ECEN 377: Engineering Applications of AI

Dr. Mahmoud Nabil Mahmoud
mnmahmoud@ncat.edu

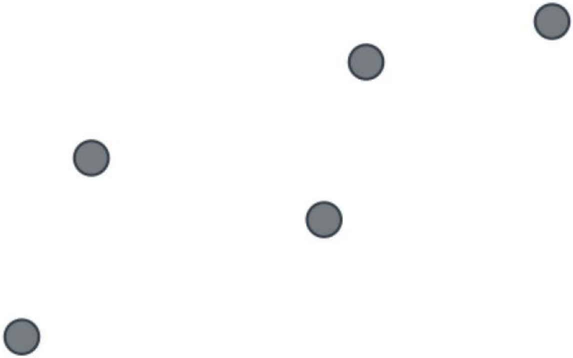
North Carolina A & T State University

September 20, 2024

Outline

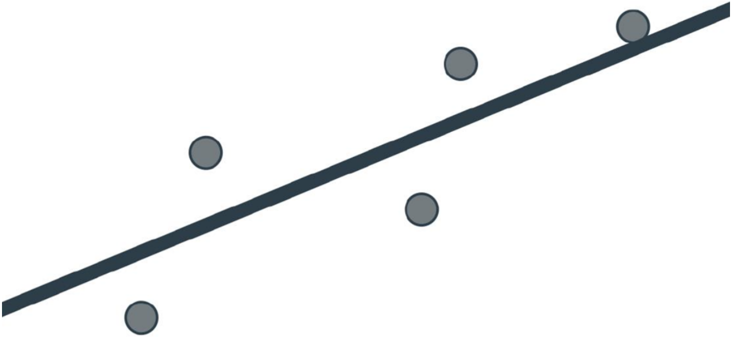
Linear Regression

Can you design a linear road that pass by all these houses equally?!



Linear Regression

Can you design a linear road that pass by all these houses equally?!



Housing prices problem

- What are the features and labels here?
- Is it classification or regression problem?
- What is the price of the house of 4 rooms?
- What is the price per extra room? (#Room $\times$ weight)
- What is the base price? (Bias)
- What is the equation that represents the price?
- Price $= 100 + 50 * (\#\text{Rooms})$

Number of rooms	Price
1	150
2	200
3	250
4	?
5	350
6	400
7	450

Outline

Weights and Bias

Price Equation

$$\text{Price} = 100 + 50 * (\#\text{Rooms})$$

WEIGHTS

Each feature gets multiplied by a corresponding factor. These factors are the weights. In the above formula the only feature is the number of rooms, and its value is 50.

BIAS

Constant that is not attached to any of the features. It is called the bias. In this model, the bias is 100 and it corresponds to the base price of a house.

Weights and Bias

Price Equation

$$\text{Price} = 100 + 50 * (\#\text{Rooms})$$

WEIGHTS

Each feature gets multiplied by a corresponding factor. These factors are the weights. In the above formula the only feature is the number of rooms, and its value is 50.

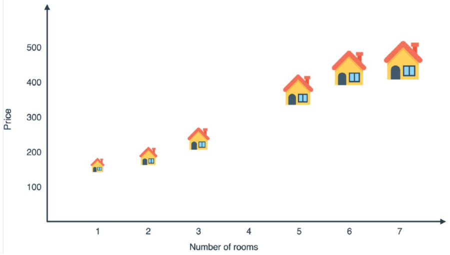
BIAS

Constant that is not attached to any of the features. It is called the bias. In this model, the bias is 100 and it corresponds to the base price of a house.

How do machines learn this equation?

Outline

How machines learn it? [Remember-Formulate-Predict]



Number of rooms	Price
1	150
2	200
3	250
4	?
5	350
6	400
7	450

How machines learn it? [Remember-Formulate-Predict]

Linear Regression

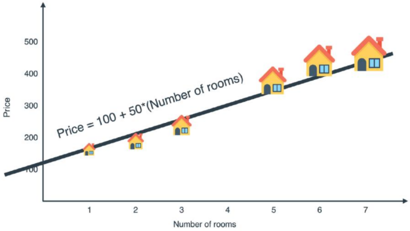
Price = 100 + 50 * (#Rooms)

Slope

Masures how steep the line is.

Y-intercept

It is the height at which the line crosses the vertical (y) axis.



Linear Equation

Linear Equation is the equation of a line. $y = mx + b$

How machines learn it? [Remember-Formulate-Predict]

Linear Regression

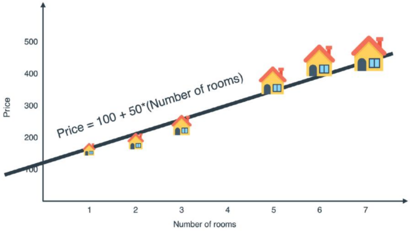
Price = 100 + 50 * (#Rooms)

Slope

Masures how steep the line is.

Y-intercept

It is the height at which the line crosses the vertical (y) axis.



Linear Equation

Linear Equation is the equation of a line. $y = mx + b$

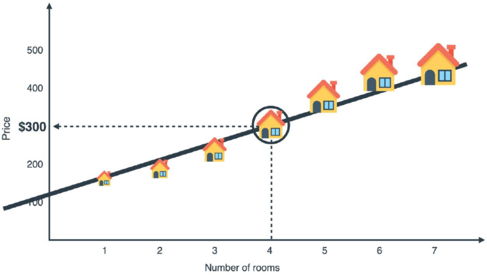
How many lines can solve the problem?

How machines learn it? [Remember-Formulate-Predict]

$$\text{Price} = 100 + 50 \times (\#\text{Rooms})$$
$$\text{Price} = 100 + 50 \times (4) = 300$$

Some Questions:

- Can we have multiple features data?
- How does computer learn this equation?



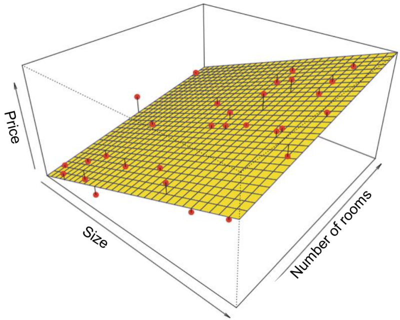
Multivariate Linear Regression

Multivariate Linear Regression

Price = 30*(#Rooms) + 1.5*(Size) + 10*(Schools Quality) - 2*(Age) + 50

What do you notice in this equation?

- 1 bias and multiple weights
- Different sign of weights
- Different weights value
- What is the shape of the model? Still linear?



How machines formulate this equation?[Overview]

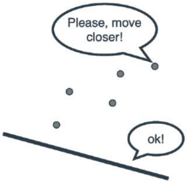
Overview

Inputs: A dataset of points.
Outputs: A linear regression model that fits that dataset.

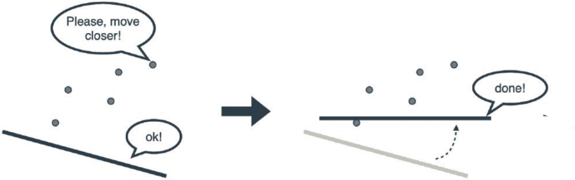
Procedure:

- Pick a model with random weights and a random bias.
- Repeat many times:
  1. Pick a random data point.
  2. Slightly adjust the weights (Slope) and bias (y-intercept) in order to improve the prediction for that particular data point.
- Return the model you've obtained.

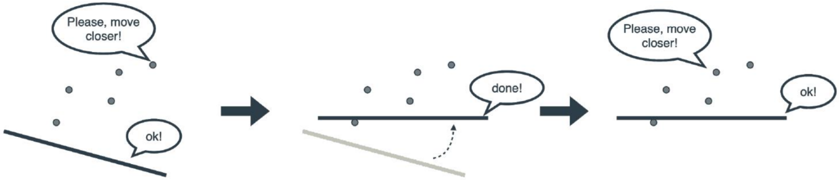
How machines formulate this equation?[Overview]



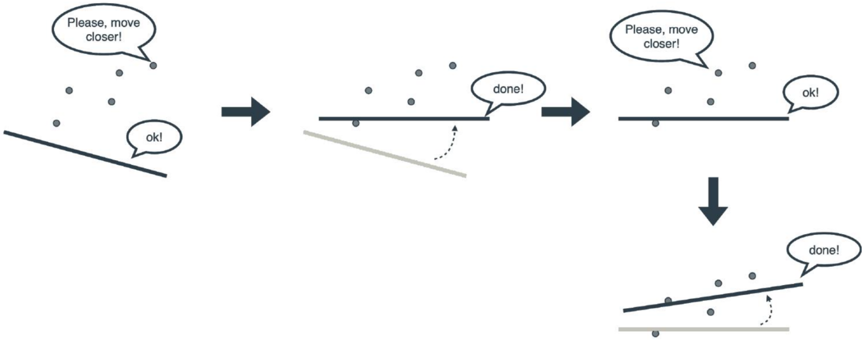
How machines formulate this equation?[Overview]



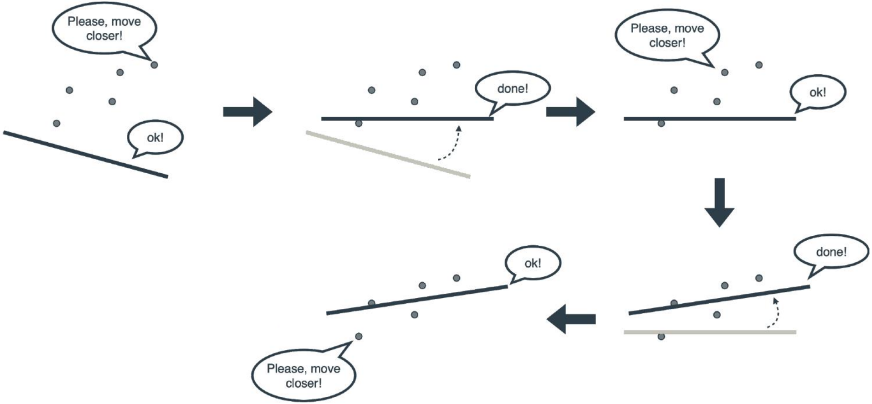
How machines formulate this equation?[Overview]



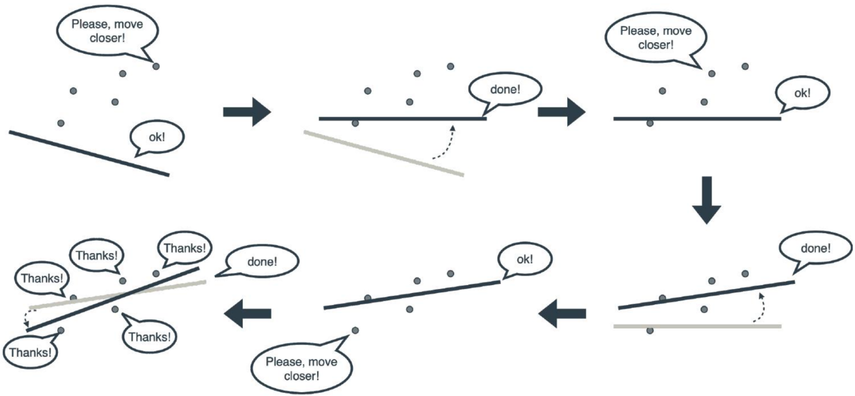
How machines formulate this equation?[Overview]



How machines formulate this equation?[Overview]



How machines formulate this equation?[Overview]



Linear Relationship

A linear relationship

- True, the line doesn't pass through every dot.
- However, the line does clearly show the relationship between rooms and price.

$$y' = mx + b$$

where:
y': is the price that value we're trying to predict.
m: is the slope of the line.
x: is the number of rooms value of our input feature.
b: is the y-intercept.

Linear Relationship in Machine Learning

In machine learning, we'll write the equation for a model slightly differently:

$$y' = w_1 x_1 + w_0$$

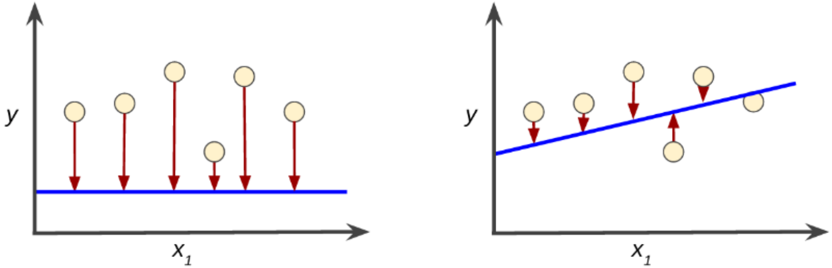
where:

- y': is the predicted label (a desired output).
- w_1: is the weight of feature 1. Weight is the same concept as the "slope".
- x_1: is feature 1.
- w_0 or b: is the bias (the y-intercept).

Training and Loss

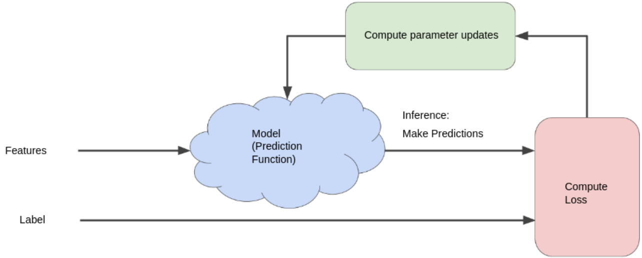
- **Training** a model simply means learning (determining) good values for all the weights and the bias from labeled examples.
- **Loss** is the penalty for a bad prediction.
  - Perfect prediction means the loss is zero
  - Bad model have large loss.
- Suppose we selected the following weights and biases.

Which of them have lower loss?



Reducing Loss

- **Training** is a feedback iterative process that use the loss function to improve the model parameters.



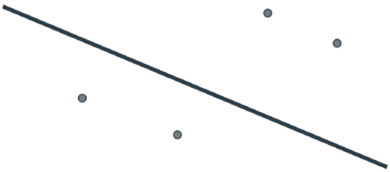
Some Questions

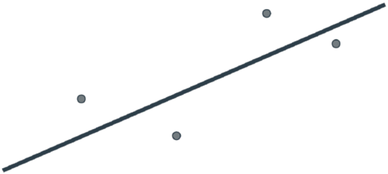
- How to define loss to measure the performance of the model?
- What initial values should we set for w_1 and w_0?
- How to update w_1 and w_0?

Outline

Loss Definition

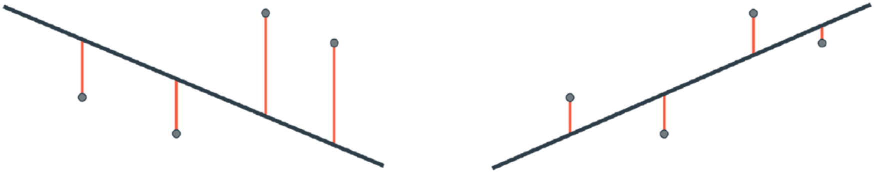
Which model is **better** and why? Which model have a **lower loss**?





Absolute Loss (L1 Loss)

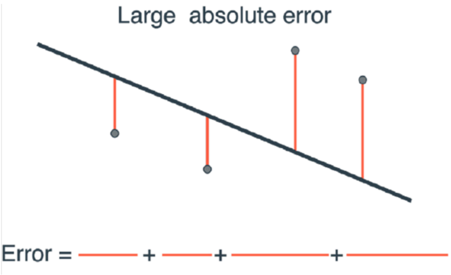
The absolute loss is the **sum** of the absolute differences between the observed and predicted values.



$$|obsevation(x) - prediction(x)|$$
$$= |(y - y')|$$

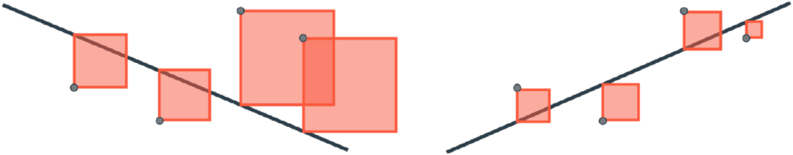
Absolute Loss (L1 Loss)

The absolute loss is the **sum** of the absolute differences between the observed and predicted values.



Squared Loss (L2 Loss)

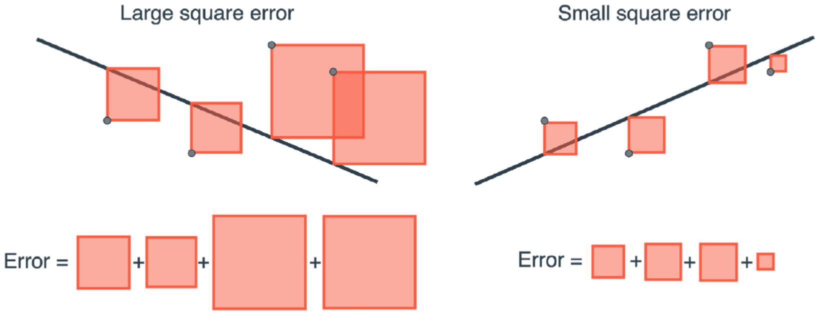
The squared loss is the **sum** of the squared differences between the observed and predicted values.



$$[obsevation(x) - prediction(x)]^2$$
$$= [(y - y')]^2$$

Squared Loss (L2 Loss)

The squared loss is the **sum** of the squared differences between the observed and predicted values..



Why Squared Loss?

Why Squared Loss?

- The squared loss is used in many machine learning algorithms because **it penalizes larger errors more than smaller ones**. This property makes it more sensitive to outliers compared to absolute loss.

- The squared loss also leads to **a nicer derivative** compared to the absolute loss, which simplifies optimization algorithms like gradient descent.

Mean square error (MSE)

- Is the average squared loss per example over the whole dataset.

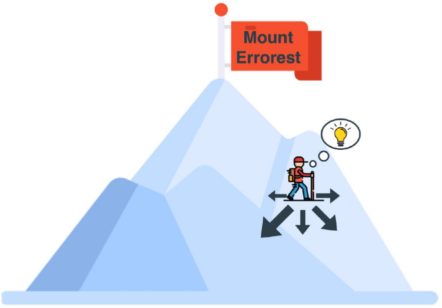
$$\text{MSE} = \frac{1}{N} \sum_{(x,y)\in D} (y - prediction(x))^2$$

- (x,y) is an example in which
  - y is the label
  - x is a feature
- prediction(x) is equal $y' = w_1 x + w_0$
- D is the dataset that contains all (x,y) pairs
- N is the number of samples in D

Reducing Loss

Gradient Descent

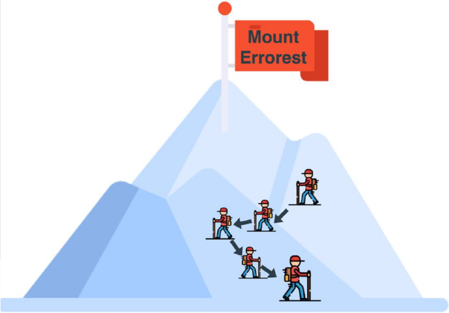
- The story starts with the error (cost) function.
- The machine's goal is to decrease the error.
- Here comes the gradient descent magic.



Reducing Loss

Gradient Descent

- The story starts with the error (cost) function.
- The machine's goal is to decrease the error.
- Here comes the gradient descent magic.



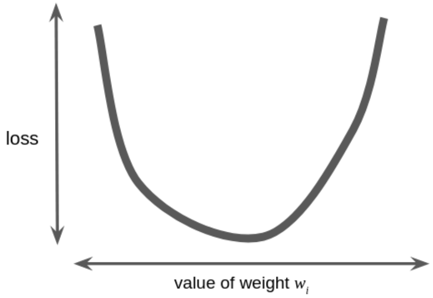
Gradient Descent (1/3)

- Assume (for symplicity) we are only concerned with finding w_1.
- Assume we had the time and the computing resources to calculate the loss for all possible values of w_1.

Gradient Descent (1/3)

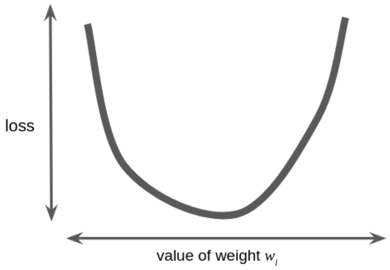
- Assume (for symplicity) we are only concerned with finding w_1.
- Assume we had the time and the computing resources to calculate the loss for all possible values of w_1.

Regression problems yield convex loss vs. weight plots.



Gradient Descent (2/3)

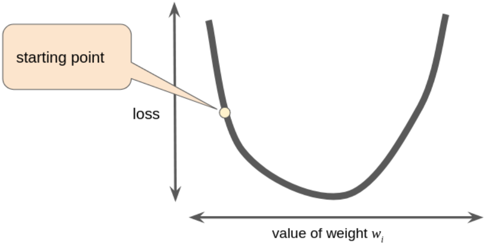
- Gradient descent enables you to find the optimal w without computing for all possible values.
- Gradient descent has the following steps
  1. Pick a random starting point for w
  2. Calculates the gradient of the loss curve at w.
  3. Update w
  4. go to 2, till convergence



Gradient Descent (3/3)

Note that a gradient is a vector, so it has both of the following characteristics:

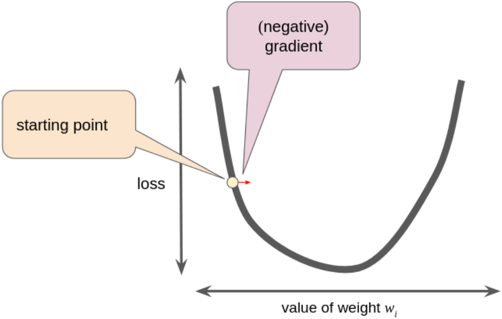
- Magnitude
- Direction



$$w_{new} = w_{old} - \eta * \frac{d \ \ loss}{dw}$$

Gradient Descent (3/3)

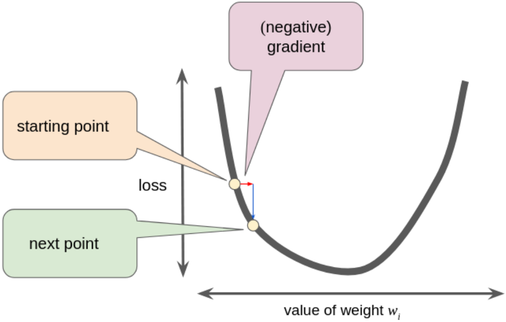
The gradient descent algorithm takes a step in the direction of the negative gradient



$$w_{new} = w_{old} - \eta * \frac{d \ \ loss}{dw}$$

Gradient Descent (3/3)

the gradient descent algorithm adds **some fraction** of the gradient's magnitude **(Learning Rate η)** to the previous point



$$w_{new} = w_{old} - \eta * \frac{d\ \ loss}{dw}$$

Outline

Gradient for Linear Regression

- For linear regression, the gradient of the loss function with respect to the weights w_1 and w_0 can be derived as follows:
- The loss function for linear regression is typically the l2 loss:

$$loss(w_0, w_1) = [y - y']^2 = [y - (w_1 x + w_0)]^2 \tag{1}$$

Gradient for Linear Regression

- For linear regression, the gradient of the loss function with respect to the weights w_1 and w_0 can be derived as follows:
- The loss function for linear regression is typically the l2 loss:

$$loss(w_0, w_1) = [y - y']^2 = [y - (w_1 x + w_0)]^2 \tag{1}$$

- The gradient of the loss function with respect to the weights w_1 and w_0 can be derived as follows:

$$\frac{d\ loss}{dw_1} = 2(y - y')(-x) = 2x(y' - y) \tag{2}$$

$$\frac{d\ loss}{dw_0} = 2(y - y')(-1) = 2(y' - y) \tag{3}$$

Full Gradient for Linear Regression

Procedure:

- Pick random weight w_1 and a random bias w_0.
- Repeat **many times**:
  1. Pick a random data point $(x^{(i)}, y^{(i)})$.
  2. Compute Model Prediction $y'^{(i)} = w_1 x_1^{(i)} + w_0$
  3. Update the weights and bias using the following equations:

$$w_1 = w_1 - \eta \frac{d\ loss}{dw_1} \tag{4}$$

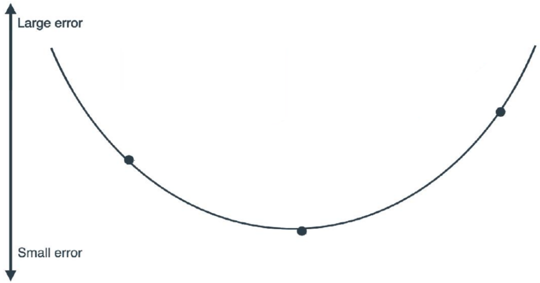
$$= w_1 - \eta 2 x_1^{(i)} (\underbrace{y'^{(i)} - y^{(i)}}_{\text{error}}) \tag{5}$$

$$w_0 = w_0 - \eta \frac{d\ loss}{dw_0} \tag{6}$$

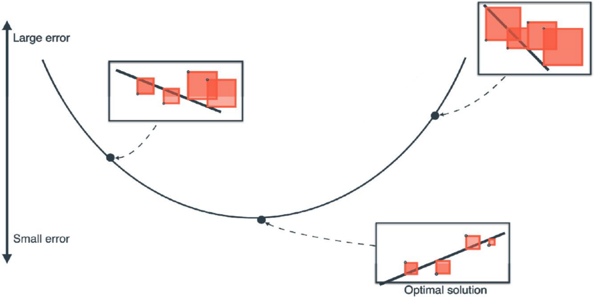
$$= w_0 - \eta 2 (\underbrace{y'^{(i)} - y^{(i)}}_{\text{error}}) \tag{7}$$

- Return the model you've obtained.

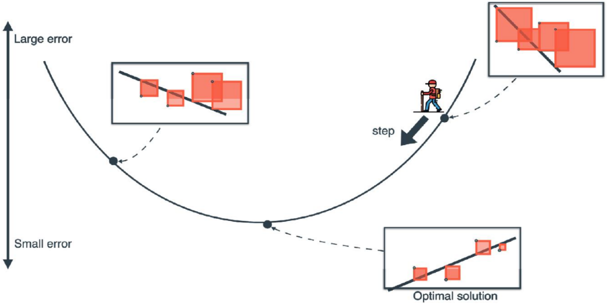
Gradient Descent for Linear Regression



Gradient Descent for Linear Regression



Gradient Descent for Linear Regression



Outline

Convergence Criteria

- For convex functions, optimum occurs when
  - $|\frac{d\ loss}{dw}| = 0$
- In practice, stop when
  - $|\frac{d\ loss}{dw}| \leq \epsilon$

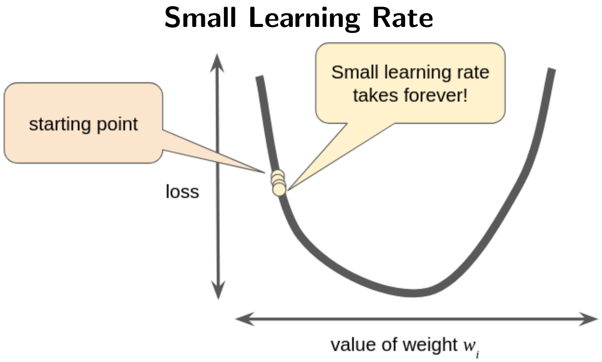
Outline

Learning rate

- Gradient descent algorithms multiply the gradient by a scalar known as the learning rate (also sometimes called step size) .
- How can we choose the learning rate?

Learning rate

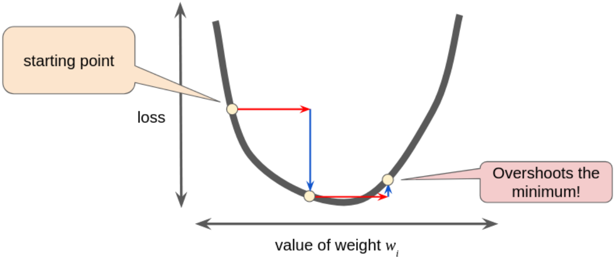
- Gradient descent algorithms multiply the gradient by a scalar known as the learning rate (also sometimes called step size) .
- How can we choose the learning rate?



Learning rate

- Gradient descent algorithms multiply the gradient by a scalar known as the learning rate (also sometimes called step size) .
- How can we choose the learning rate?

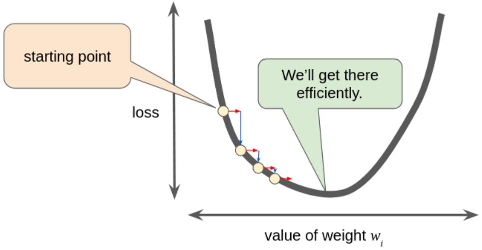
Large Learning Rate



Learning rate

- Gradient descent algorithms multiply the gradient by a scalar known as the learning rate (also sometimes called step size) .
- How can we choose the learning rate?

Optimal Learning Rate usually (0.01)



Outline

Types of Gradient Descents

- **Batch Gradient Descent:**
  - MSE loss assumes taking gradient for the total number of samples in the data set
  - Data sets often contain billions or even hundreds of billions of examples
  - Can take a very long time to compute.
- **Stochastic Gradient Descent (SGD):**
  - Uses only a single example (a batch size of 1) per iteration.
  - Very noisy.
- **Mini-Batch Gradient Descent:**
  - Compromise between full-batch iteration and SGD
  - Typically a batch of size between 10 and 1,000 examples, chosen at random.

Outline

Outline

Multivariate Linear Regression

- The general case of linear regression has more than one input feature.
- The model now becomes:

$$y' = w_1x_1 + w_2x_2 + \ldots + w_nx_n + w_0 \tag{8}$$

- We can rewrite this as:

$$y' = \sum_{i=0}^{i=n} w_ix_i \tag{9}$$

- Note w_0 is the bias (intercept), and $x_0 = 1$.

Generalization and Gradient

- For n features: $y' = \sum_{i=0}^{i=n} w_i x_i$
- Note w_0 is the bias (intercept), and $x_0 = 1$.
- vector representation y' $= \mathbf{w}^T \mathbf{x}$
- Loss $= \ell = (y - y')^2$
- Gradient derivation

$$\begin{aligned} \frac{d\ell}{dw_i} &= \frac{d\ell}{dy'} \frac{dy'}{dw_i} \\ &= [2(y' - y) * x_i] \end{aligned}$$

Multivariate Linear Regression

Size ($feet^2$)	Number of bedrooms	Number of floors	Age of home (years)	Price (\$1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

Multivariate Linear Regression

Size ($feet^2$) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

Multivariate Linear Regression

Size ($feet^2$) X_1	Number of bedrooms X_2	Number of floors X_3	Age of home (years) X_4	Price ($1000) Y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178

Multivariate Linear Regression

	Size (feet²) X_1	Number of bedrooms X_2	Number of floors X_3	Age of home (years) X_4	Price ($1000) Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178

Multivariate Linear Regression

feature

	Size (feet^2) X_1	Number of bedrooms X_2	Number of floors X_3	Age of home (years) X_4	Price ($1000) Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178

Multivariate Linear Regression

feature

data sample

	Size (feet^2) X_1	Number of bedrooms X_2	Number of floors X_3	Age of home (years) X_4	Price ($1000) Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178

Multivariate Linear Regression

	Size (feet^2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
feature	X_1	X_2	X_3	X_4	Y
data sample $X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178
Weights	W_1	W_2	W_3	W_4	

Multivariate Linear Regression

	Size ($feet^2$)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
	X_1	X_2	X_3	X_4	Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178
Weights	W_1	W_2	W_3	W_4	

feature

data sample

$X \quad * W \quad + b \quad = Y'$

Multivariate Linear Regression

feature

data sample

	Size (feet²) X_1	Number of bedrooms X_2	Number of floors X_3	Age of home (years) X_4	Price ($1000) Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178
Weights	W_1	W_2	W_3	W_4	

$$\begin{bmatrix} 2104 & 5 & 1 & 45 \end{bmatrix}$$

X *** W** **+ b** **= Y’**

Multivariate Linear Regression

feature

data sample

	Size ($feet^2$) X_1	Number of bedrooms X_2	Number of floors X_3	Age of home (years) X_4	Price ($1000) Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178
Weights	W_1	W_2	W_3	W_4	

$$X \begin{bmatrix} 2104 & 5 & 1 & 45 \end{bmatrix} * W \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} + b = Y'$$

Multivariate Linear Regression

feature

data sample

	Size (feet²)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
	X_1	X_2	X_3	X_4	Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178
Weights	W_1	W_2	W_3	W_4	

$$\mathbf{X} \begin{bmatrix} 2104 & 5 & 1 & 45 \end{bmatrix} * \mathbf{W} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} + b = Y' \begin{bmatrix} y'_1 \end{bmatrix}$$

Multivariate Linear Regression

	Size ($feet^2$)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
	X_1	X_2	X_3	X_4	Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178
Weights	W_1	W_2	W_3	W_4	

feature

data sample

$$\mathbf{X} \begin{bmatrix} 2104 & 5 & 1 & 45 \\ 1416 & 3 & 2 & 40 \\ 1534 & 3 & 2 & 30 \\ 852 & 2 & 1 & 36 \end{bmatrix} * \mathbf{W} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} + b = Y' \begin{bmatrix} y'_1 \end{bmatrix}$$

Multivariate Linear Regression

feature

data sample

	Size ($feet^2$) X_1	Number of bedrooms X_2	Number of floors X_3	Age of home (years) X_4	Price (\$1000) Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178
Weights	W_1	W_2	W_3	W_4	

$$X \begin{bmatrix} 2104 & 5 & 1 & 45 \\ 1416 & 3 & 2 & 40 \\ 1534 & 3 & 2 & 30 \\ 852 & 2 & 1 & 36 \end{bmatrix} * W \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} + b = Y' \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \end{bmatrix}$$

Multivariate Linear Regression

feature

data sample

	Size (feet²) X_1	Number of bedrooms X_2	Number of floors X_3	Age of home (years) X_4	Price ($1000) Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178
Weights	W_1	W_2	W_3	W_4	

$$\mathbf{X} \underbrace{\begin{bmatrix} 2104 & 5 & 1 & 45 \\ 1416 & 3 & 2 & 40 \\ 1534 & 3 & 2 & 30 \\ 852 & 2 & 1 & 36 \end{bmatrix}}_{[\#samples * \#features]} * \mathbf{W} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix} + b = Y' \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \end{bmatrix}$$

Multivariate Linear Regression

feature

data sample

	Size (feet²) X_1	Number of bedrooms X_2	Number of floors X_3	Age of home (years) X_4	Price ($1000) Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178
Weights	W_1	W_2	W_3	W_4	

$$\mathbf{X}\ \underbrace{\begin{bmatrix} 2104 & 5 & 1 & 45 \\ 1416 & 3 & 2 & 40 \\ 1534 & 3 & 2 & 30 \\ 852 & 2 & 1 & 36 \end{bmatrix}}_{[\#samples*\#features]} * \mathbf{W}\ \underbrace{\begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix}}_{[\#features*1]} + \mathbf{b} = \mathbf{Y'}\ \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \end{bmatrix}$$

Multivariate Linear Regression

feature

data sample

	Size (feet^2) X_1	Number of bedrooms X_2	Number of floors X_3	Age of home (years) X_4	Price ($1000) Y
$X^{(1)}$	2104	5	1	45	460
$X^{(2)}$	1416	3	2	40	232
$X^{(3)}$	1534	3	2	30	315
$X^{(4)}$	852	2	1	36	178
Weights	W_1	W_2	W_3	W_4	

$$\mathbf{X} \underbrace{\begin{bmatrix} 2104 & 5 & 1 & 45 \\ 1416 & 3 & 2 & 40 \\ 1534 & 3 & 2 & 30 \\ 852 & 2 & 1 & 36 \end{bmatrix}}_{[\#samples*\#features]} * \mathbf{W} \underbrace{\begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \end{bmatrix}}_{[\#features*1]} + b = Y' \underbrace{\begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \end{bmatrix}}_{[\#samples*1]}$$

Outline

Normal Equation

$$\mathbf{X} \begin{bmatrix} 2104 & 5 & 1 & 45 \\ 1416 & 3 & 2 & 40 \\ 1534 & 3 & 2 & 30 \\ 852 & 2 & 1 & 36 \end{bmatrix} * \mathbf{W} \begin{bmatrix} \mathbf{W_1} \\ \mathbf{W_2} \\ \mathbf{W_3} \\ \mathbf{W_4} \end{bmatrix} + \mathbf{b} \quad = \mathbf{Y'} \begin{bmatrix} \mathbf{y'_1} \\ \mathbf{y'_2} \\ \mathbf{y'_3} \\ \mathbf{y'_4} \end{bmatrix}$$

[#samples* #features] [#features*1] [#samples*1]

$\downarrow$

Normal Equation

$$\mathbf{X}\ \begin{bmatrix} 2104 & 5 & 1 & 45 \\ 1416 & 3 & 2 & 40 \\ 1534 & 3 & 2 & 30 \\ 852 & 2 & 1 & 36 \end{bmatrix} * \mathbf{W}\ \begin{bmatrix} \mathbf{W_1} \\ \mathbf{W_2} \\ \mathbf{W_3} \\ \mathbf{W_4} \end{bmatrix} + \mathbf{b} = \mathbf{Y'}\ \begin{bmatrix} \mathbf{y'_1} \\ \mathbf{y'_2} \\ \mathbf{y'_3} \\ \mathbf{y'_4} \end{bmatrix}$$

[#samples* #features] [#features*1] [#samples*1]

$$\mathbf{X}\ \begin{bmatrix} 2104 & 5 & 1 & 45 \\ 1416 & 3 & 2 & 40 \\ 1534 & 3 & 2 & 30 \\ 852 & 2 & 1 & 36 \end{bmatrix} * \mathbf{W}\ \begin{bmatrix} \mathbf{W_1} \\ \mathbf{W_2} \\ \mathbf{W_3} \\ \mathbf{W_4} \end{bmatrix} + \mathbf{b} = \mathbf{Y'}\ \begin{bmatrix} \mathbf{y'_1} \\ \mathbf{y'_2} \\ \mathbf{y'_3} \\ \mathbf{y'_4} \end{bmatrix}$$

Normal Equation

$$
\underbrace{\mathbf{X} \begin{bmatrix} 2104 & 5 & 1 & 45 \\ 1416 & 3 & 2 & 40 \\ 1534 & 3 & 2 & 30 \\ 852 & 2 & 1 & 36 \end{bmatrix}}_{[\#samples * \#features]} * \underbrace{\mathbf{W} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ \mathbf{w}_4 \end{bmatrix}}_{[\#features * 1]} + \mathbf{b} \quad = \underbrace{\mathbf{Y'} \begin{bmatrix} \mathbf{y'}_1 \\ \mathbf{y'}_2 \\ \mathbf{y'}_3 \\ \mathbf{y'}_4 \end{bmatrix}}_{[\#samples * 1]}
$$

$$\downarrow$$

$$
\mathbf{X} \begin{bmatrix} & 2104 & 5 & 1 & 45 \\ & 1416 & 3 & 2 & 40 \\ & 1534 & 3 & 2 & 30 \\ & 852 & 2 & 1 & 36 \end{bmatrix} * \mathbf{W} \begin{bmatrix} \\ \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ \mathbf{w}_4 \end{bmatrix} \quad = \mathbf{Y'} \begin{bmatrix} \mathbf{y'}_1 \\ \mathbf{y'}_2 \\ \mathbf{y'}_3 \\ \mathbf{y'}_4 \end{bmatrix}
$$

Normal Equation

$$\mathbf{X}\begin{bmatrix} 2104 & 5 & 1 & 45 \\ 1416 & 3 & 2 & 40 \\ 1534 & 3 & 2 & 30 \\ 852 & 2 & 1 & 36 \end{bmatrix} * \mathbf{W}\begin{bmatrix} \mathbf{W_1} \\ \mathbf{W_2} \\ \mathbf{W_3} \\ \mathbf{W_4} \end{bmatrix} + \mathbf{b} \quad = \mathbf{Y'}\begin{bmatrix} \mathbf{y'_1} \\ \mathbf{y'_2} \\ \mathbf{y'_3} \\ \mathbf{y'_4} \end{bmatrix}$$

[#samples* #features] [#features*1] [#samples*1]

$$\mathbf{X}\begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} * \mathbf{W}\begin{bmatrix} \mathbf{b} \\ \mathbf{W_1} \\ \mathbf{W_2} \\ \mathbf{W_3} \\ \mathbf{W_4} \end{bmatrix} \quad = \mathbf{Y'}\begin{bmatrix} \mathbf{y'_1} \\ \mathbf{y'_2} \\ \mathbf{y'_3} \\ \mathbf{y'_4} \end{bmatrix}$$

Normal Equation

$$\mathbf{X} \begin{bmatrix} 2104 & 5 & 1 & 45 \\ 1416 & 3 & 2 & 40 \\ 1534 & 3 & 2 & 30 \\ 852 & 2 & 1 & 36 \end{bmatrix} * \mathbf{W} \begin{bmatrix} \mathbf{W_1} \\ \mathbf{W_2} \\ \mathbf{W_3} \\ \mathbf{W_4} \end{bmatrix} + \mathbf{b} = \mathbf{Y'} \begin{bmatrix} \mathbf{y'_1} \\ \mathbf{y'_2} \\ \mathbf{y'_3} \\ \mathbf{y'_4} \end{bmatrix}$$

[#samples* #features]   [#features*1]   [#samples*1]

$$\downarrow$$

$$\mathbf{X} \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} * \mathbf{W} \begin{bmatrix} \mathbf{b} \\ \mathbf{W_1} \\ \mathbf{W_2} \\ \mathbf{W_3} \\ \mathbf{W_4} \end{bmatrix} = \mathbf{Y'} \begin{bmatrix} \mathbf{y'_1} \\ \mathbf{y'_2} \\ \mathbf{y'_3} \\ \mathbf{y'_4} \end{bmatrix}$$

[#samples*(#features+1)]   [(#features+1)*1]   [#samples*1]

Normal Equation

- Normal equation is a closed-form solution to the linear regression problem.

Normal Equation

- Normal equation is a closed-form solution to the linear regression problem.
- It provides an exact solution to the model parameters w that minimize the squared error between the predicted and actual values.

Normal Equation

- Normal equation is a closed-form solution to the linear regression problem.
- It provides an exact solution to the model parameters w that minimize the squared error between the predicted and actual values.
- Since we have:

$$\mathbf{Y} = \mathbf{X}.\mathbf{W} \tag{10}$$

Normal Equation

- Normal equation is a closed-form solution to the linear regression problem.
- It provides an exact solution to the model parameters w that minimize the squared error between the predicted and actual values.
- Since we have:

$$\mathbf{Y} = \mathbf{X}.\mathbf{W} \tag{10}$$

- Multiply both sides by the inverse of $\mathbf{X}$:

$$\mathbf{X}^{-1}.\mathbf{Y} = \mathbf{W} \tag{11}$$

Normal Equation

- Normal equation is a closed-form solution to the linear regression problem.
- It provides an exact solution to the model parameters w that minimize the squared error between the predicted and actual values.
- Since we have:

$$\mathbf{Y} = \mathbf{X}.\mathbf{W} \tag{10}$$

- Multiply both sides by the inverse of $\mathbf{X}$:

$$\mathbf{X}^{-1}.\mathbf{Y} = \mathbf{W} \tag{11}$$

- Since $\mathbf{X}$ is not a square matrix, we can't find the inverse of $\mathbf{X}$.

Normal Equation

- Normal equation is a closed-form solution to the linear regression problem.
- It provides an exact solution to the model parameters w that minimize the squared error between the predicted and actual values.
- Since we have:

$$\mathbf{Y} = \mathbf{X}.\mathbf{W} \tag{10}$$

- Multiply both sides by the inverse of $\mathbf{X}$:

$$\mathbf{X}^{-1}.\mathbf{Y} = \mathbf{W} \tag{11}$$

- Since $\mathbf{X}$ is not a square matrix, we can't find the inverse of $\mathbf{X}$.
- We can use the following trick.

$$\mathbf{X}^T\mathbf{Y} = \mathbf{X}^T\mathbf{X}\mathbf{W} \tag{12}$$

Normal Equation

- Normal equation is a closed-form solution to the linear regression problem.
- It provides an exact solution to the model parameters w that minimize the squared error between the predicted and actual values.
- Since we have:

$$\mathbf{Y} = \mathbf{X}.\mathbf{W} \tag{10}$$

- Multiply both sides by the inverse of $\mathbf{X}$:

$$\mathbf{X}^{-1}.\mathbf{Y} = \mathbf{W} \tag{11}$$

- Since $\mathbf{X}$ is not a square matrix, we can't find the inverse of $\mathbf{X}$.
- We can use the following trick.

$$\mathbf{X}^T\mathbf{Y} = \mathbf{X}^T\mathbf{X}\mathbf{W} \tag{12}$$

- Now, multiply both sides by the inverse of $\mathbf{X}^T\mathbf{X}$:

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \mathbf{W} \tag{13}$$

Exercise 1

- What is the purpose of the normal equation in linear regression?
- What are the advantages of using gradient descent over normal equations for model optimization?
- How does the learning rate affect the convergence of gradient descent?
- What is the role of the loss function in training a machine learning model?
- Compare and contrast batch gradient descent, stochastic gradient descent, and mini-batch gradient descent.

Exercise 2

- Consider the following dataset with four points: (1, 2), (2, 4), (3, 5), (4, 4)
- Initial values: $w_0 = 1$, $w_1 = 0.5$, learning rate $\eta = 0.1$
- Use stochastic gradient descent to update w_0 and w_1 for one epoch (4 iterations)
- Complete the table below, showing your calculations for each iteration

Iteration	(x, y)	y'	$y' - y$	$(y' - y)^2$	$\frac{\partial L}{\partial w_0}$	$\frac{\partial L}{\partial w_1}$	New w_0	New w_1
1	(1, 2)							
2	(2, 4)							
3	(3, 5)							
4	(4, 4)							

- Remember: $y' = w_1 x + w_0$
- Use the formulas: $\frac{\partial L}{\partial w_0} = 2(y' - y)$ and $\frac{\partial L}{\partial w_1} = 2x(y' - y)$
- Update rule: $w_{new} = w_{old} - \eta \frac{\partial L}{\partial w}$

Questions