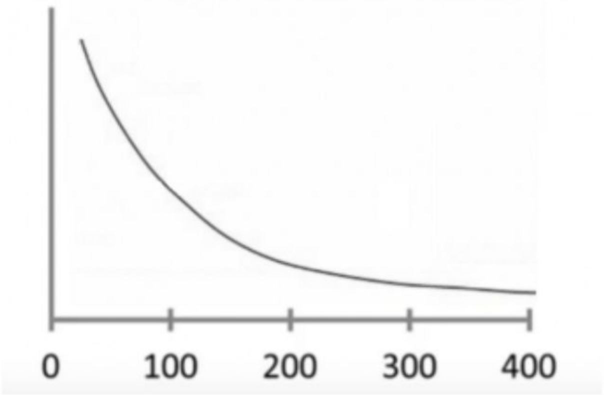
Gradient Descent: Analysis

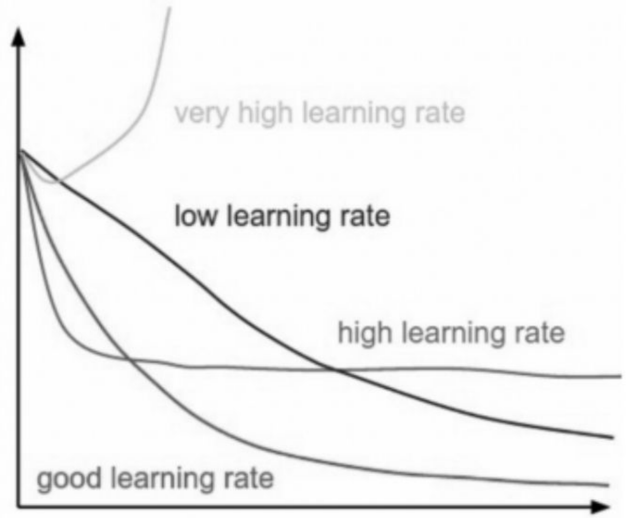


Gradient Descent: Learning Rate



Gradient Descent: Learning Rate

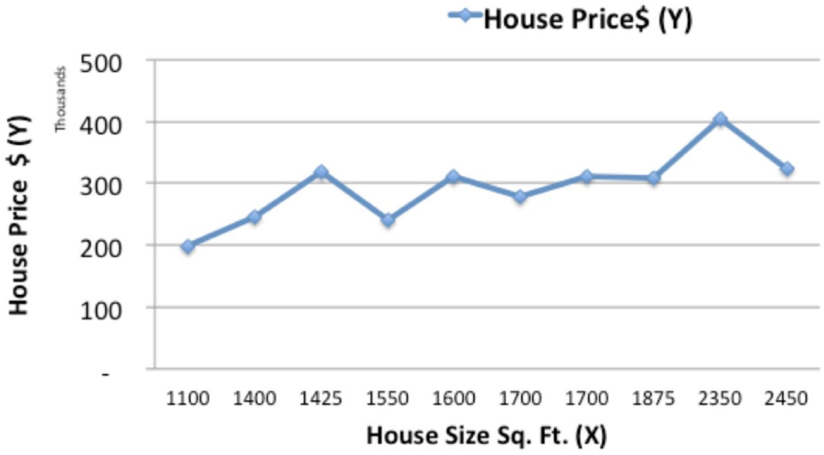




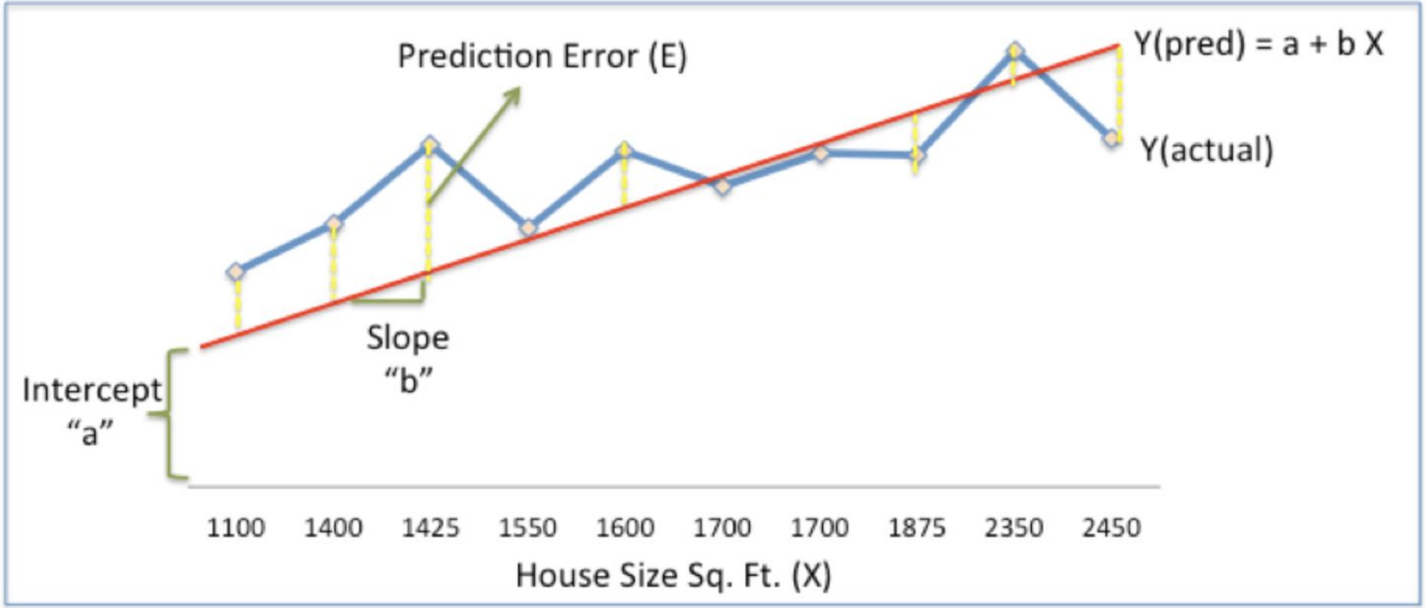
Gradient Descent: Example

House Size sq.ft (X)	1400	1600	1700	1875	1100	1550	2350	2450	1425	1700
House Price$ (Y)	245,000	312,000	279,000	308,000	199,000	219,000	405,000	324,000	319,000	255,000

Given its size (X), what will its price (Y) be?



Gradient Descent: Example



Sum of Squared Errors (SSE) = ½ Sum (Actual House Price – Predicted House Price)^2

= **½ Sum(Y – Ypred)^2**

Gradient Descent: Example

Step 1: Initialize the weights(a & b) with random values and calculate Error (SSE)

Step 2: Calculate the gradient i.e. change in SSE when the weights (a & b) are changed by a very small value from their original randomly initialized value. This helps us move the values of a & b in the direction in which SSE is minimized.

Step 3: Adjust the weights with the gradients to reach the optimal values where SSE is minimized

Step 4: Use the new weights for prediction and to calculate the new SSE

Step 5: Repeat steps 2 and 3 till further adjustments to weights doesn’t significantly reduce the Error

Gradient Descent: Example

HOUSING DATA

House Size (X)	House Price (Y)
1,100	1,99,000
1,400	2,45,000
1,425	3,19,000
1,550	2,40,000
1,600	3,12,000
1,700	2,79,000
1,700	3,10,000
1,875	3,08,000
2,350	4,05,000
2,450	3,24,000

Normalize

Min-Max Standardization

X (X-Min/Max-min)	Y (Y-Min/Max-Min)
0.00	0.00
0.22	0.22
0.24	0.58
0.33	0.20
0.37	0.55
0.44	0.39
0.44	0.54
0.57	0.53
0.93	1.00
1.00	0.61

Gradient Descent: Example

Step 1

a	b	X	Y	YP=a+bX	SSE=1/2(Y-YP)^2
0.45	0.75	0.00	0.00	0.45	0.101
		0.22	0.22	0.62	0.077
		0.24	0.58	0.63	0.001
		0.33	0.20	0.70	0.125
		0.37	0.55	0.73	0.016
		0.44	0.39	0.78	0.078
		0.44	0.54	0.78	0.030
		0.57	0.53	0.88	0.062
		0.93	1.00	1.14	0.010
		1.00	0.61	1.20	0.176
				Total SSE	**0.677**

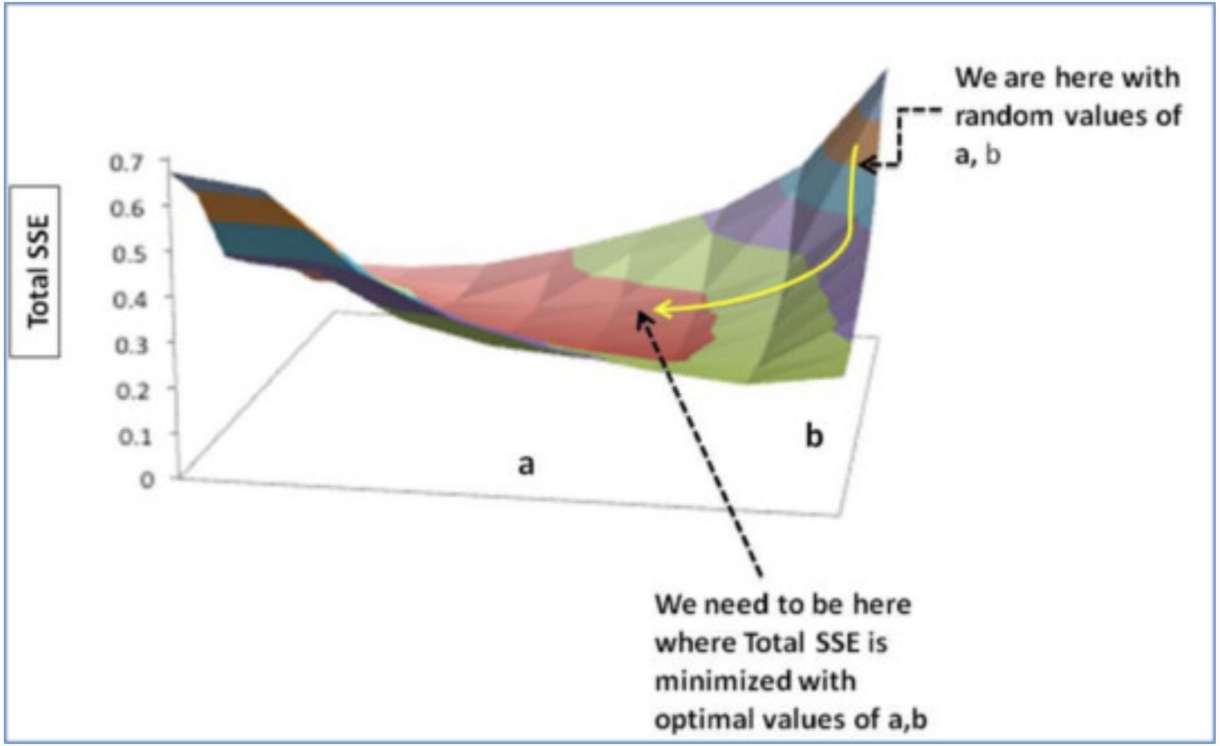
Gradient Descent: Example

Step 2

a	b	X	Y	YP=a+bX		SSE		∂SSE/∂a = -(Y-YP)	∂SSE/∂b = -(Y-YP)X
0.45	0.75	0.00	0.00	0.45		0.101		0.45	0.00
		0.22	0.22	0.62		0.077		0.39	0.09
		0.24	0.58	0.63		0.001		0.05	0.01
		0.33	0.20	0.70		0.125		0.50	0.17
		0.37	0.55	0.73		0.016		0.18	0.07
		0.44	0.39	0.78		0.078		0.39	0.18
		0.44	0.54	0.78		0.030		0.24	0.11
		0.57	0.53	0.88		0.062		0.35	0.20
		0.93	1.00	1.14		0.010		0.14	0.13
		1.00	0.61	1.20		0.176		0.59	0.59
					Total SSE	**0.677**	**Sum**	**3.300**	**1.545**

Gradient Descent: Example

Step 3



Gradient Descent: Example

Step 4

a	b	X	Y	YP=a+bX	SSE		∂SSE/∂a	∂SSE/∂b
0.42	0.73	0.00	0.00	0.42	0.087		0.42	0.00
		0.22	0.22	0.58	0.064		0.36	0.08
		0.24	0.58	0.59	0.000		0.01	0.00
		0.33	0.20	0.66	0.107		0.46	0.15
		0.37	0.55	0.69	0.010		0.14	0.05
		0.44	0.39	0.74	0.063		0.36	0.16
		0.44	0.54	0.74	0.021		0.20	0.09
		0.57	0.53	0.84	0.048		0.31	0.18
		0.93	1.00	1.10	0.005		0.10	0.09
		1.00	0.61	1.15	0.148		0.54	0.54
				Total SSE	0.553	Sum	2.900	1.350

Gradient Descent: In depth Analysis

Formula:

$$X = X - lr * \frac{d}{dX} f(X)$$

$Where,$

$$\begin{aligned} X &= input \\ F(X) &= output\ \ based\ on\ X \\ lr &= learning\ rate \end{aligned}$$

Gradient Descent: Single Variable

<u>Cost Function</u>

$$J(\theta) = \theta^2$$

<u>Goal</u>

$$\min J(\theta)$$

<u>Update Function</u>

$$\theta := \theta - \alpha * \frac{d}{d\theta} J(\theta)$$

<u>Learning Rate</u>

$$Learning\ Rate:$$

$$\alpha = 0.1$$

Gradient Descent: Single Variable

Updating Parameters

$$\theta := \theta - \alpha * \frac{d}{d\theta} J(\theta)$$

$$\theta := \theta - \alpha * 2\theta$$

$$\theta := \theta - 2\alpha\theta$$

$$\theta := 0.8 * \theta$$

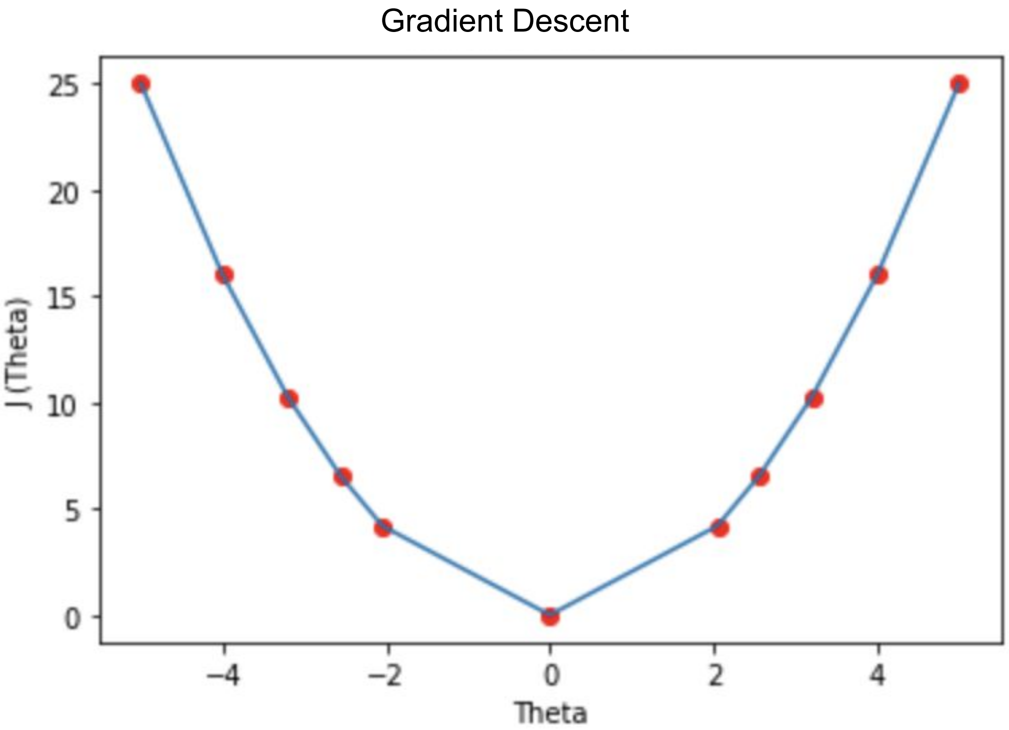
Table Generation

Gradient Descent: Single Variable

θ	J(θ)
5	25
4	16
3.2	10.24
2.56	6.55
2.04	4.19
|	|
|	|
0	0

θ	J(θ)
-5	25
-4	16
-3.2	10.24
-2.56	6.55
-2.04	4.19
|	|
|	|
0	0

Gradient Descent: Single Variable



Gradient Descent: Multiple Variables

Cost Function

$$J(\theta_1, \theta_2) = \theta_1{}^2 + \theta_2{}^2$$

Goal

$$\min \; J(\theta_1, \theta_2)$$

Update Function

$$\theta_1 := \theta_1 - \alpha * \frac{d}{d\theta_1} J(\theta_1, \theta_2)$$

$$\theta_2 := \theta_2 - \alpha * \frac{d}{d\theta_2} J(\theta_1, \theta_2)$$

Gradient Descent: Multiple Variables

Derivatives

$$\frac{d}{d\theta_1}J(\theta_1,\theta_2) = \frac{d}{d\theta_1}({\theta_1}^2 + {\theta_2}^2)$$

$$= \frac{d}{d\theta_1}({\theta_1}^2) + \frac{d}{d\theta_1}({\theta_2}^2)$$

$$= 2\theta_1 + 0$$

$$= 2\theta_1$$

$$\frac{d}{d\theta_2}J(\theta_1,\theta_2) = \frac{d}{d\theta_2}({\theta_1}^2 + {\theta_2}^2)$$

$$= \frac{d}{d\theta_2}({\theta_1}^2) + \frac{d}{d\theta_2}({\theta_2}^2)$$

$$= 0 + 2\theta_2$$

$$= 2\theta_2$$

Gradient Descent: Multiple Variables

Update Values

$$\theta_1 := \theta_1 - \alpha * 2\theta_1$$

$$\theta_1 := \theta_1 - 2\alpha\theta_1$$

$$\theta_2 := \theta_2 - \alpha * 2\theta_2$$

$$\theta_2 := \theta_2 - 2\alpha\theta_2$$

Learning Rate

$$Learning\ Rate:$$

$$\alpha = 0.1$$

Gradient Descent: Multiple Variables

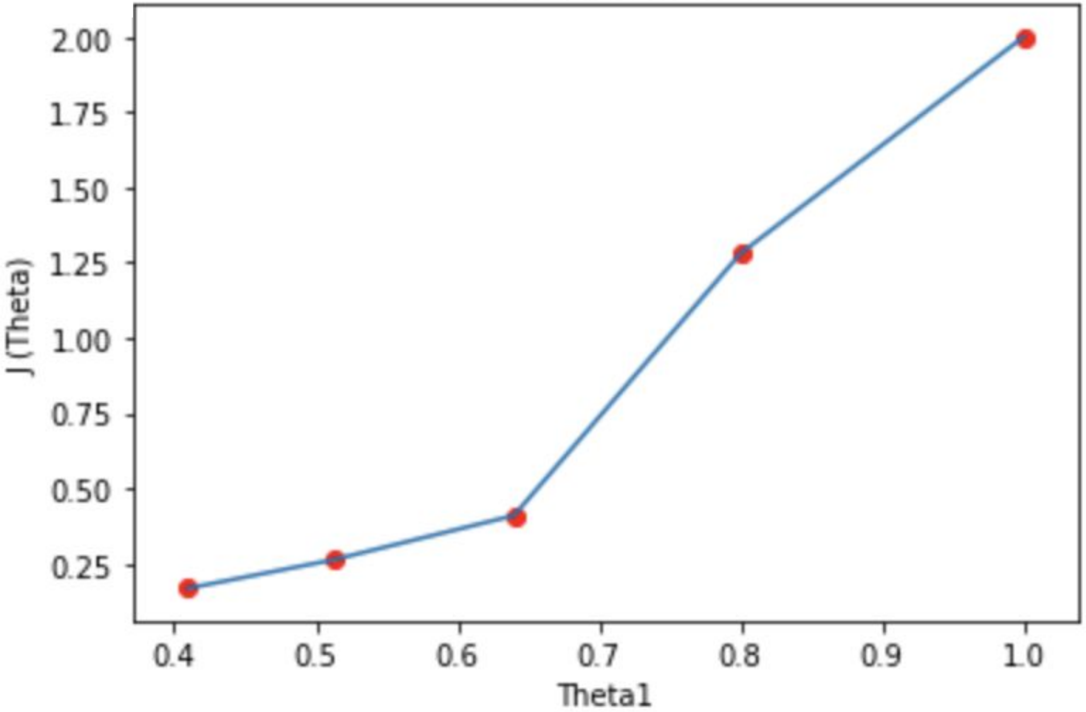
Table

θ1	θ2	J(θ)
1	1	2
0.8	0.8	1.28
0.64	0.64	0.4096
0.512	0.512	0.2621
0.4096	0.4096	0.1677
0	0	0

Gradient Descent: Multiple Variables

Gradient Descent

Graph



Gradient Descent: General Formulation

General Formulation

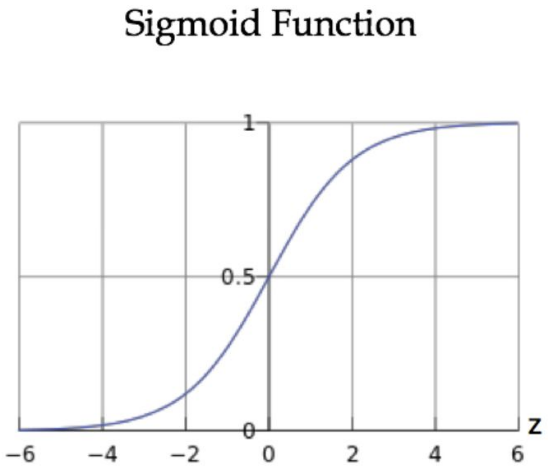
- Model ("hypothesis"): $Y_i = \alpha + \beta X_i$
- Parameters: α, β
- Cost function:

$$J(\alpha, \beta) = \frac{1}{2N} \sum_{i=1}^{N} (Y_i - \alpha - \beta X_i)^2$$

- Objective:

$$\min_{\alpha,\beta} J(\alpha, \beta)$$

Gradient Descent: Logistic Regression



Sigmoid Function

- Logistic (sigmoid) function: $g(z) = \frac{e^z}{e^z+1} = \frac{1}{1+e^{-z}}$
- In logistic regression: $z = \alpha + \beta X + \ldots$
- Transforms: $[-\infty, +\infty] \rightarrow [0, 1]$
- Constrains output of our model between 0 and 1

Gradient Descent: Logistic Regression

Models and Parameters

- Model (hypothesis): $P\left(Y_i = 1 \middle| x : \theta\right) = g\left(z\right) = \frac{1}{1+e^{-z}}$
- Parameters:
  - Above, $\boldsymbol{\theta}$, and in our case, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$

$$\text{If } \theta = (\alpha, \beta), P(Y_i = 1) = \frac{1}{1+e^{-(\alpha+\beta X_i)}}$$

Cost Function

- Cost function, in general: $J\left(\theta\right) = \frac{1}{N} \sum_{i=1}^{N} \text{Cost}\left(\widehat{Y}_i, Y_i\right)$
  - $\hat{\boldsymbol{Y}}$ = predicted value of $\boldsymbol{Y}$
  - $\hat{\boldsymbol{Y}}_{\boldsymbol{i}}$ = predicted value of $\boldsymbol{i}^{\text{th}}$ observation
  - $\boldsymbol{Y}_{\boldsymbol{i}}$ = actual value of $\boldsymbol{i}^{\text{th}}$ observation
  - Cost function, in logistic regression:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} Y_i \cdot \log \hat{Y}_i + (1 - Y_i) \log(1 - \hat{Y}_i)$$

Gradient Descent: Logistic Regression

Objective

- Minimize cost function subject to parameters $\boldsymbol{\theta}$:

$$\min_{\theta} J(\theta)$$

- In our case, minimize cost function subject to parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$.:

$$\min_{\alpha,\beta} J(\alpha, \beta)$$

Gradient Descent: Logistic Regression

Examining the Cost Function

- Logistic regression cost function:

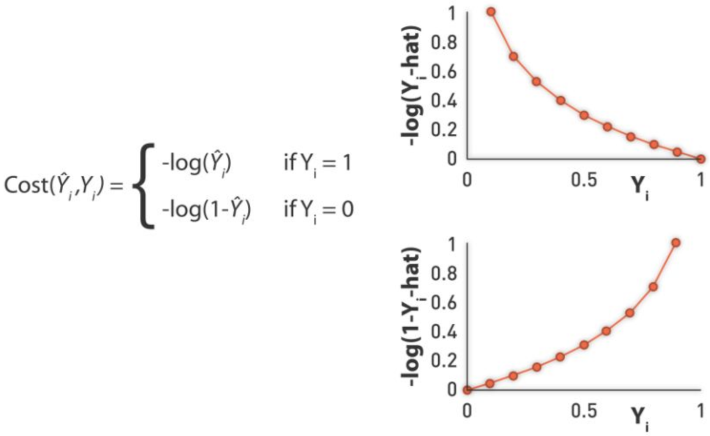
$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} Y_i \cdot \log \hat{Y}_i + (1 - Y_i)\log(1 - \hat{Y}_i)$$

- Can rewrite single part as two different components:

$$\text{Cost}(\hat{Y}_i, Y_i) = \begin{cases} -\log(\hat{Y}_i) & \text{if } Y_i = 1 \\ -\log(1 - \hat{Y}_i) & \text{if } Y_i = 0 \end{cases}$$

- Produces a convex surface

Graphs of Cost Function



Gradient Descent: Logistic Regression

Logistic Regression: Gradient Descent

- Benefit: leads to getting predicted cost values closer to actual values
  - Cost function:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} Y_i \cdot \log \hat{Y}_i + (1 - Y_i)\log(1 - \hat{Y}_i)$$

- Use the update rule:

$$\theta <- \theta - R\frac{\partial}{\partial\theta}J(\theta)$$

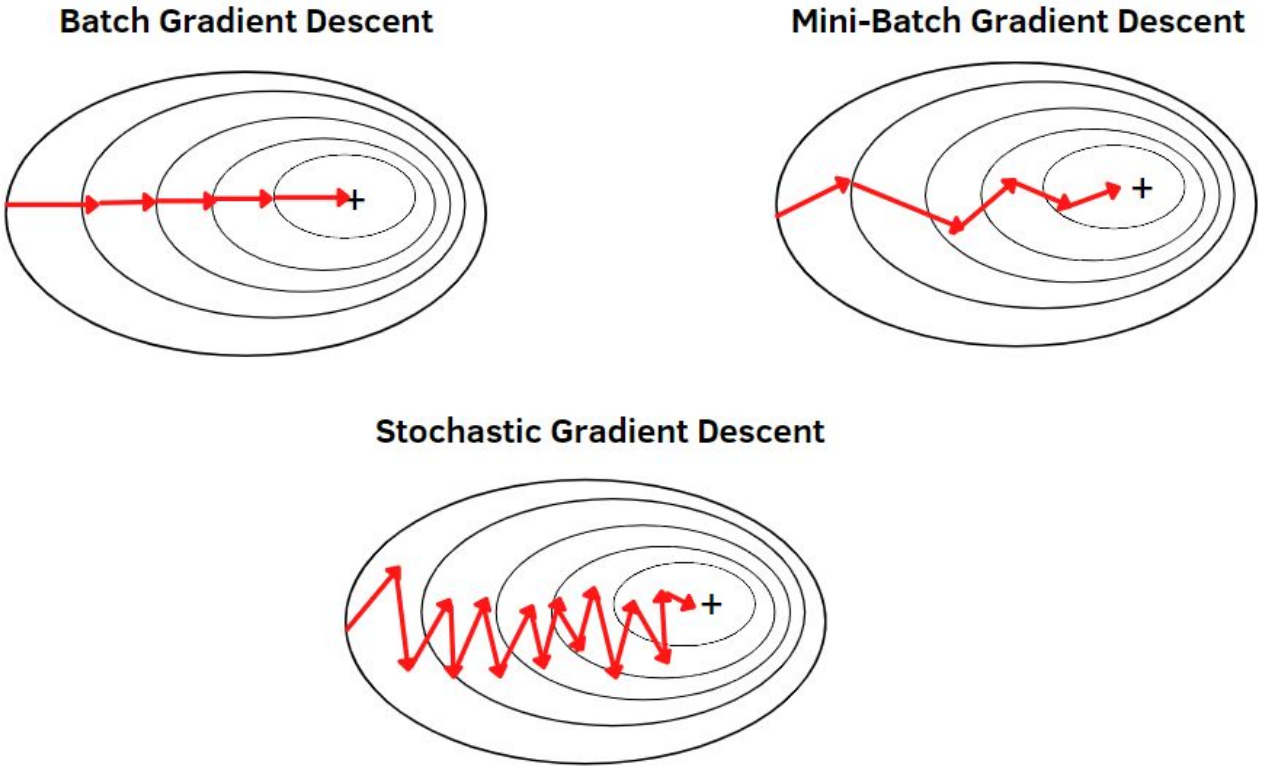
- Benefit: derivative is very simple:

$$\frac{\partial}{\partial\theta}J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)X_i$$

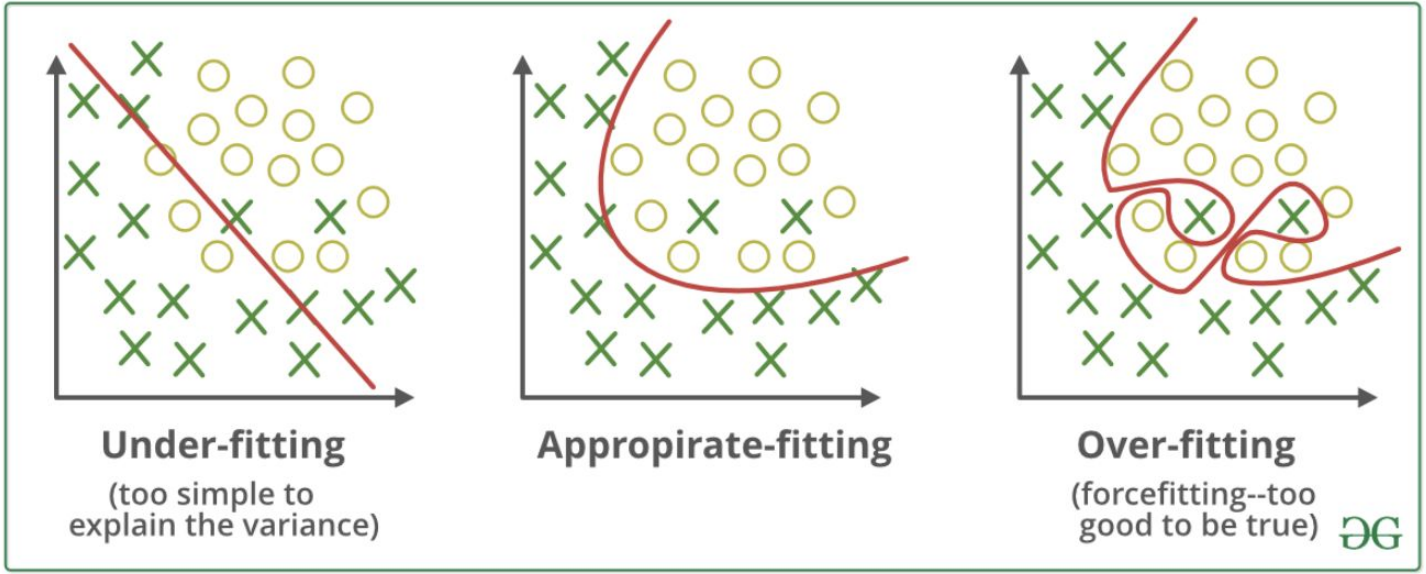
- Gradient descent algorithm:

$$\beta <- \beta - R\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \frac{1}{1+e^{-(\alpha+\beta X_i)}}\right)X_i$$

Gradient Descent: Types



Regularization



Regularization

Changing the Cost Function

- Original cost function:

$$J(\alpha\ ,\beta) = \frac{1}{2N}\sum_{i=1}^{N}(Y_i - \theta_0 + \theta_1 X_i + \ \ldots\ + \theta_k X_i^k)^2$$

- Modified cost function:

$$J(\alpha, \beta) = \frac{1}{2N}\sum_{i=1}^{N}(Y_i - \theta_0 + \theta_1 X_i + \ \ldots\ + \theta_k X_i^k)^2 + \lambda_3\theta_3 + \ \ldots\ + \lambda_k\theta_k$$

  - Add **λ** terms to account for additional, unnecessary terms.
  - Penalized cost function:

$$J(\alpha, \beta) = \frac{1}{2N}\sum_{i=1}^{N}(Y_i - \theta_0 + \theta_1 X_i + \ldots + \theta_k X_i^k)^2 + \overbrace{\boxed{\underbrace{\boxed{\lambda}}_{\text{Regularization parameter}} \sum_{j=1}^{k}\theta_j^2}}^{\text{Penalty}}$$

- Regularized version with new partials:

$$\beta <- \ \beta - R\frac{1}{N}\sum_{i=1}^{N}(Y_i - \alpha - \beta X_i)X_i + \frac{\lambda}{N}\beta$$

Gradient Descent Algorithm

- Pseudocode:
  - Choose an initial vector of parameters α, β.
  - Choose learning rate R.
  - Repeat until an approximate minimum is obtained (randomly shuffle examples in training set).
  - For each example i:

$$\alpha <- \alpha - R \frac{\partial}{\partial \alpha} J(\alpha, \beta)$$

$$\beta <- \beta - R \frac{\partial}{\partial \beta} J(\alpha, \beta)$$

Regression Cost Function

- Regression cost function:

$$J(\alpha, \beta) = \frac{1}{2N} \sum_{i=1}^{N} (Y_i - \alpha - \beta X_i)^2$$

- Repeat until convergence:

$$\alpha <- \alpha - R \frac{\partial}{\partial \alpha} J(\alpha, \beta)$$

$$\beta <- \beta - R \frac{\partial}{\partial \beta} J(\alpha, \beta)$$

- Missing pieces:

$$\frac{\partial}{\partial \alpha} J(\alpha, \beta) = \frac{\partial}{\partial \alpha} \frac{1}{2N} \sum_{i=1}^{N} (Y_i - \alpha - \beta X_i)^2$$

$$\frac{\partial}{\partial \beta} J(\alpha, \beta) = \frac{\partial}{\partial \alpha} \frac{1}{2N} \sum_{i=1}^{N} (Y_i - \alpha - \beta X_i)^2$$

Partial Derivatives:
Cost Function With Respect to α

$$\frac{\partial}{\partial \alpha} J(\alpha, \beta) = \frac{\partial}{\partial \alpha} \frac{1}{2N} \sum_{i=1}^{N} (Y_i - \alpha - \beta X_i)^2$$

$$= \frac{1}{2N} \sum_{i=1}^{N} \frac{\partial}{\partial \alpha} (Y_i - \alpha - \beta X_i)^2$$

$$= \frac{1}{2N} \sum_{i=1}^{N} 2(Y_i - \alpha - \beta X_i) \frac{\partial}{\partial \alpha} (Y_i - \alpha - \beta X_i)$$

$$= -\frac{1}{N} \sum_{i=1}^{N} (Y_i - \alpha - \beta X_i)$$

Partial Derivatives:
Cost Function With Respect to β

$$\frac{\partial}{\partial \beta} J(\alpha, \beta) = \frac{\partial}{\partial \beta} \frac{1}{2N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2$$

$$= \frac{1}{2N} \sum_{i=1}^{N} \frac{\partial}{\partial \beta} (Y_i - \alpha - \beta X_i)^2$$

$$= \frac{1}{2N} \sum_{i=1}^{N} 2(Y_i - \alpha - \beta X_i) \frac{\partial}{\partial \beta} (Y_i - \alpha - \beta X_i)$$

$$= \frac{1}{N} \sum_{i=1}^{N} (Y_i - \alpha - \beta X_i)(-X_i)$$

$$= -\frac{1}{N} \sum_{i=1}^{N} (Y_i - \hat{Y}_i) X_i$$