Linear Models

Characteristics of Linear Models

Linear Models are parametric i.e. they have a fixed form with a small number of numeric parameters that are to be learnt from data.

Linear Models are stable. It means that small variations in the training data have only limited impact on the learned model.

Linear Models are less likely to overfit the training data because they have relatively few parameters.

Linear models have high bias and low variance. (Underfitting)

Linear models are preferable when there is limited training data and overfitting is to be avoided.

Linear Regression

Linear regression is a method for finding the straight line or hyperplane that best fits a set of points.

It a very simple supervised learning approach.

It is used to predict a quantitative response like price, temperature etc.

It is a widely used statistical learning method.

When there is only feature it is called ***Univariate Linear Regression***

Eg: Predicting prices of house based on the area of the house

and if there are multiple features, it is called ***Multiple Linear Regression***.

Eg: Predicting prices of house based on the area of the house, number of floors, number of rooms etc.

Advertising Data (Multivariate Regression)

	TV	radio	newspaper	sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
6	8.7	48.9	75	7.2
7	57.5	32.8	23.5	11.8
8	120.2	19.6	11.6	13.2
9	8.6	2.1	1	4.8
10	199.8	2.6	21.2	10.6

Simple Linear Regression

Simple Linear Regression

It is a simple approach for predicting a quantitative response Y on the basis of a single predictor variable X.

It assumes that there is approximately a linear relationship between X and Y.

Mathematically, it can be written as:

$$Y \approx \beta_0 + \beta_1 X$$

β_0 and β_1 are two unknown constants that represent the intercept and slope terms in the linear model.

Together, β_0 and β_1 are known as model coefficients or parameters (to be learnt).

This is also called as *regressing Y on X*.

Simple Linear Regression

For this example, it can be written as

$$\text{sales} \approx \beta_0 + \beta_1 * \text{TV}$$

Once the values of β_0 and β_1 have been estimated using the training data, we can predict future sales on the basis of expenditure done on TV advertising.

Estimating the Coefficients

Let us say, we have a dataset as

$$(x_1,y_1), (x_2,y_2),......(x_n,y_n)$$

Now, we want to find an intercept β_0 and slope β_1 such that the resulting straight line is as close as possible to the n data points.

Most common approach involves minimising the *least squares* criterion.

Ways to Estimate the Coefficients

- Ordinary Least Square Method
- Gradient Descent Method
- Normal Equation Method

Ordinary Least Square Method

Least Squares Method

Let $\hat{y}_i = \beta_0 + \beta_1 * x_i$ be the prediction for Y based on the *i*th value of X.

Then, $e_i = y_i - \hat{y}_i$ represents the *i*th residual - this is the difference between the *i*th observed response value and the *i*th response value that is predicted by the linear model.

Residual sum of squares is defined as:

$$RSS = e_1^2 + e_2^2 + \ldots + e_n^2$$

Understanding Residual Sum of Squares

X	Y	Y_predicted_1	Y_predicted_2	(Y-Y1)^2	(Y-Y2)^2	
95	85	86	88	1	9	
85	95	88	81	49	196	
80	70	72	66	4	16	
70	65	64	72	1	49	
60	70	69	64	1	36	
				56	306	SUM

RSS of Model 1 is lower than that of Model 2. So Model 1 is preferable over Model 2.

But the question is, how to get to (find out) Model 1 (or the best model)?

Derivation

$$SSE\left(\widehat{\beta}_0, \widehat{\beta}_1\right) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$= \sum_{i=1}^{n} \left(y_i - \left(\widehat{\beta}_0 + \widehat{\beta}_1 x_i\right)\right)^2$$

$$= \sum_{i=1}^{n} \left(y_i^2 + \left(\widehat{\beta}_0 + \widehat{\beta}_1 x_i\right)^2 - 2y_i\left(\widehat{\beta}_0 + \widehat{\beta}_1 x_i\right)\right)$$

$$= \sum_{i=1}^{n} \left(y_i^2 + \widehat{\beta}_0^2 + \widehat{\beta}_1^2 x_i^2 + 2\widehat{\beta}_0 \widehat{\beta}_1 x_i - 2y_i \widehat{\beta}_0 - 2y_i \widehat{\beta}_1 x_i\right)$$

taking partial derivative w.r.t $\widehat{\beta}_0$

$$= \sum_{i=1}^{n} \left(0 + 2\widehat{\beta}_0 + 0 + 2\widehat{\beta}_1 x_i - 2y_i - 0\right)$$

expanding summation

$$= 2n\widehat{\beta}_0 + 2n\widehat{\beta}_1\overline{x} - 2n\overline{y}$$

$$\frac{\delta}{\delta\widehat{\beta}_0} SSE\left(\widehat{\beta}_0, \widehat{\beta}_1\right) = 0$$

$$-2n\overline{y} + 2n\widehat{\beta}_1\overline{x} + 2n\widehat{\beta}_0 = 0$$

$$\boxed{\widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1\overline{x}}$$

Now solving for β_1

$$\text{differentiating w.r.t. } \widehat{\beta}_1$$

$$SSE(\beta_0, \beta_1) = \sum_{i=1}^{n} \left[y_i^2 + \beta_0^2 + \widehat{\beta}_1^2 x_i^2 + 2\widehat{\beta}_0 \widehat{\beta}_1 x_i - 2y_i \widehat{\beta}_0 - 2y_i \widehat{\beta}_1 x_i \right]$$

$$= \sum_{i=1}^{n} \left[0 + 0 + 2\widehat{\beta}_1 x_i^2 + 2\widehat{\beta}_0 x_i - 0 - 2y_i x_i \right]$$

$$= -\sum x_i y_i + \widehat{\beta}_1 \sum x_i^2 + \widehat{\beta}_0 \sum x_i$$

$$\text{substituting } \widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1 \overline{x}$$

$$= -\sum x_i y_i + \widehat{\beta}_1 \sum x_i^2 + \bar{y} \sum x_i - \widehat{\beta}_1 \bar{x} \sum x_i = 0$$

$$\bar{y} \sum x_i - \sum x_i y_i = \widehat{\beta}_1 \left(\bar{x} \sum x_i - \sum x_i^2 \right)$$

$$\widehat{\beta}_1 = \frac{\bar{y} \sum x_i - \sum x_i y_i}{\bar{x} \sum x_i - \sum x_i^2}$$

$$\widehat{\beta}_1 = \frac{n\bar{x}\bar{y} - \sum x_i y_i}{n x_i^2 - \sum x_i^2}$$

$$\boxed{\widehat{\beta}_1 = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sum x_i^2 - n x_i^2}}$$

Alternative Formulas

With a bit of algebra, we can write the numberator as

$$\sum_{i=1}^{n} x_i y_i - n\overline{y}\,\overline{x} = \sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y}) =: S_{XY}$$

and the denominator as

$$\sum_{i=1}^{n} x_i^2 - n\overline{x}^2 = \sum_{i=1}^{n} (x_i - \overline{x})^2 =: S_{XX}.$$

Thus, we can write $\hat{\beta}_1$ as

$$\hat{\beta}_1 = \frac{S_{XY}}{S_{XX}}.$$

The least squares estimates of β_0 and β_1 are:

$$
\begin{aligned}
\hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2} \\
\hat{\beta}_0 &= \bar{Y} - \hat{\beta}_1 \bar{X}
\end{aligned}
$$

Sample Question

3.

Given below is the data of five students who took a proficiency test as well as language course.

S. No.	Marks in proficiency test	Marks in language course
1	95	85
2	85	95
3	80	70
4	70	65
5	60	70

- Use the least square approximation to estimate the linear equation that best predicts language course performance, based on proficiency test scores?
- Compute the sum of squared error (SSE) using the estimated model.
- If a student scored 80 on the proficiency test, what marks would we expect her to obtain in the language course?

Working

β_1 = ((30500) - (5*78*77)) / (31150 - 5*6084)

= (30500 - 30030) / (31150 - 30420)

= (470)/ (730) = **0.6438**

β_0 = 77 - (-0.6438)(78)

= 77 - 50.2191

= **26.7808**

Column1	X	Y	xiyi	xi^2
	95	85	8075	9025
	85	95	8075	7225
	80	70	5600	6400
	70	65	4550	4900
	60	70	4200	3600
SUM			30500	31150
	n=	5		
	mean of x	78		6084
	mean of y	77		

Marks_in_lang_course = 26.7808 + (0.6438)(marks_in_prof_course)

Computing the sum of squared error (SSE)

Column1	X	Y	xiyi	xi^2	Predicted y	residual=observed - predicted	squared error
	95	85	8075	9025	87.9418	-2.9418	8.65418724
	85	95	8075	7225	81.5038	13.4962	182.1474144
	80	70	5600	6400	78.2848	-8.2848	68.63791104
	70	65	4550	4900	71.8468	-6.8468	46.87867024
	60	70	4200	3600	65.4088	4.5912	21.07911744
SUM			30500	31150			327.3973004
	n=	5					
	mean of x	78		6084			
	mean of y	77					
	beta_0	26.7808					
	beta_1	0.6438					

Making Prediction

If a student scored 80 on the proficiency test, what marks would we expect her to obtain in the language course?

Set $x=80$, $\beta_1 = 0.6438$ and $\beta_0 = 26.7808$

Predicted marks = $26.7808 + (0.6438)(80)$

$= 26.7808 + 51.504$

$= 78.2845$

Points to Ponder

The intercept β_0 is such that the regression line goes through $(\overline{x}, \overline{y})$

Sum of residuals of the least squares solution is zero.

$$\sum_{i=1}^{n}\left(y_i - \left(\widehat{\beta}_0 + \widehat{\beta}_1 x_i\right)\right) = n\left(\overline{y} - \widehat{\beta}_0 - \widehat{\beta}_1 \overline{x}\right) = 0$$

This property also makes linear regression susceptible to ***outliers***.

Outliers are the points that too far away from the regression line (or most of the data points), often because of measurement errors.

Multivariate Linear Regression

Given marks in english, marks in mathematics then predict the GATE score

Y (gate_score) [RESPONSE VARIABLE]

x_1 (eng_score) and x_2 (math_score) [INPUT VARIABLES]

$Y = beta_0 + beta_1\ x_1 + beta_2\ x_2$

Normal Equation Method

Matrix Notation

In order to deal with an arbitrary number of features it will be useful to employ matrix notation.

Univariate linear regression can be written as:

$$\begin{pmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_m \end{pmatrix} = \begin{pmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{pmatrix} \widehat{\beta}_0 + \begin{pmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_m \end{pmatrix} \widehat{\beta}_1 + \begin{pmatrix} \varepsilon_1 \\ \cdot \\ \cdot \\ \cdot \\ \varepsilon_m \end{pmatrix} \Rightarrow \begin{pmatrix} y_1 \\ \cdot \\ \cdot \\ \cdot \\ y_m \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 1 & x_m \end{pmatrix} \begin{pmatrix} \widehat{\beta}_0 \\ \widehat{\beta}_1 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \cdot \\ \cdot \\ \cdot \\ \varepsilon_m \end{pmatrix}$$

$$\boldsymbol{Y} = \boldsymbol{X}\widehat{\boldsymbol{\beta}} + \boldsymbol{\varepsilon}$$

Matrix Notation

For m examples with n features, this can be written more generally as:

$$\begin{pmatrix} y_1 \\ . \\ . \\ . \\ y_m \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{1n} \\ . & . & . \\ . & . & . \\ . & . & . \\ 1 & x_{m1} & x_{mn} \end{pmatrix} \begin{pmatrix} \widehat{\beta}_0 \\ \widehat{\beta}_1 \\ . \\ . \\ . \\ \widehat{\beta}_n \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ . \\ . \\ . \\ \varepsilon_m \end{pmatrix}$$

$$\boldsymbol{y} = \boldsymbol{X}\widehat{\beta} + \varepsilon$$

Here, X is a *mxn* matrix whose first column is all 1s and the remaining columns are the columns of X and β hat has the intercept β_0 hat as its first entry and the regression coefficients as the remaining *n* entries.

Normal Equation

The β *hat* vector can be computed using normal equation as given below:

$$\widehat{\beta} = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T y$$

Characteristics of Normal Equation Method

Normal Equation method is used:

- If n (number of features) is small.
- If m (number of training examples) is small i.e. around 20,000.

One step method for calculating regression coefficients.

Computation increases significantly when number of features increase as the size of matrix increases and matrix multiplication is a computationally intensive operation.

Practice Problem

Find the least square regression line for the given dataset using the normal equation method. Show computation at each step. [2022]

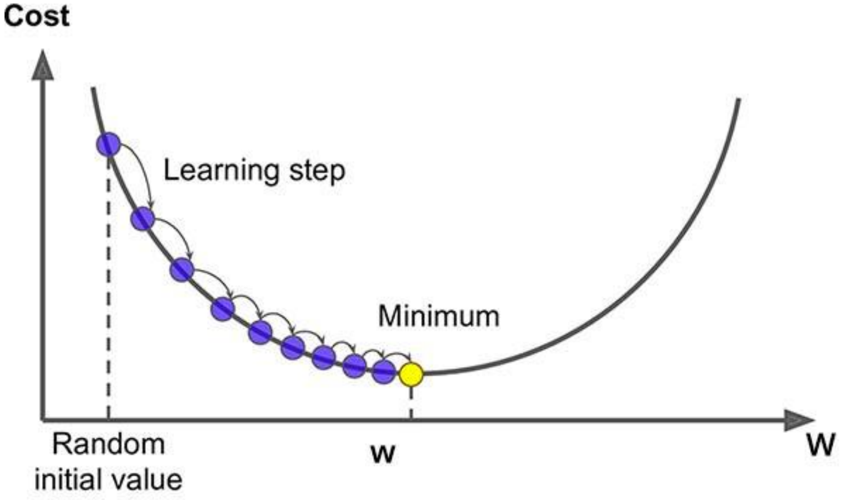
x1	x2	y
1	9	14
2	1	7
3	2	12
4	3	16
5	4	20

Gradient Descent

Gradient Descent

Gradient Descent is a an optimization algorithm that can be used to find the global or local minima of a differentiable function.

It is an iterative algorithm.



Notations

$$x_j^{(i)} = \text{value of feature } j \text{ in the } i^{th} \text{ training example}$$

$$x^{(i)} = \text{the column vector of all the feature inputs of the } i^{th} \text{ training example}$$

$$m = \text{the number of training examples}$$

$$n = \left|x^{(i)}\right|; \text{(the number of features)}$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots + \theta_n x_n$$

Matrix Notation

$$h_\theta(x) = \begin{bmatrix} \theta_0 & \theta_1 & \ldots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x$$

We will set $x_0^{(i)} = 1$, for all values of i.

$$X = \begin{bmatrix} x_0^{(1)} & x_1^{(1)} \\ x_0^{(2)} & x_1^{(2)} \\ x_0^{(3)} & x_1^{(3)} \end{bmatrix}, \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

Loss/Cost Function

A loss/cost function is a function that signifies how much our predicted values is deviated from the actual values of the dependent variable.

For the parameter vector θ (of type $\mathbb{R}^{n+1}$ or in $\mathbb{R}^{(n+1)\times 1}$, the cost function is:

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

Understanding Cost Function

Training Data

x_0	x_1	y
1	1	1
1	2	2
1	3	3

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

Assuming θ_0 = 0, find out J(θ_1)

a) $\theta_1 = 1$

$$J(1) = \frac{1}{2m}\left[0^2 + 0^2 + 0^2\right] = 0$$

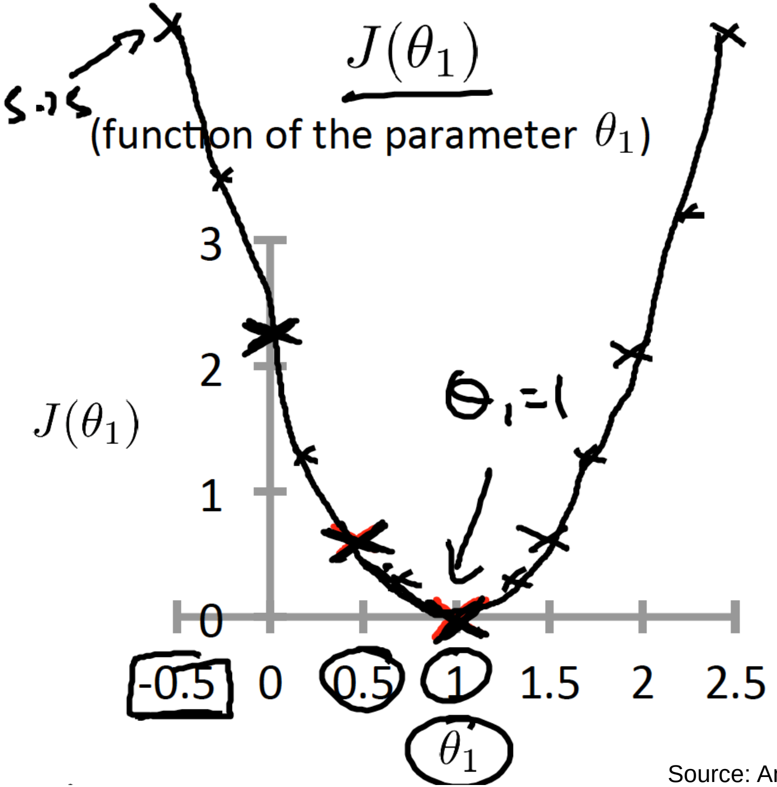
a) $\theta_1 = 0.5$

$$J(0.5) = \frac{1}{2m}\left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right] = \frac{3.5}{6} \approx 0.58$$

a) $\theta_1 = 0$

$$J(0) = \frac{1}{2m}\left[(0-1)^2 + (0-2)^2 + (0-3)^2\right] = \frac{14}{6} \approx 2.3$$

Understanding Cost Function

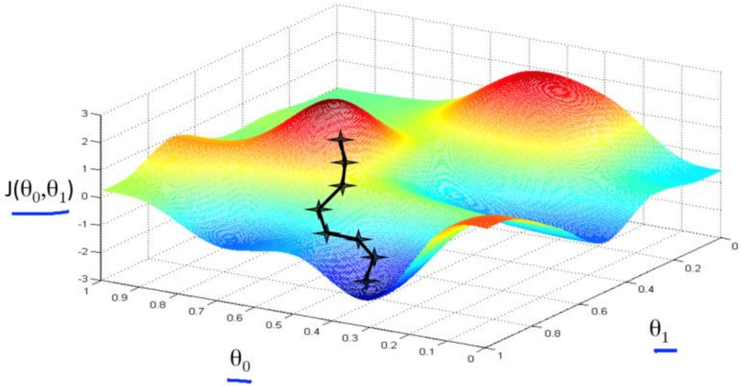


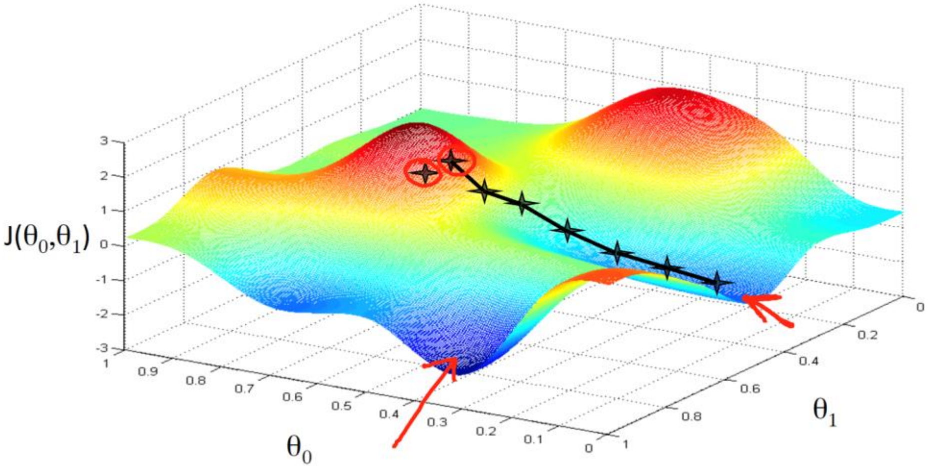
Source: Andrew Ng, Machine Learning Course, Coursera

Steps Involved in Linear Regression with Gradient Descent Implementation

1. Initialize the weight and bias (i.e. regression coefficients) randomly or with 0(both will work).
2. Make predictions with this initial weight and bias.
3. Compare these predicted values with the actual values and define the loss function using both these predicted and actual values.
4. With the help of differentiation, calculate how loss function changes with respect to weight and bias term.
5. Update the weight and bias term so as to minimize the loss function.

To update $\boldsymbol{\theta}$s, we need to calculate the gradients for each $\boldsymbol{\theta}_i$





Source: Andrew Ng, Machine Learning Course, Coursera

Differentiating J(θ_0,θ_1) w.r.t to θ_0 and θ_1

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

$$\frac{\partial}{\partial\theta_0} \frac{1}{2m} \sum_{i=1}^{m} ((\theta_0 + \theta_1 x_1) - y)^2$$

$$\frac{\partial}{\partial\theta_0} \frac{1}{2m} \sum_{i=1}^{m} \left((\theta_0 + \theta_1 x_1)^2 + y^2 - 2(\theta_0 + \theta_1 x)y\right)$$

$$\frac{\partial}{\partial\theta_0} \frac{1}{2m} \sum_{i=1}^{m} (\theta_0^2 + \theta_1^2 x_1^2 + 2\theta_0\theta_1 x + y^2 - 2(\theta_0 + \theta_1 x)y)$$

Differentiating J($\boldsymbol{\theta}_0$,$\boldsymbol{\theta}_1$) w.r.t to $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$

$$\frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(2\theta_0 + 0 + 2\theta_1 x + 0 - 2y)$$

Cancel out 2 from numerator and denominator

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

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

Now differentiating w.r.t. θ_1

$$\frac{\partial}{\partial\theta_1} J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(0 + 2\theta_1 x_1^2 + 2\theta_0 x_1 - 2x_1 y\right)$$

Cancelling out 2 from numerator and denominator and taking x_1 common

$$\frac{\partial}{\partial\theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (\theta_1 x_1 + \theta_0 - y) x_1$$

$$\frac{\partial}{\partial\theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x) - y) x_1$$

Calculating Gradients

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

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x) - y) x_1$$

$$\vdots$$

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

Updating weights

$$\theta_0 = \theta_0 - \alpha \frac{\partial J}{\partial \theta_0}$$

$$\theta_1 = \theta_1 - \alpha \frac{\partial J}{\partial \theta_1}$$

$$\vdots$$

$$\theta_n = \theta_n - \alpha \frac{\partial J}{\partial \theta_n}$$

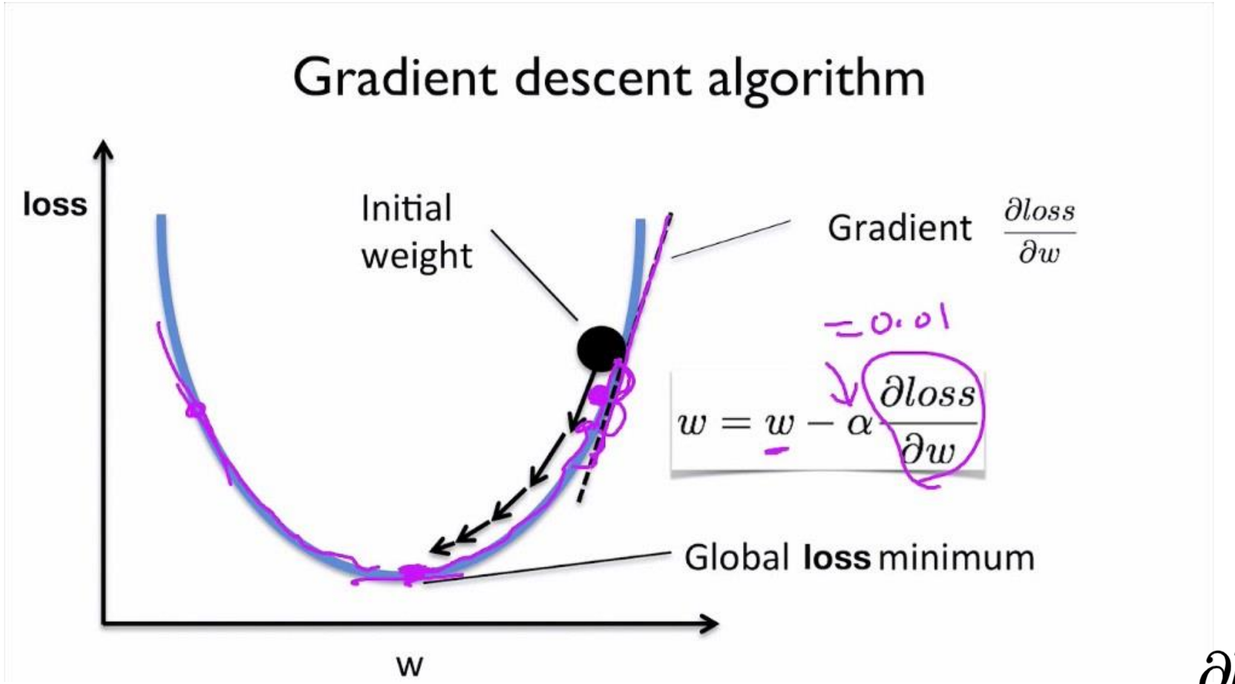
More generally, it can be written as:

repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \qquad \text{for j := 0..n}$$

}

Visualizing Gradient Descent Algorithm



The gradient basically represents the slope of the line.
When y increases with x, the line has a +ve slope, thus w is decreased.

When y decreases with x, the line has a -ve slope thus w is increased.
In both scenarios, the w moves towards the minimum.

$$\frac{\partial loss}{\partial w} \text{ is same as } \frac{\partial J}{\partial \theta}$$

What is alpha?

Alpha represents the learning rate i.e. the speed at which the algorithm moves towards the minimum point.

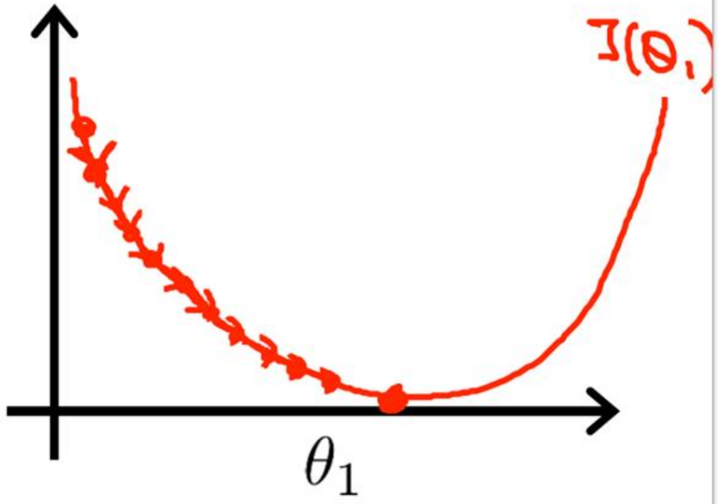
Learning rate alpha is something that we have to manually choose and it is something which we don’t know beforehand. Mostly value of 0.01 is chosen.

A too low value of alpha can make the algorithm move very very slow. The algorithm is said to converge too slowly.

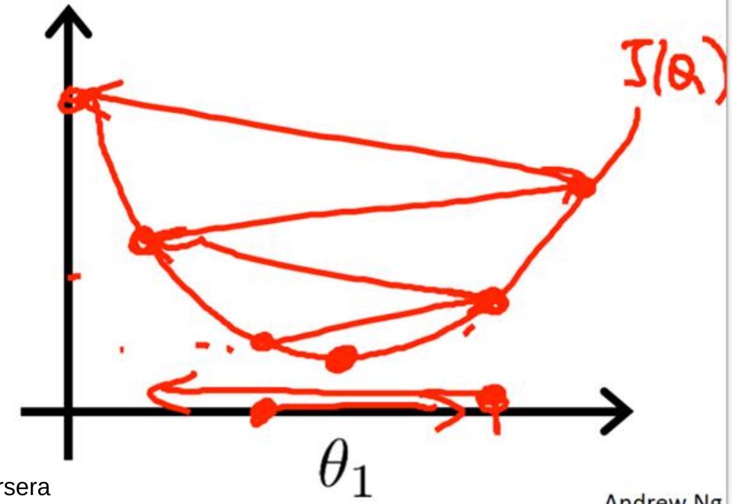
A too high value of alpha can make the algorithm overshoot the minimum point and thus never reach the minimum point.

$$\boxed{\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)}$$

If α is too small, gradient descent can be slow.

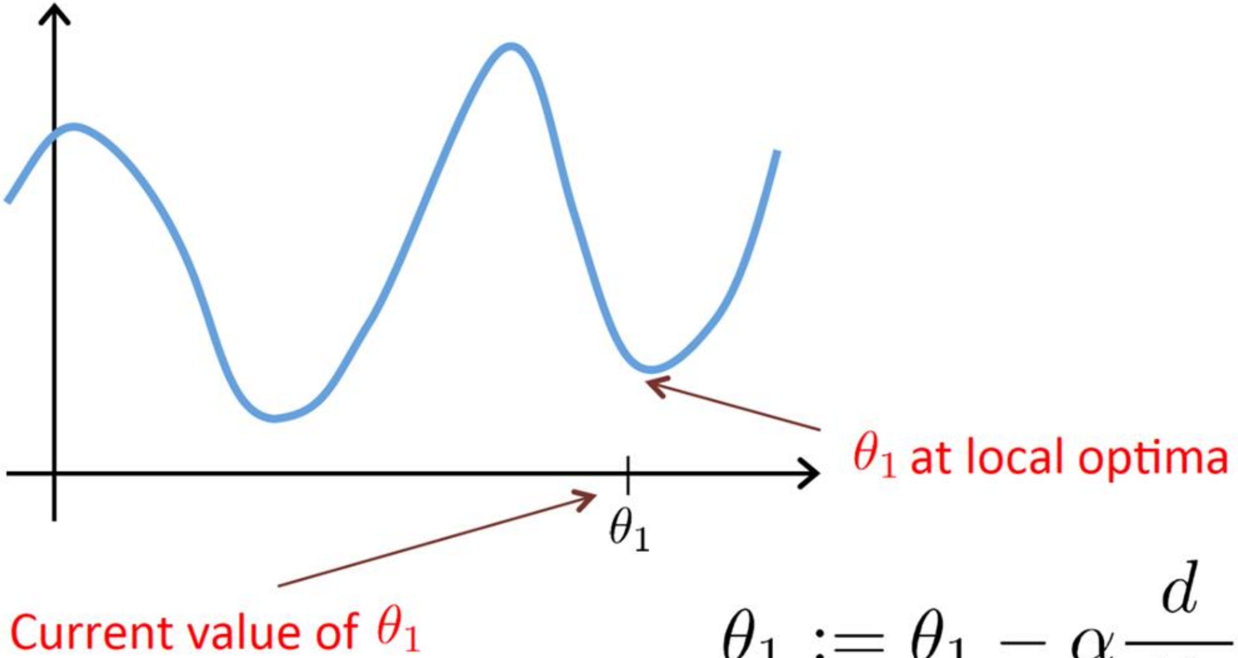


If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



Source: Andrew Ng, Machine Learning Course, Coursera

Andrew Ng



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

Source: Andrew Ng, Machine Learning Course, Coursera

Find out the values of regression coefficients for next two iterations of Gradient Descent. Take initial values of coefficients as 0. Also, find out the cost at each iteration. Take alpha = 0.01.

x	y
1	0.85
2	1.20
3	1.55
4	1.9

Iteration 1

repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \qquad \text{for j := 0..n}$$

}

$$\theta_0 = 0 \text{ and } \theta_1 = 0$$

$$\theta_0 = \theta_0 - \frac{0.01}{4}[(0 - 0.85)(1) + (0 - 1.2)(1) + (0 - 1.55)(1) + (0 - 1.9)(1)]$$

$$\theta_0 = 0 - \frac{0.01}{4}[-0.85 - 1.2 - 1.55 - 1.9]$$

$$\theta_0 = 0 - (-0.01375)$$

$$\theta_0 = 0.01375$$

Iteration 1:updating theta_1

repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \qquad \text{for j := 0..n}$$

}

$$\theta_0 = 0 \text{ and } \theta_1 = 0$$

$$\theta_1 = \theta_1 - \frac{0.01}{4}[(0-0.85)(1) + (0-1.2)(2) + (0-1.55)(3) + (0-1.9)(4)]$$

$$\theta_1 = 0 - \frac{0.01}{4}[-0.85 - 2.4 - 4.65 - 7.6]$$

$$\theta_1 = 0 - (-0.03875)$$

$$\theta_1 = 0.03875$$

Calculating Cost

J(0.01375,0.03875) = ?

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

~0.86 (approx.)

Iteration 2

repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \qquad \text{for j := 0..n}$$

}

$$\theta_0 = 0.01375 \text{ and } \theta_1 = 0.03875$$

$$\theta_0 = \theta_0 - \frac{0.01}{4}[(0.0525 - 0.85)(1) + (0.09125 - 1.2)(1) + (0.13 - 1.55)(1) + (0.1685 - 1.9)(1)]$$

$$\theta_0 = 0.01375 - \frac{0.01}{4}[-0.7975 - 1.10875 - 1.42 - 1.73125]$$

$$\theta_0 = 0.01375 - (-0.01264)$$

$$\theta_0 = 0.026393$$

Iteration 2

repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \qquad \text{for j := 0..n}$$

}

$$\theta_0 = 0.01375 \text{ and } \theta_1 = 0.03875$$

$$\theta_1 = \theta_1 - \frac{0.01}{4}[(0.0525 - 0.85)(1) + (0.09125 - 1.2)(2) + (0.13 - 1.55)(3) + (0.1685 - 1.9)(4)]$$

$$\theta_1 = 0.03875 - \frac{0.01}{4}[-0.7975 - 2.2175 - 4.26 - 6.925]$$

$$\theta_1 = 0.03875 - (-0.0355)$$

$$\theta_1 = 0.4225$$

Calculating Cost

J(0.026393,0.4225) = ?

0.046

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

Vectorised Notation for Gradient Descent

$$\theta = \theta - \frac{\alpha}{m} X^T \left(X\theta - \overrightarrow{y} \right)$$

Solving previous example using vectorized notation

Initially:

$$X = \begin{array}{c} \begin{matrix} x_0 & x_1 \end{matrix} \\ \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix} \end{array} \qquad y = \begin{array}{c} y \\ \begin{pmatrix} 0.85 \\ 1.2 \\ 1.55 \\ 1.9 \end{pmatrix} \end{array} \qquad \theta = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Updating theta by placing these values in the equation given below:

$$\theta = \theta - \frac{\alpha}{m} X^T \left(X\theta - \overrightarrow{y} \right)$$

First Iteration

$$\theta = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \left(\frac{0.01}{4}\right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.85 \\ 1.2 \\ 1.55 \\ 1.9 \end{bmatrix} \right)$$

$$\theta = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \left(\frac{0.01}{4}\right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.85 \\ 1.2 \\ 1.55 \\ 1.9 \end{bmatrix} \right)$$

$$\theta = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \left(\frac{0.01}{4}\right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} -0.85 \\ -1.2 \\ -1.55 \\ -1.9 \end{bmatrix}$$

$$\theta = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - (0.0025) \begin{bmatrix} -5.5 \\ -15.5 \end{bmatrix}$$

$$\theta = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -0.01375 \\ -0.03875 \end{bmatrix}$$

$$\theta = \begin{bmatrix} 0.01375 \\ 0.03875 \end{bmatrix}$$

Second Iteration

$$\theta = \begin{bmatrix} 0.01375 \\ 0.03875 \end{bmatrix} - \left(\frac{0.01}{4}\right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix} \left(\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 0.01375 \\ 0.03875 \end{bmatrix} - \begin{bmatrix} 0.85 \\ 1.2 \\ 1.55 \\ 1.9 \end{bmatrix} \right)$$

$$\theta = \begin{bmatrix} 0.01375 \\ 0.03875 \end{bmatrix} - \left(\frac{0.01}{4}\right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \end{bmatrix} \left(\begin{bmatrix} 0.0525 \\ 0.09125 \\ 0.13 \\ 0.16875 \end{bmatrix} - \begin{bmatrix} 0.85 \\ 1.2 \\ 1.55 \\ 1.9 \end{bmatrix} \right) \quad \theta = \begin{bmatrix} 0.01375 \\ 0.03875 \end{bmatrix} - \left(\frac{0.01}{4}\right) \begin{bmatrix} -5.0575 \\ -14.2 \end{bmatrix}$$

$$\theta = \begin{bmatrix} 0.01375 \\ 0.03875 \end{bmatrix} - (0.0025) \begin{bmatrix} -5.0575 \\ -14.2 \end{bmatrix} \quad \theta = \begin{bmatrix} 0.01375 \\ 0.03875 \end{bmatrix} - \begin{bmatrix} -0.01264 \\ -0.0355 \end{bmatrix} \quad \theta = \begin{bmatrix} 0.02639 \\ 0.07425 \end{bmatrix}$$

Polynomial Regression

Polynomial Regression

Our hypothesis function need not be linear (a straight line) if that does not fit the data well.

We can change the behavior or curve of our hypothesis function by making it a quadratic, cubic or square root function (or any other form).

For example, if our hypothesis function is $h_\theta(x) = \theta_0 + \theta_1 x_1$

then we can create additional features based on x_1, to get the quadratic function

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2$$

or the cubic function

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^3$$

In the cubic version, we have created new features x_2 and x_3 where $x_2 = x_1^2$ and $x_3 = x_1^3$