3. Multiple linear regression and Gradient descent

Shabana K M
PhD Research Scholar
Computer Science and Engineering

IIT Palakkad

14 August 2021

INDIAN INSTITUTE
OF TECHNOLOGY
PALAKKAD

Recap

Recap

- Regression

Recap

- Regression
  - dependent, independent variables

Recap

- Regression
  - dependent, independent variables
  - loss functions

Recap

- Regression
  - dependent, independent variables
  - loss functions
- Different regression models

Recap

- Regression
  - dependent, independent variables
  - loss functions
- Different regression models
  - simple, multiple

Recap

- Regression
  - dependent, independent variables
  - loss functions
- Different regression models
  - simple, multiple
  - linear, polynomial, non-linear, logistic
- Linear regression

Recap

- Regression
  - dependent, independent variables
  - loss functions
- Different regression models
  - simple, multiple
  - linear, polynomial, non-linear, logistic
- Linear regression
  - intercept, regression coefficients(weights)

Recap

- Regression
  - dependent, independent variables
  - loss functions
- Different regression models
  - simple, multiple
  - linear, polynomial, non-linear, logistic
- Linear regression
  - intercept, regression coefficients(weights)
- Simple linear regression

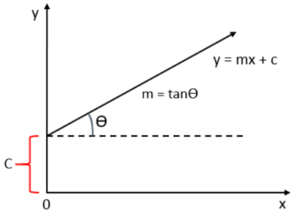
Recap

- Regression
  - dependent, independent variables
  - loss functions
- Different regression models
  - simple, multiple
  - linear, polynomial, non-linear, logistic
- Linear regression
  - intercept, regression coefficients(weights)
- Simple linear regression
  - Ordinary least squares

Recap

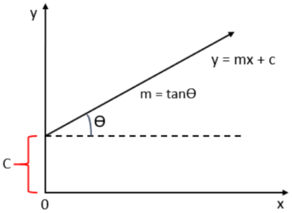
- Regression
  - dependent, independent variables
  - loss functions
- Different regression models
  - simple, multiple
  - linear, polynomial, non-linear, logistic
- Linear regression
  - intercept, regression coefficients(weights)
- Simple linear regression
  - Ordinary least squares
  - Interpreting the regression coefficients

Revisiting simple linear regression

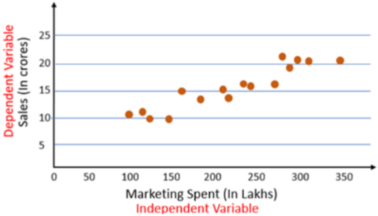


(a) A simple linear regression model

Revisiting simple linear regression

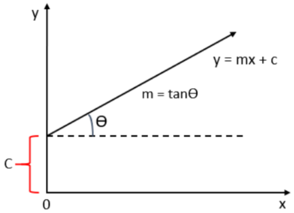


(a) A simple linear regression model

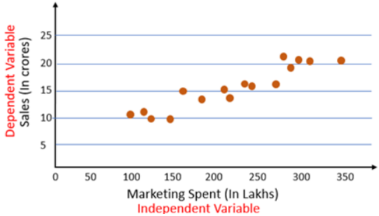


(b) Training dataset

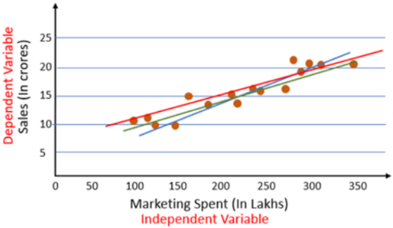
Revisiting simple linear regression



(a) A simple linear regression model

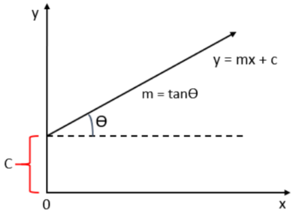


(b) Training dataset

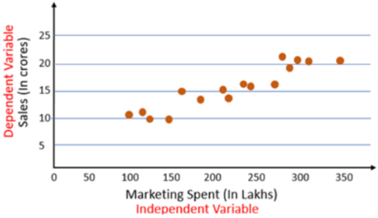


(c) Fitting a linear regression model

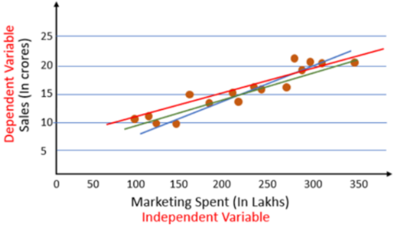
Revisiting simple linear regression



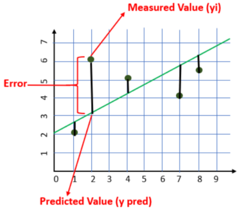
(a) A simple linear regression model



(b) Training dataset



(c) Fitting a linear regression model



(d) Loss function for the model

Multiple regression

- more than one independent variable: $x_1, x_2, ..., x_D$

Multiple regression

- more than one independent variable: $x_1, x_2, ..., x_D$
- the model is of the form

$$\hat{y} = f(w, x) = w_0 + w_1 x_1 + ... + w_D x_D$$

Multiple regression

- more than one independent variable: $x_1, x_2, ..., x_D$
- the model is of the form

$$\hat{y} = f(w, x) = w_0 + w_1 x_1 + ... + w_D x_D$$

- when $D = 2$, the model is of the form $\hat{y} = w_0 + w_1 x_1 + w_2 x_2$
  - describes a plane in the three dimensional space of $\hat{y}, x_1$ and x_2 with w_0 as the intercept of the plane

Multiple regression

- more than one independent variable: $x_1, x_2, ..., x_D$
- the model is of the form

$$\hat{y} = f(w, x) = w_0 + w_1 x_1 + ... + w_D x_D$$

- when $D = 2$, the model is of the form $\hat{y} = w_0 + w_1 x_1 + w_2 x_2$
    - describes a plane in the three dimensional space of $\hat{y}, x_1$ and x_2 with w_0 as the intercept of the plane
- regression coefficient w_i measures the association between the predictor variable x_i and the outcome y

Multiple regression

- more than one independent variable: $x_1, x_2, ..., x_D$
- the model is of the form

$$\hat{y} = f(w, x) = w_0 + w_1 x_1 + ... + w_D x_D$$

- when $D = 2$, the model is of the form $\hat{y} = w_0 + w_1 x_1 + w_2 x_2$
  - describes a plane in the three dimensional space of $\hat{y}, x_1$ and x_2 with w_0 as the intercept of the plane
- regression coefficient w_i measures the association between the predictor variable x_i and the outcome y
  - w_i represents the mean change in y corresponding to a unit increase in x_i when all other predictors are held fixed

Multiple regression

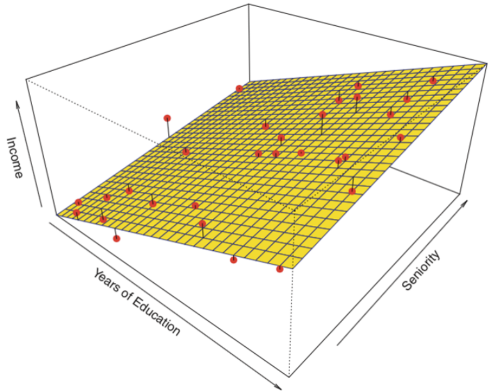


Figure: Visualizing the multiple regression model for predicting the response variable `Income` (y) based on the independent variables `Seniority` ($x1$) and `Years of Education` ($x2$)

Problem definition

Given: Training data set comprising N observations $(x_n, y_n)_{n=1}^{N}$, where $x_n = [x_{n1}, x_{n2}, ..., x_{nD}]$ is the input and y_n is the corresponding output

Problem definition

Given: Training data set comprising N observations $(x_n, y_n)_{n=1}^{N}$, where $x_n = [x_{n1}, x_{n2}, ..., x_{nD}]$ is the input and y_n is the corresponding output

Goal: Predict the y value for a new value of x

Problem definition

Given: Training data set comprising N observations $(x_n, y_n)_{n=1}^{N}$, where $x_n = [x_{n1}, x_{n2}, ..., x_{nD}]$ is the input and y_n is the corresponding output

Goal: Predict the y value for a new value of x

Estimate: The weights $w = [w_0, w_1, ..., w_D]$

Problem definition

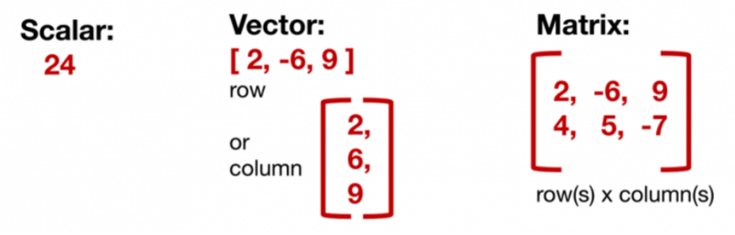
Given: Training data set comprising N observations $(x_n, y_n)_{n=1}^N$, where $x_n = [x_{n1}, x_{n2}, ..., x_{nD}]$ is the input and y_n is the corresponding output

Goal: Predict the y value for a new value of x

Estimate: The weights $w = [w_0, w_1, ..., w_D]$

Minimize: `Mean-squared error`: $l(w) = \frac{1}{N}\sum_{i=1}^N (y_i - \hat{y}_i)^2$
where $\hat{y}_i = f(w, x_i) = w_0 + w_1 x_1 + ... + w_D x_D$

A small mathematics refresher

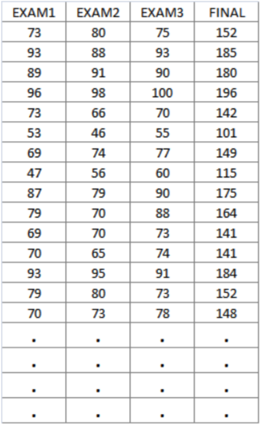


Scalar: a single number

Vector: an ordered array of numbers
can be in a row or a column
an index points to a specific value within the vector

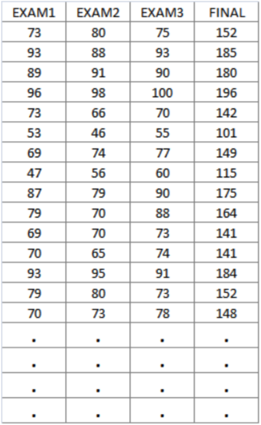
Matrix: two dimensional array of numbers
each element identified by two numbers

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

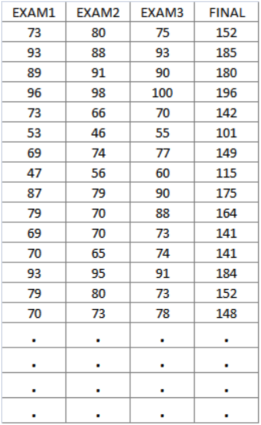
Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

Consider the problem of predicting the Final exam score (y) based on the scores obtained in the first three exams (x_1, x_2, x_3)

Multiple regression



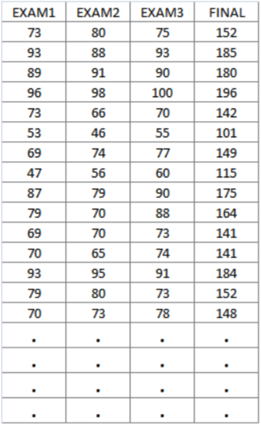
EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

Consider the problem of predicting the Final exam score (y) based on the scores obtained in the first three exams (x_1, x_2, x_3)

- Using a linear regression model for this problem, we have:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

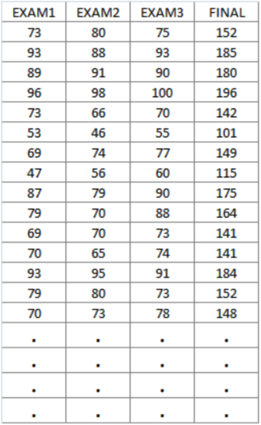
Consider the problem of predicting the Final exam score (y) based on the scores obtained in the first three exams (x_1, x_2, x_3)

- Using a linear regression model for this problem, we have:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

- x_{ij} - j^{th} feature of the i^{th} observation

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

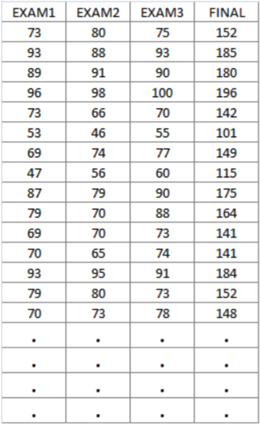
Consider the problem of predicting the Final exam score (y) based on the scores obtained in the first three exams (x_1, x_2, x_3)

- Using a linear regression model for this problem, we have:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

- x_{ij} - j^{th} feature of the i^{th} observation
    - $x_{13} = 75$

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

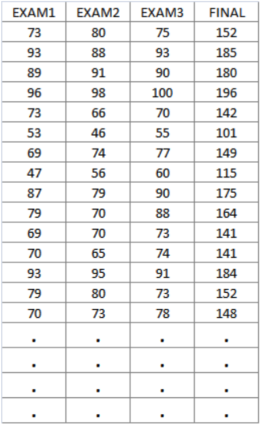
Consider the problem of predicting the Final exam score (y) based on the scores obtained in the first three exams (x_1, x_2, x_3)

- Using a linear regression model for this problem, we have:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

- x_{ij} - j^{th} feature of the i^{th} observation
  - $x_{13} = 75$
  - $x_{42} =$

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

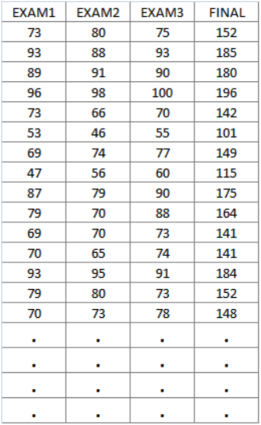
Consider the problem of predicting the Final exam score (y) based on the scores obtained in the first three exams (x_1, x_2, x_3)

- Using a linear regression model for this problem, we have:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

- x_{ij} - j^{th} feature of the i^{th} observation
  - ○ $x_{13} = 75$
  - ○ $x_{42} = 98$

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

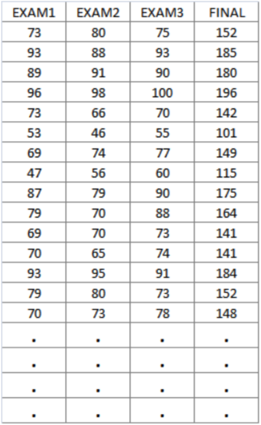
Consider the problem of predicting the Final exam score (y) based on the scores obtained in the first three exams (x_1, x_2, x_3)

- Using a linear regression model for this problem, we have:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

- x_{ij} - j^{th} feature of the i^{th} observation
  - $x_{13} = 75$
  - $x_{42} = 98$
- y_i -

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

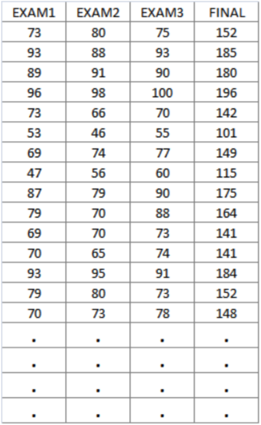
Consider the problem of predicting the Final exam score (y) based on the scores obtained in the first three exams (x_1, x_2, x_3)

- Using a linear regression model for this problem, we have:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

- x_{ij} - j^{th} feature of the i^{th} observation
  - $x_{13} = 75$
  - $x_{42} = 98$
- y_i - y value of the i^{th} observation

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

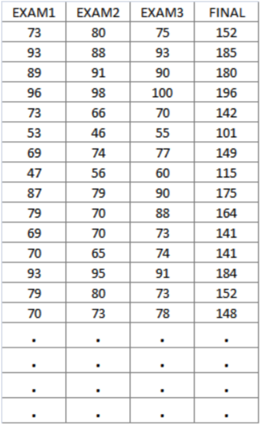
Consider the problem of predicting the Final exam score (y) based on the scores obtained in the first three exams (x_1, x_2, x_3)

- Using a linear regression model for this problem, we have:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

- x_{ij} - j^{th} feature of the i^{th} observation
  - $x_{13} = 75$
  - $x_{42} = 98$
- y_i - y value of the i^{th} observation
  - $y_1 =$

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

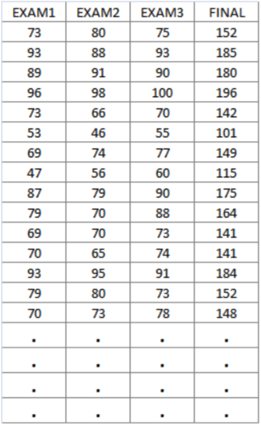
Consider the problem of predicting the Final exam score (y) based on the scores obtained in the first three exams (x_1, x_2, x_3)

- Using a linear regression model for this problem, we have:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

- x_{ij} - j^{th} feature of the i^{th} observation
  - $x_{13} = 75$
  - $x_{42} = 98$
- y_i - y value of the i^{th} observation
  - $y_1 = 152$
  - $y_4 =$

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

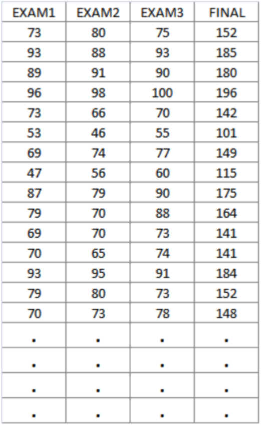
Consider the problem of predicting the Final exam score (y) based on the scores obtained in the first three exams (x_1, x_2, x_3)

- Using a linear regression model for this problem, we have:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

- x_{ij} - j^{th} feature of the i^{th} observation
  - $x_{13} = 75$
  - $x_{42} = 98$
- y_i - y value of the i^{th} observation
  - $y_1 = 152$
  - $y_4 = 196$

Multiple regression

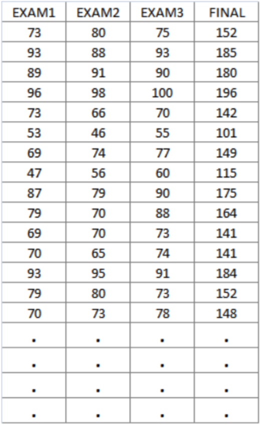


EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

Multiple regression

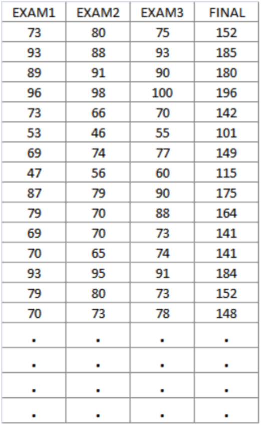
Our model:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

Our model:

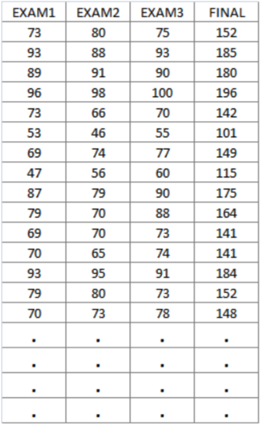
$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

We can write:

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + e$$

where $e = y - \hat{y}$

Multiple regression



EXAM1	EXAM2	EXAM3	FINAL
73	80	75	152
93	88	93	185
89	91	90	180
96	98	100	196
73	66	70	142
53	46	55	101
69	74	77	149
47	56	60	115
87	79	90	175
79	70	88	164
69	70	73	141
70	65	74	141
93	95	91	184
79	80	73	152
70	73	78	148
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

Our model:

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

We can write:

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + e$$

where $e = y - \hat{y}$

Now we have:

$$\begin{aligned} y_1 &= w_0 + w_1 x_{11} + w_2 x_{12} + w_3 x_{13} + e_1 \\ y_2 &= w_0 + w_1 x_{21} + w_2 x_{22} + w_3 x_{23} + e_2 \\ &\vdots \\ y_N &= w_0 + w_1 x_{N1} + w_2 x_{N2} + w_3 x_{N3} + e_N \end{aligned}$$

Matrix Representation

$$\begin{aligned} y_1 &= w_0 + w_1 x_{11} + w_2 x_{12} + w_3 x_{13} + e_1 \\ y_2 &= w_0 + w_1 x_{21} + w_2 x_{22} + w_3 x_{23} + e_2 \\ &\vdots \\ y_N &= w_0 + w_1 x_{N1} + w_2 x_{N2} + w_3 x_{N3} + e_N \end{aligned}$$

Matrix Representation

$$
\begin{aligned}
y_1 &= w_0 + w_1 x_{11} + w_2 x_{12} + w_3 x_{13} + e_1 \\
y_2 &= w_0 + w_1 x_{21} + w_2 x_{22} + w_3 x_{23} + e_2 \\
&\vdots \\
y_N &= w_0 + w_1 x_{N1} + w_2 x_{N2} + w_3 x_{N3} + e_N
\end{aligned}
$$

$$
y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}
X = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ \vdots & & & \\ 1 & x_{N1} & x_{N2} & x_{N3} \end{bmatrix}
w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}
e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}
$$

Matrix Representation

$$
\begin{aligned}
y_1 &= w_0 + w_1 x_{11} + w_2 x_{12} + w_3 x_{13} + e_1 \\
y_2 &= w_0 + w_1 x_{21} + w_2 x_{22} + w_3 x_{23} + e_2 \\
&\vdots \\
y_N &= w_0 + w_1 x_{N1} + w_2 x_{N2} + w_3 x_{N3} + e_N
\end{aligned}
$$

$$
y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}
X = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ \vdots & & & \\ 1 & x_{N1} & x_{N2} & x_{N3} \end{bmatrix}
w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}
e = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}
$$

The N equations can be written as:

$$y = Xw + e$$

Mean Square Error (MSE)

$$y = Xw + e$$

Mean Square Error (MSE)

$$y = Xw + e$$

Rearranging the terms, we get

$$e = y - Xw$$

Mean Square Error (MSE)

$$y = Xw + e$$

Rearranging the terms, we get

$$e = y - Xw$$

$$MSE = \frac{\sum_{i}^{N} e_i^2}{N}$$

Mean Square Error (MSE)

$$y = Xw + e$$

Rearranging the terms, we get

$$e = y - Xw$$

$$MSE = \frac{\sum_i^N e_i^2}{N}$$

$$\sum_i^N e^2 = e_1^2 + e_2^2 + ... + e_N^2$$

$$= \begin{bmatrix} e_1 & e_2 & \dots & e_N \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = e^T e = (y - Xw)^T (y - Xw)$$

Computing *w* using Ordinary Least Squares (OLS)

$$MSE = \frac{\sum_i^N e_i^2}{N} = \frac{(y - Xw)^T(y - Xw)}{N}$$

Computing *w* using Ordinary Least Squares (OLS)

$$MSE = \frac{\sum_i^N e_i^2}{N} = \frac{(y - Xw)^T(y - Xw)}{N}$$

$$= \frac{1}{N}(y^Ty - y^TXw - w^TX^Ty + w^TX^TXw)$$

Computing w using Ordinary Least Squares (OLS)

$$MSE = \frac{\sum_i^N e_i^2}{N} = \frac{(y - Xw)^T(y - Xw)}{N}$$

$$= \frac{1}{N}(y^Ty - y^TXw - w^TX^Ty + w^TX^TXw)$$

Now $y^TXw = (w^TX^Ty)^T$ is a scalar, so we have

Computing *w* using Ordinary Least Squares (OLS)

$$MSE = \frac{\sum_i^N e_i^2}{N} = \frac{(y - Xw)^T(y - Xw)}{N}$$

$$= \frac{1}{N}(y^T y - y^T Xw - w^T X^T y + w^T X^T Xw)$$

Now $y^T Xw = (w^T X^T y)^T$ is a scalar, so we have

$$MSE = \frac{1}{N}(y^T y - 2w^T X^T y + w^T X^T Xw)$$

Computing *w* using Ordinary Least Squares (OLS)

$$MSE = \frac{\sum_i^N e_i^2}{N} = \frac{(y - Xw)^T(y - Xw)}{N}$$

$$= \frac{1}{N}(y^T y - y^T Xw - w^T X^T y + w^T X^T Xw)$$

Now $y^T Xw = (w^T X^T y)^T$ is a scalar, so we have

$$MSE = \frac{1}{N}(y^T y - 2w^T X^T y + w^T X^T Xw)$$

Using OLS, the regression coefficients w are estimated by solving the following minimization problem:

$$\min_w \frac{1}{N}(y^T y - 2w^T X^T y + w^T X^T Xw)$$

Computing *w* using Ordinary Least Squares (OLS)

$$l(w) = \frac{1}{N}(y^T y - 2w^T X^T y + w^T X^T X w)$$

Computing *w* using Ordinary Least Squares (OLS)

$$l(w) = \frac{1}{N}(y^T y - 2w^T X^T y + w^T X^T X w)$$

The error is minimized for the value of w such that $\frac{\partial l(w)}{\partial w} = 0$

Computing w using Ordinary Least Squares (OLS)

$$l(w) = \frac{1}{N}(y^T y - 2w^T X^T y + w^T X^T X w)$$

The error is minimized for the value of w such that $\frac{\partial l(w)}{\partial w} = 0$

$$-2 * X^T y + 2 * X^T X w = 0$$

Computing w using Ordinary Least Squares (OLS)

$$l(w) = \frac{1}{N}(y^T y - 2w^T X^T y + w^T X^T X w)$$

The error is minimized for the value of w such that $\frac{\partial l(w)}{\partial w} = 0$

$$-2 * X^T y + 2 * X^T X w = 0$$

$$w = (X^T X)^{-1}(X^T y)$$

Computing w using Ordinary Least Squares (OLS)

$$l(w) = \frac{1}{N}(y^T y - 2w^T X^T y + w^T X^T X w)$$

The error is minimized for the value of w such that $\frac{\partial l(w)}{\partial w} = 0$

$$-2 * X^T y + 2 * X^T X w = 0$$

$$w = (X^T X)^{-1}(X^T y)$$

Normal equation

- $w = (X^T X)^{-1}(X^T y)$

Normal equation

$$w = (X^T X)^{-1}(X^T y)$$

☺

- gives the solution in one go

Normal equation

$$w = (X^T X)^{-1}(X^T y)$$

☺

- gives the solution in one go
- works well when using small feature sets

Normal equation

$$w = (X^T X)^{-1}(X^T y)$$

☺

- gives the solution in one go
- works well when using small feature sets

☹

- numerical complexity
  - need to compute $(X^T X)^{-1}$ - can be very slow

Normal equation

$$w = (X^T X)^{-1}(X^T y)$$

☺

- gives the solution in one go
- works well when using small feature sets

☹

- numerical complexity
  - need to compute $(X^T X)^{-1}$ - can be very slow
- $X^T X$ may not be always invertible

Normal equation

$$w = (X^TX)^{-1}(X^Ty)$$

☺

- gives the solution in one go
- works well when using small feature sets

☹

- numerical complexity
  - need to compute $(X^TX)^{-1}$ - can be very slow
- X^TX may not be always invertible
  - too many features (eg:- $N < D$)

Normal equation

$$w = (X^T X)^{-1}(X^T y)$$

☺

- gives the solution in one go
- works well when using small feature sets

☹

- numerical complexity
  - need to compute $(X^T X)^{-1}$ - can be very slow
- $X^T X$ may not be always invertible
  - too many features (eg:- $N < D$)
  - some columns are linearly dependent (redundant features)

Normal equation

$$w = (X^T X)^{-1}(X^T y)$$

☺

- gives the solution in one go
- works well when using small feature sets

☹

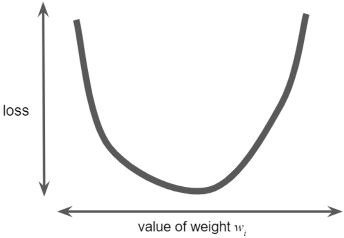
- numerical complexity
  - need to compute $(X^T X)^{-1}$ - can be very slow
- $X^T X$ may not be always invertible
  - too many features (eg:- $N < D$)
  - some columns are linearly dependent (redundant features)
- cannot be easily parallelized

Loss function for simple linear regression

Consider a simple linear regression problem. If we compute the cost/error function for all possible values of w_1, the resulting plot will always be convex, or kind of bowl-shaped

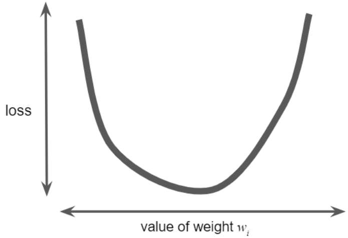
Loss function for simple linear regression

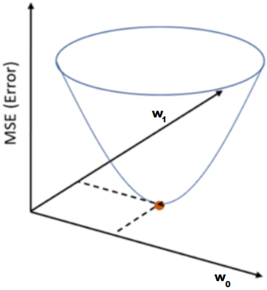
Consider a simple linear regression problem. If we compute the cost/error function for all possible values of w_1, the resulting plot will always be convex, or kind of bowl-shaped



Loss function for simple linear regression

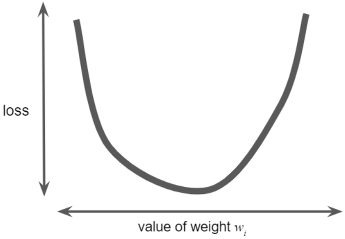
Consider a simple linear regression problem. If we compute the cost/error function for all possible values of w_1, the resulting plot will always be convex, or kind of bowl-shaped

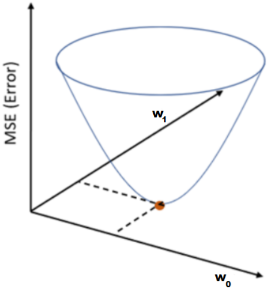




Loss function for simple linear regression

Consider a simple linear regression problem. If we compute the cost/error function for all possible values of w_1, the resulting plot will always be convex, or kind of bowl-shaped

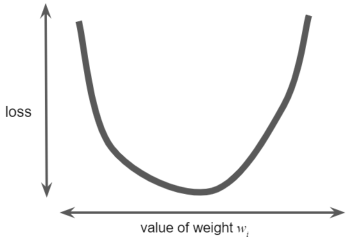


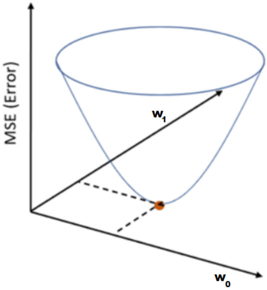


- Convex problems have only one minimum; i.e., only one place where the slope is exactly 0

Loss function for simple linear regression

Consider a simple linear regression problem. If we compute the cost/error function for all possible values of w_1, the resulting plot will always be convex, or kind of bowl-shaped

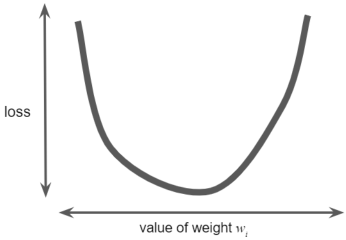


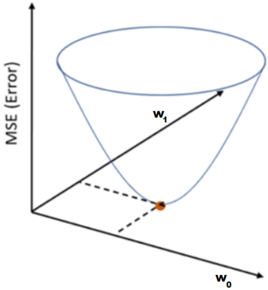


- Convex problems have only one minimum; i.e., only one place where the slope is exactly 0 - convergence point

Loss function for simple linear regression

Consider a simple linear regression problem. If we compute the cost/error function for all possible values of w_1, the resulting plot will always be convex, or kind of bowl-shaped

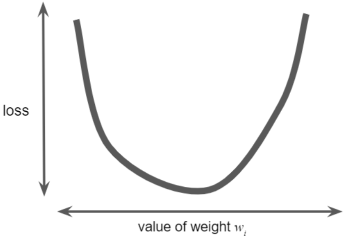


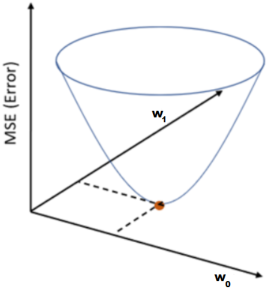


- Convex problems have only one minimum; i.e., only one place where the slope is exactly 0 - convergence point
- Compute the loss function for all possible values of w_1 over the entire data set and then find the minimum

Loss function for simple linear regression

Consider a simple linear regression problem. If we compute the cost/error function for all possible values of w_1, the resulting plot will always be convex, or kind of bowl-shaped





- Convex problems have only one minimum; i.e., only one place where the slope is exactly 0 - convergence point
- Compute the loss function for all possible values of w_1 over the entire data set and then find the minimum - inefficient!

Gradient Descent

Gradient descent

- iterative optimization algorithm to find the minimum of a function

Gradient Descent

Gradient descent

- iterative optimization algorithm to find the minimum of a function
  loss function in our case!!

Gradient Descent

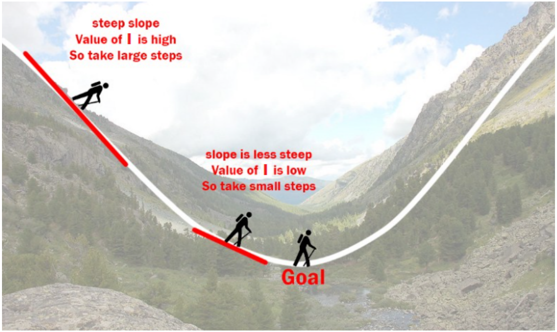
Gradient descent

- iterative optimization algorithm to find the minimum of a function
  loss function in our case!!
- idea: follow the gradients of the loss function

Gradient Descent

Gradient descent

- iterative optimization algorithm to find the minimum of a function
  loss function in our case!!
- idea: follow the gradients of the loss function



Gradient

- used for functions with several inputs and a single output

Gradient

- used for functions with several inputs and a single output
- vector of partial derivatives with respect to each of its inputs

Gradient

- used for functions with several inputs and a single output
- vector of partial derivatives with respect to each of its inputs
- points in the direction of greatest increase of a function

Gradient

- used for functions with several inputs and a single output
- vector of partial derivatives with respect to each of its inputs
- points in the direction of greatest increase of a function
- is zero at a local maximum or local minimum

Gradient

- used for functions with several inputs and a single output
- vector of partial derivatives with respect to each of its inputs
- points in the direction of greatest increase of a function
- is zero at a local maximum or local minimum
- $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \cdots \end{bmatrix}^T$

Gradient

- used for functions with several inputs and a single output
- vector of partial derivatives with respect to each of its inputs
- points in the direction of greatest increase of a function
- is zero at a local maximum or local minimum
- $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \cdots \end{bmatrix}^T$

Let $f(x, y) = ax^2 + by^2$,

then $\nabla f =$

Gradient

- used for functions with several inputs and a single output
- vector of partial derivatives with respect to each of its inputs
- points in the direction of greatest increase of a function
- is zero at a local maximum or local minimum
- $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \cdots \end{bmatrix}^T$

Let $f(x, y) = ax^2 + by^2$,

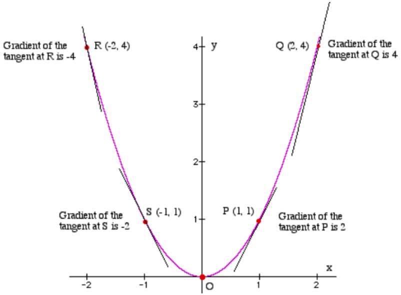
then $\nabla f = \begin{bmatrix} 2ax \\ 2by \end{bmatrix}$

Gradient

- used for functions with several inputs and a single output
- vector of partial derivatives with respect to each of its inputs
- points in the direction of greatest increase of a function
- is zero at a local maximum or local minimum
- $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \cdots \end{bmatrix}^T$

Let $f(x, y) = ax^2 + by^2$,

then $\nabla f = \begin{bmatrix} 2ax \\ 2by \end{bmatrix}$



Gradient Descent

AIM: Find the value of w_1 corresponding to the minimum value of the loss function

Gradient Descent

AIM: Find the value of w_1 corresponding to the minimum value of the loss function

BASIC IDEA: Start with an initial parameter value of w_1 and iteratively move towards a new value of w_1 that minimizes the loss function

Gradient Descent

AIM: Find the value of w_1 corresponding to the minimum value of the loss function

BASIC IDEA: Start with an initial parameter value of w_1 and iteratively move towards a new value of w_1 that minimizes the loss function

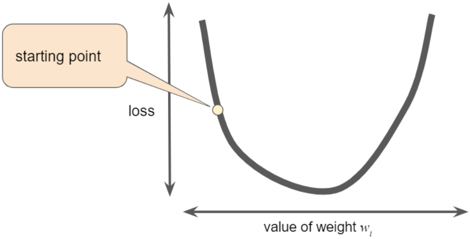
- can set w_1 to 0 or some random value

Gradient Descent

AIM: Find the value of w_1 corresponding to the minimum value of the loss function

BASIC IDEA: Start with an initial parameter value of w_1 and iteratively move towards a new value of w_1 that minimizes the loss function

- can set w_1 to 0 or some random value



Gradient Descent:Determining the next value of w_1

1. **Calculate the gradient of the loss curve at the starting point**

Gradient Descent:Determining the next value of w_1

1. **Calculate the gradient of the loss curve at the starting point**
   - the gradient ∇f points in the direction of the greatest rate of increase of the function

Gradient Descent:Determining the next value of w_1

1. **Calculate the gradient of the loss curve at the starting point**
    - the gradient ∇f points in the direction of the greatest rate of increase of the function
        - its magnitude is the slope of the graph in that direction

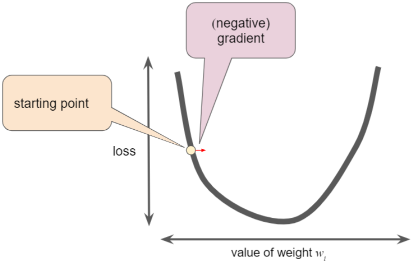
Gradient Descent:Determining the next value of w_1

1. **Calculate the gradient of the loss curve at the starting point**
   - the gradient ∇f points in the direction of the greatest rate of increase of the function
     - its magnitude is the slope of the graph in that direction
   - the negative of the gradient $(-\nabla f)$ points in the direction of maximum decrease in height

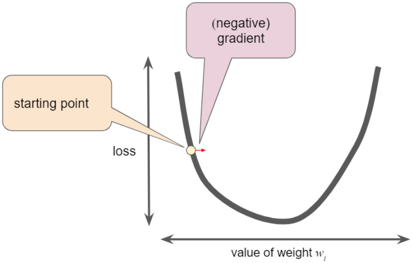
Gradient Descent:Determining the next value of w_1

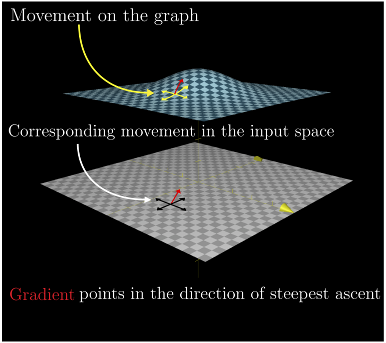
1. **Calculate the gradient of the loss curve at the starting point**
   - the gradient ∇f points in the direction of the greatest rate of increase of the function
     - its magnitude is the slope of the graph in that direction
   - the negative of the gradient $(-\nabla f)$ points in the direction of maximum decrease in height
   - gradient descent algorithm takes a step in the direction of the negative gradient in order to reduce loss as quickly as possible

Gradient Descent



Gradient Descent




Movement on the graph
Corresponding movement in the input space
Gradient points in the direction of steepest ascent

Gradient Descent: Determining the next value

2. Add some fraction of the gradient's magnitude to the starting point

Gradient Descent: Determining the next value

2. Add some fraction of the gradient's magnitude to the starting point

$$w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$$

Gradient Descent: Determining the next value

2. Add some fraction of the gradient's magnitude to the starting point

$$w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$$

This process is repeated until convergence

Gradient Descent: Determining the next value

2. Add some fraction of the gradient's magnitude to the starting point

$$w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$$

This process is repeated until convergence

α (`learning rate`) - controls the step size

Gradient Descent: Determining the next value

2. Add some fraction of the gradient's magnitude to the starting point

$$w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$$

This process is repeated until convergence

α (`learning rate`) - controls the step size

- high value - might step over the minimum

Gradient Descent: Determining the next value

2. Add some fraction of the gradient's magnitude to the starting point

$$w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$$

This process is repeated until convergence

α (`learning rate`) - controls the step size

- high value - might step over the minimum
- low value - slow convergence

Gradient Descent: Determining the next value

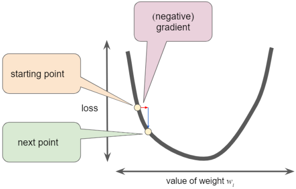
2. Add some fraction of the gradient's magnitude to the starting point

$$w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$$

This process is repeated until convergence

α (`learning rate`) - controls the step size

- high value - might step over the minimum
- low value - slow convergence



Gradient Descent: Determining the next value

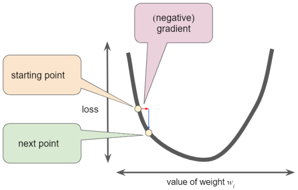
2. Add some fraction of the gradient's magnitude to the starting point

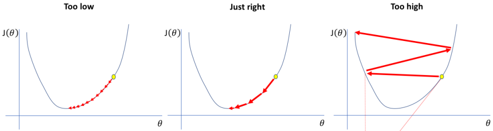
$$w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$$

This process is repeated until convergence

α (`learning rate`) - controls the step size

- high value - might step over the minimum
- low value - slow convergence



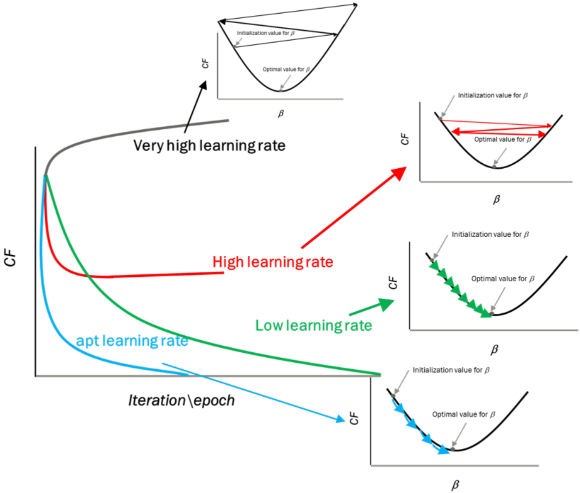


A small learning rate requires many updates before reaching the minimum point

The optimal learning rate swiftly reaches the minimum point

Too large of a learning rate causes drastic updates which lead to divergent behaviors

Choosing the learning rate



Gradient Descent for linear regression

Loss function: $l(w) = \frac{1}{N}\sum_{i=1}^{N}((\underbrace{w_0 + w_1 x_{i1} + ... + w_D x_{iD}}_{\hat{y}_i}) - y_i)^2$

Gradient Descent for linear regression

Loss function: $l(w) = \frac{1}{N}\sum_{i=1}^{N}((\underbrace{w_0 + w_1 x_{i1} + ... + w_D x_{iD}}_{\hat{y}_i}) - y_i)^2$

Gradient descent update: $w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$

Gradient Descent for linear regression

Loss function: $l(w) = \frac{1}{N}\sum_{i=1}^{N}((\underbrace{w_0 + w_1 x_{i1} + ... + w_D x_{iD}}_{\hat{y}_i}) - y_i)^2$

Gradient descent update: $w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$

Updates:

Gradient Descent for linear regression

Loss function: $l(w) = \frac{1}{N}\sum_{i=1}^{N}((\underbrace{w_0 + w_1 x_{i1} + ... + w_D x_{iD}}_{\hat{y}_i}) - y_i)^2$

Gradient descent update: $w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$

Updates:

$$w_0 := w_0 - \alpha * \frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)$$

Gradient Descent for linear regression

Loss function: $l(w) = \frac{1}{N}\sum_{i=1}^{N}((\underbrace{w_0 + w_1 x_{i1} + ... + w_D x_{iD}}_{\hat{y}_i}) - y_i)^2$

Gradient descent update: $w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$

Updates:

$$w_0 := w_0 - \alpha * \frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)$$

$$w_1 := w_1 - \alpha * \frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) * x_{i1}$$

Gradient Descent for linear regression

Loss function: $l(w) = \frac{1}{N}\sum_{i=1}^{N}((\underbrace{w_0 + w_1 x_{i1} + \ldots + w_D x_{iD}}_{\hat{y}_i}) - y_i)^2$

Gradient descent update: $w_i^{next} = w_i^{old} - \alpha * \frac{\partial l(w)}{\partial w_i}$

Updates:

$$w_0 := w_0 - \alpha * \frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)$$

$$w_1 := w_1 - \alpha * \frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) * x_{i1}$$

$$\vdots$$

$$w_D := w_D - \alpha * \frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) * x_{iD}$$

Gradient Descent for linear regression

repeat until convergence {

$$w_j = w_j - \alpha * \frac{2}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i) * x_{ij} \qquad \text{for } j := 0...D \quad \}$$

Gradient Descent for linear regression

repeat until convergence {

$$w_j = w_j - \alpha * \frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) * x_{ij} \quad \text{for } j := 0...D \quad \}$$

Matrix notation: $w := w - \alpha \nabla l(w)$

Gradient Descent for linear regression

repeat until convergence {

$$w_j = w_j - \alpha * \frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) * x_{ij} \quad \text{for } j := 0...D \quad \}$$

Matrix notation: $w := w - \alpha \nabla l(w)$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1D} \\ 1 & x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix} w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix}$$

Gradient Descent for linear regression

repeat until convergence {

$$w_j = w_j - \alpha * \frac{2}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i) * x_{ij} \qquad \text{for } j := 0 \ldots D \quad \}$$

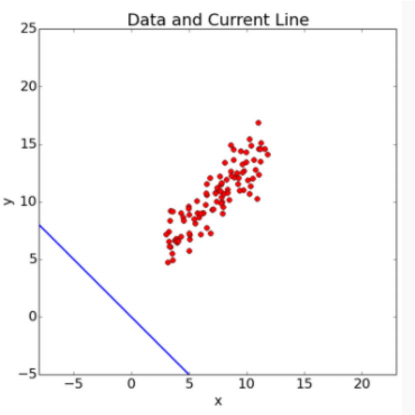
Matrix notation: $w := w - \alpha \nabla l(w)$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1D} \\ 1 & x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix} w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix}$$

The matrix notation of the Gradient Descent rule is:

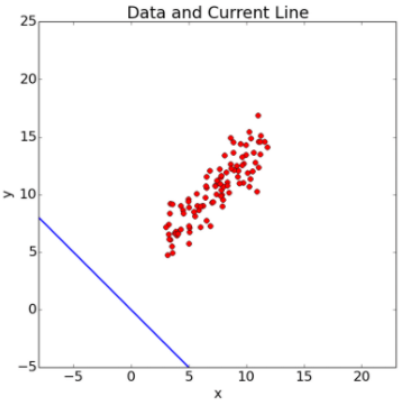
$$w := w - \frac{2\alpha}{N} X^T(Xw - y)$$

Gradient Descent: Demo

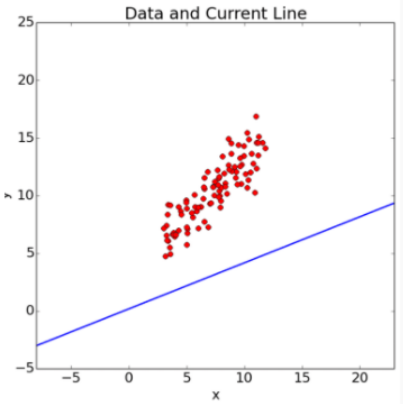


(a) Initial *w*

Gradient Descent: Demo

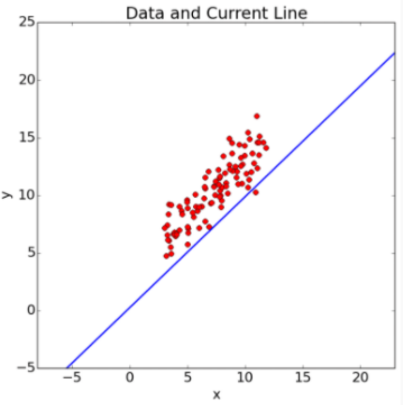


(a) Initial *w*



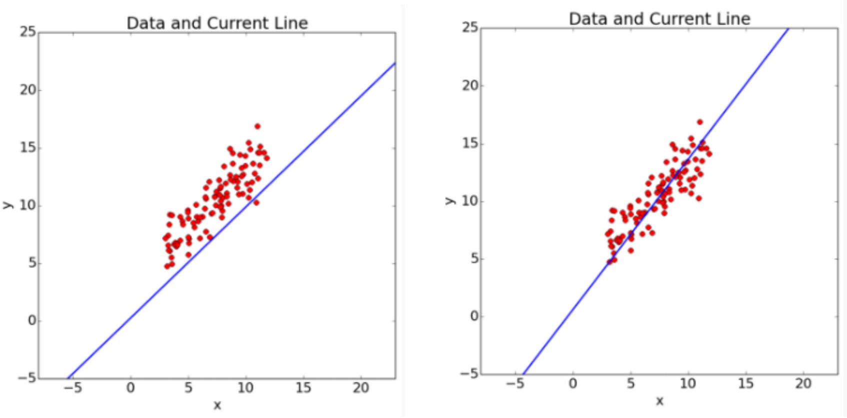
(b) After one iteration

Gradient Descent: Demo



(a) After 2 iterations

Gradient Descent: Demo



(a) After 2 iterations

(b) After 3 iterations

References

1. https://towardsdatascience.com/multiple-linear-regression-with-math-and-code-c1052f3c7446
2. https://developers.google.com/machine-learning/crash-course/reducing-loss/gradient-descent
3. https://www.coursera.org/learn/machine-learning/resources/QQx8l

Thanks Google for the pictures!