

MACHINE LEARNING



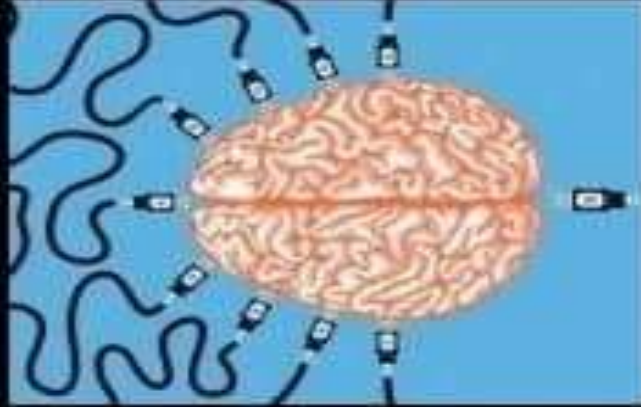
INTRODUCTION TO MACHINE LEARNING

WEEK 1

Machine Learning



What society thinks I do.



What my friends think I do.



What computer scientists think I do.



What my boss thinks I do.

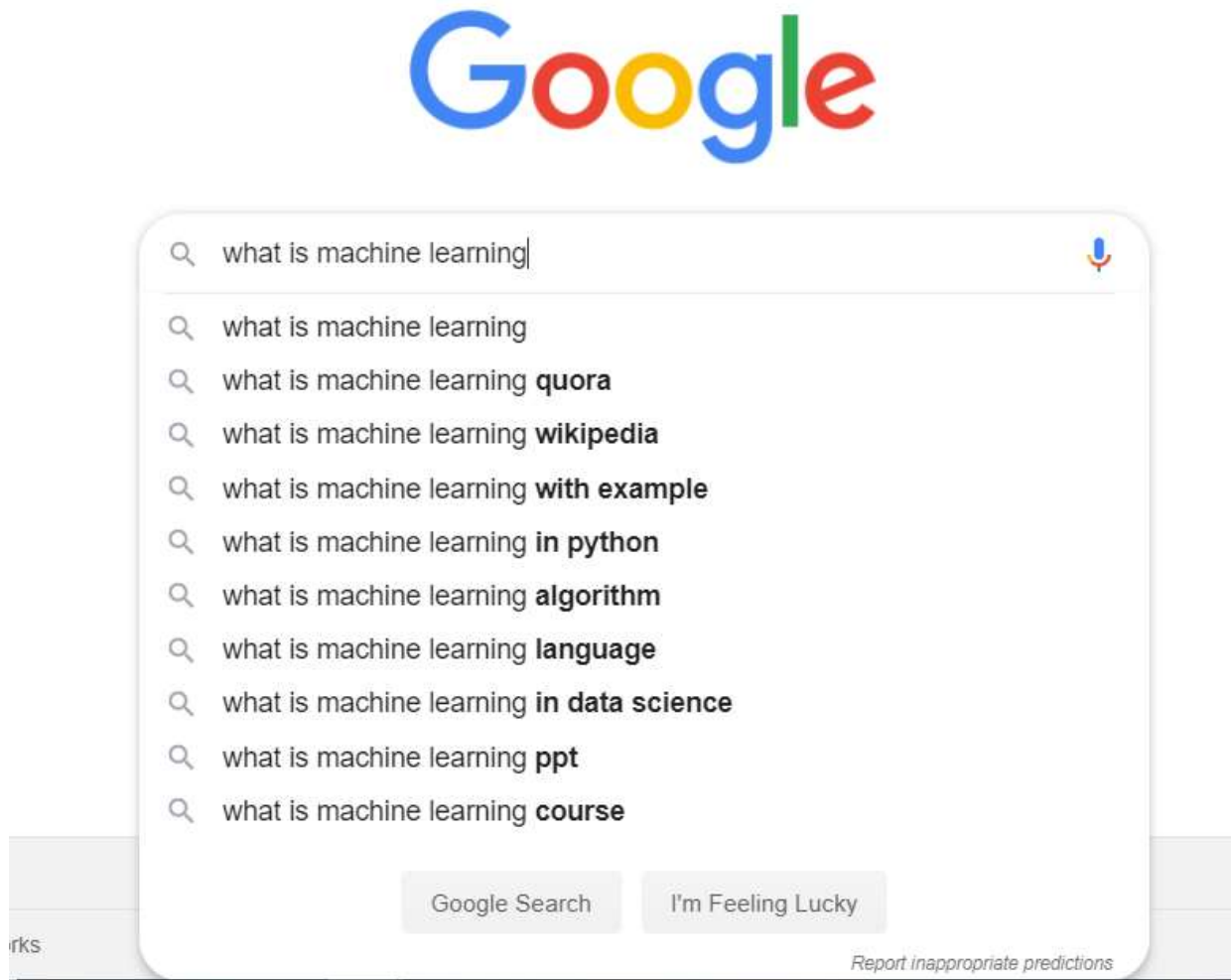


What I think I do.



What I really do.

Ever wondered how do these work?



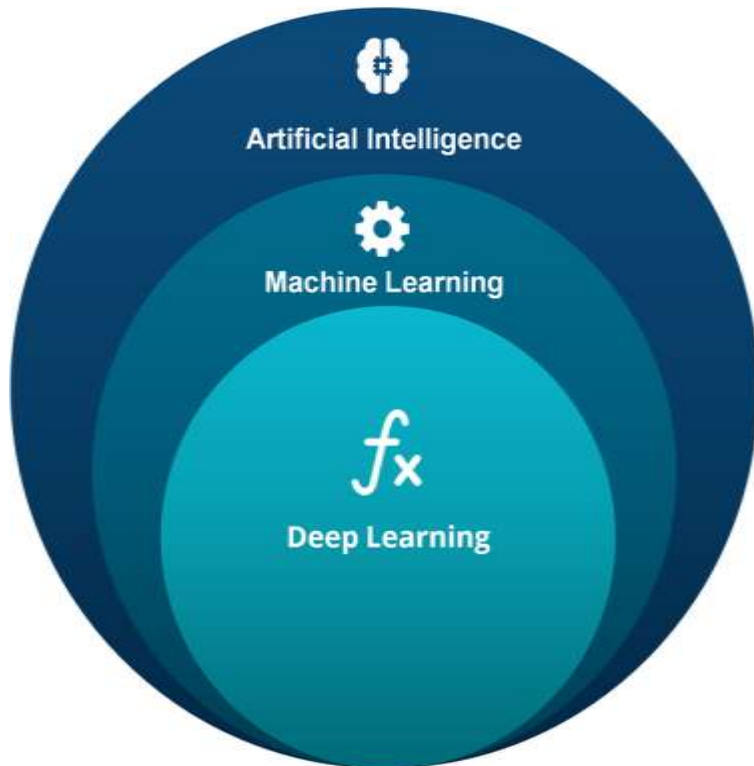
Google Search Engine



Facebook Photo Tagging

Why ML?

- It was not possible to explicitly program different applications.
- There was a need to build intelligent machine systems to write AI programs themselves to do more interesting things like finding a shortest path from A to B or web search, photo tagging or even spam filter.

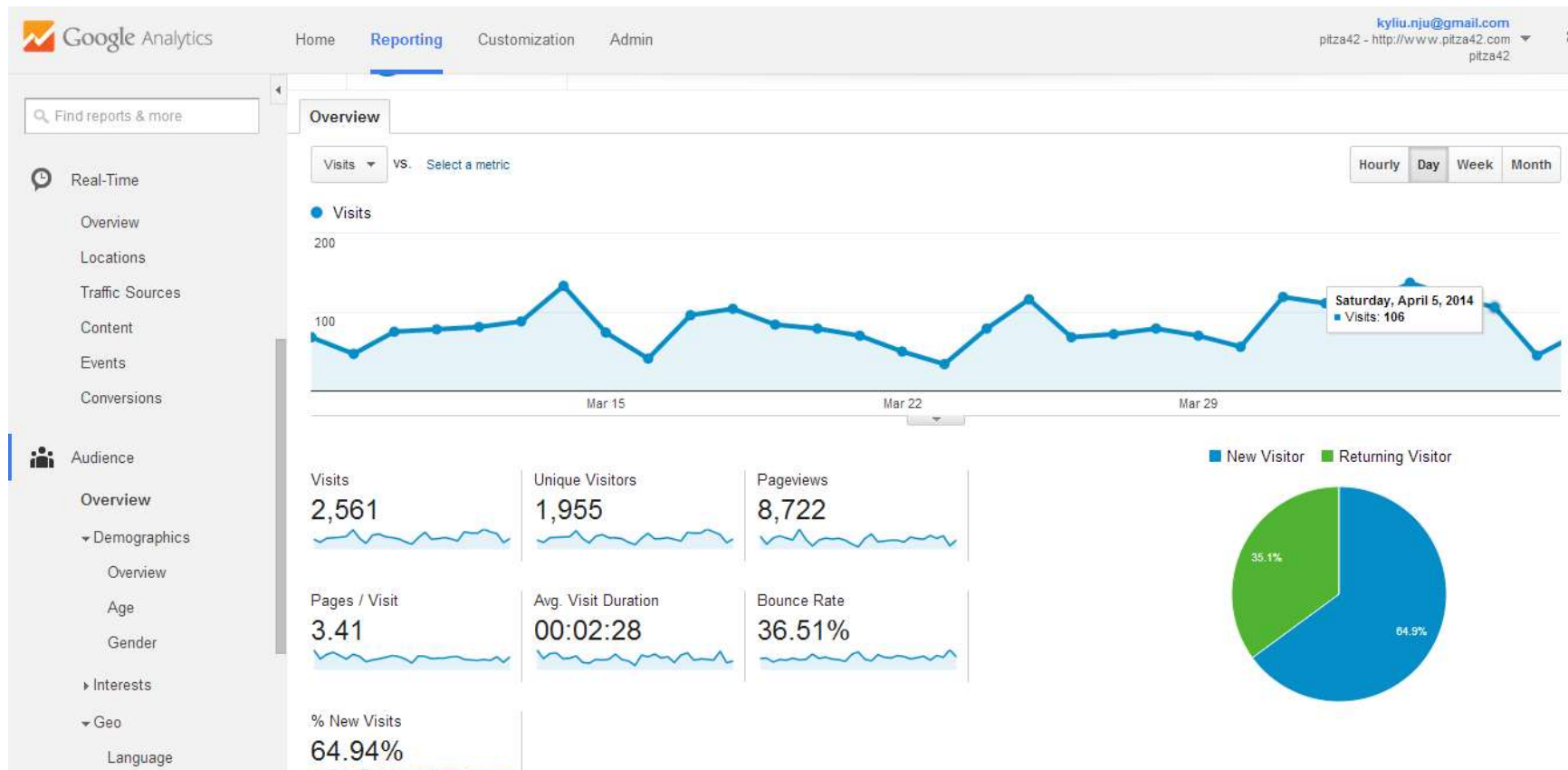


- **Machine Learning grew out of AI to solve these problems**
- **ML is a new capability for computers which solves these problems using certain learning algorithms**

Examples: Database Mining

Web Click Data:

Large dataset from growth of automation/web like **clickstream data** to understand and serve the users better.



Medical Record:

Electronic medical records that can be transformed into medical knowledge to understand diseases better.



MEDICAL RECORD

ADMINISTRATIVE & SELF-REPORT INFORMATION (May Be Completed by Patient or Health Care Provider)

Patient: _____ Date: _____

Address: _____

City: _____ Home: (____) _____

Telephone: Work (____) _____

Health Plan or other Patient ID#: _____

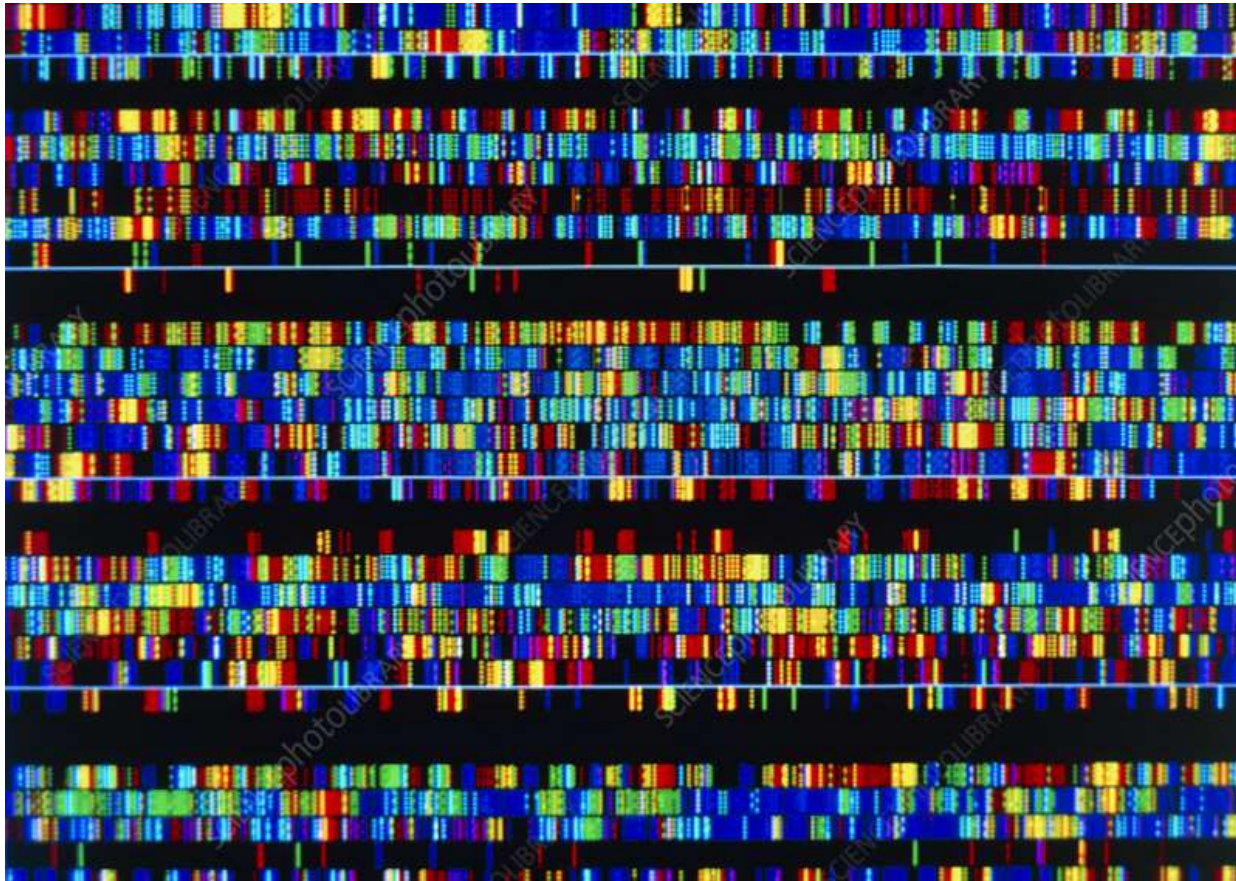
Employer/School: _____

☐ Male ☐ Female Marital Status: ☐ Married

Contact: _____

Computational Biology:

Collecting lots of data about gene sequences, DNA sequences, etc. which give much better understanding of human genome and what it means to human.



Applications that can't be programmed manually



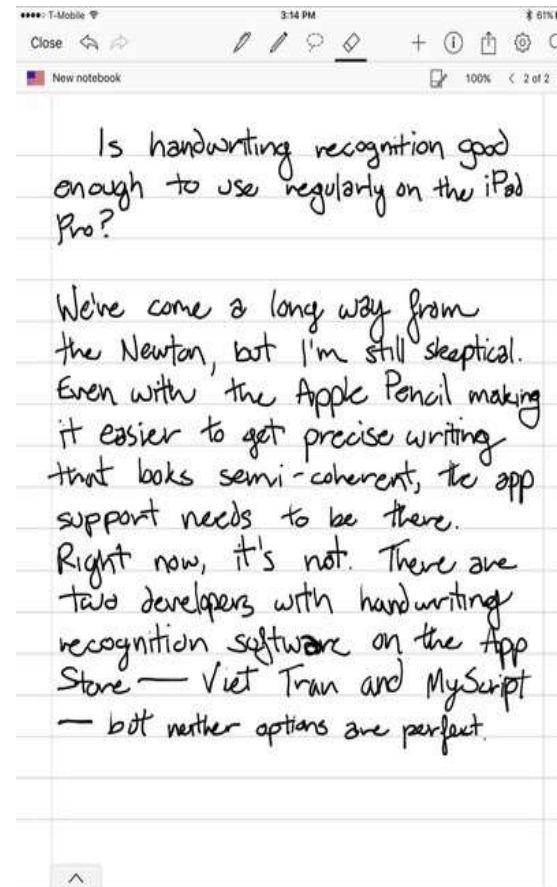
Autonomous Helicopter, Stanford University



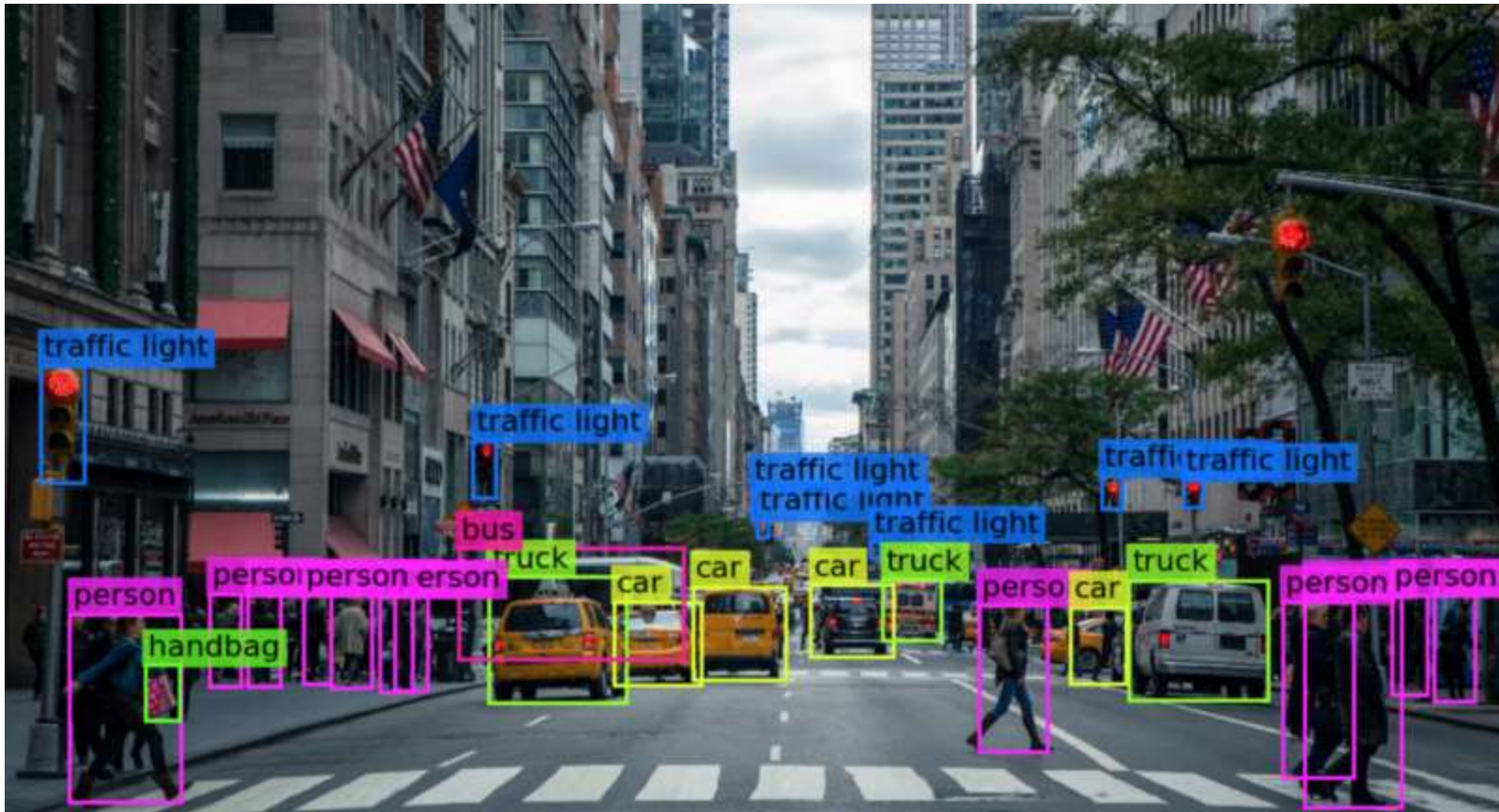
Tesla's self-driving car



Natural Language Processing



Handwriting Recognition



Computer Vision

Self-customizing programs

Product/video recommendations on Amazon, Netflix, YouTube, etc.



Understanding and mimic human behaviour - AI

Artificial Intelligence – The most advanced use of Machine Learning



Sophia – The Humanoid Robot

Definition

Arthur Samuel (1959) Machine Learning:

“Field of study that gives computers the ability to learn without being explicitly programmed.”

Tom Mitchell (1998) Well-posed Learning Problem:

“A computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .”

Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam. What is the task T in this setting?

- Classify emails as spam or not spam.
- Watching you label emails as spam or not spam.
- The number (or fraction) of emails correctly classified as spam/not spam.
- None of the above. This is not an ML algorithm.

ML Algorithms

Machine Learning Algorithms:

- Supervised Learning
- Unsupervised Learning

Others:

- Reinforcement Learning
- Recommender Systems

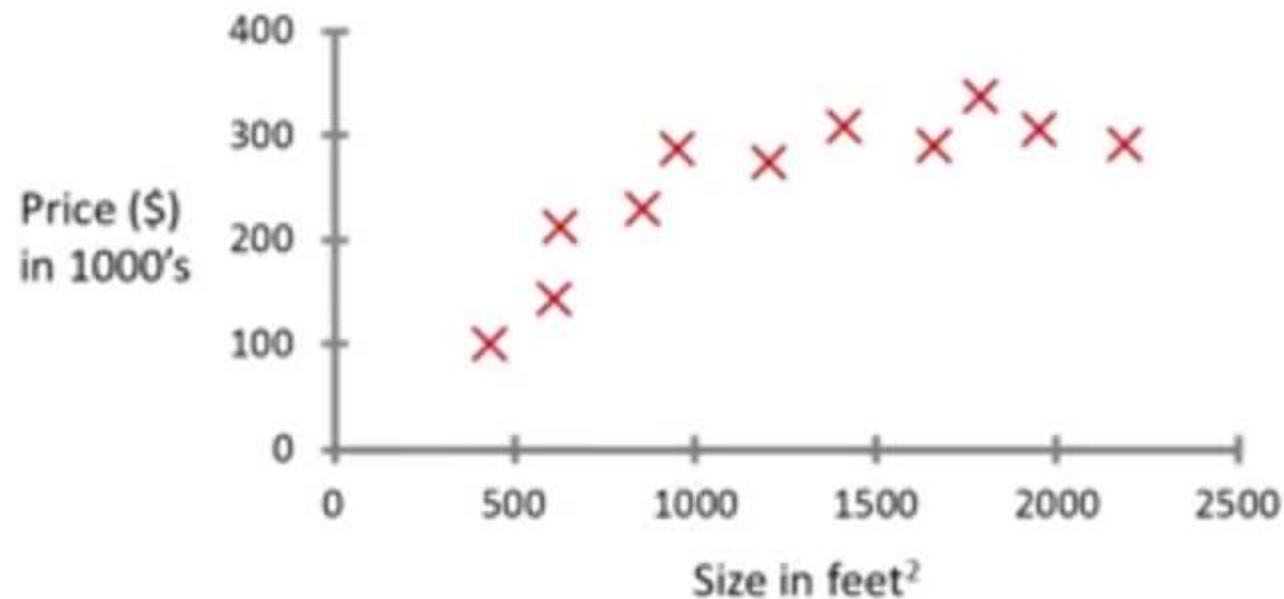
Supervised Learning

- In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.
- Supervised learning problems are categorized into "**regression**" and "**classification**" problems
- In a regression problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function.
- In a classification problem, we are instead trying to predict results in a discrete output. In other words, we are trying to map input variables into discrete categories.

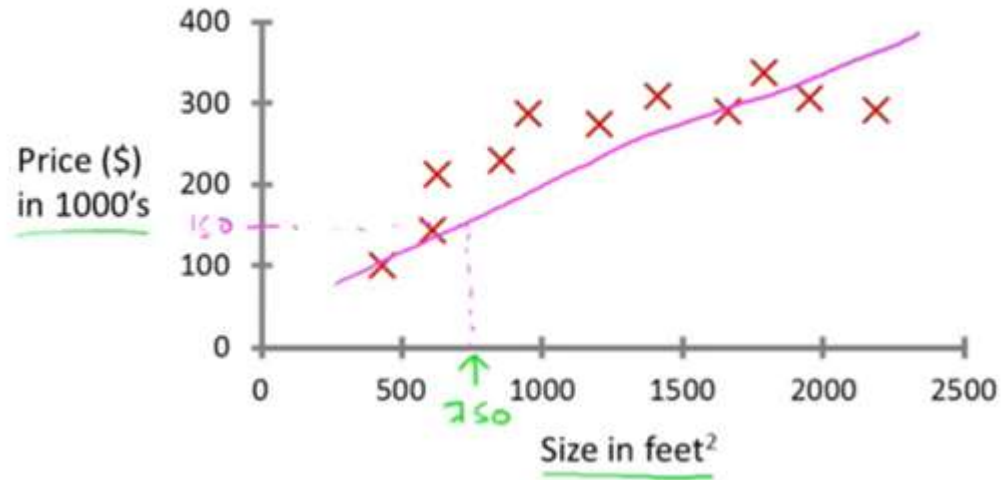
Regression

Given data about the size of houses on the real estate market, try to predict their price.

House Price Prediction



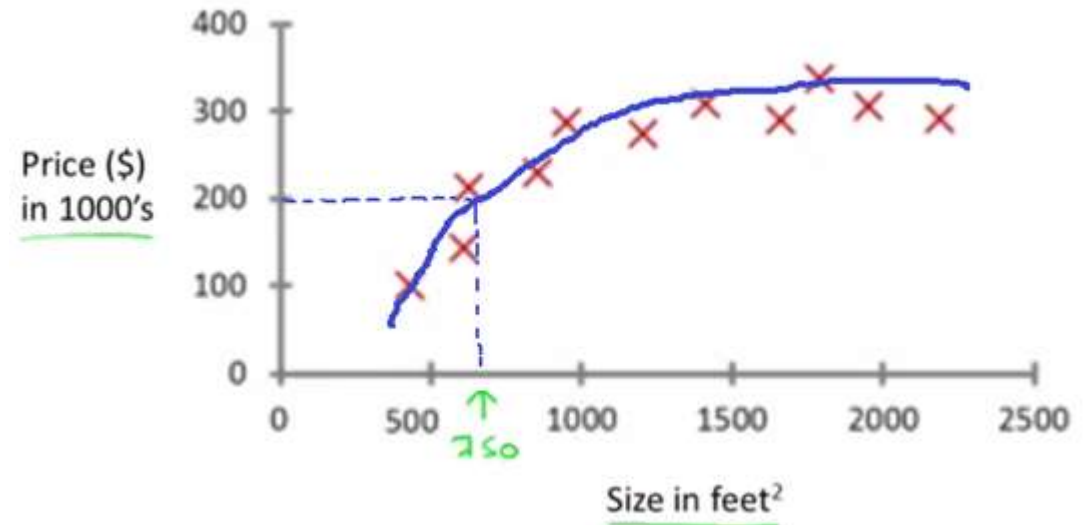
Case I:



Size : 750 sq. feet

Price : \$150k

Case II:



Size : 750 sq. feet

Price : \$200k

Supervised Learning gives **“right answers”**

In this example, the algorithm predicts value of house for every data set given and the task of the algorithm is to produce more of these right answers.

This problem is also called **“Regression Problem”**

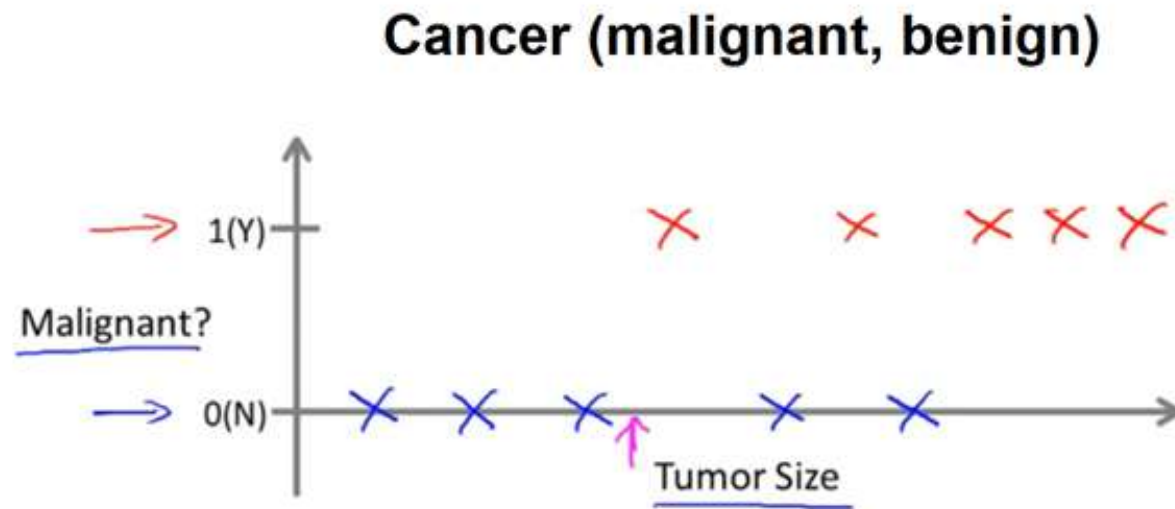
Regression: Predict continuous valued output (price)

Classification

Consider an example:

Malignant (harmful), Benign (not harmful)

What is the probability that cancer is malignant or benign?



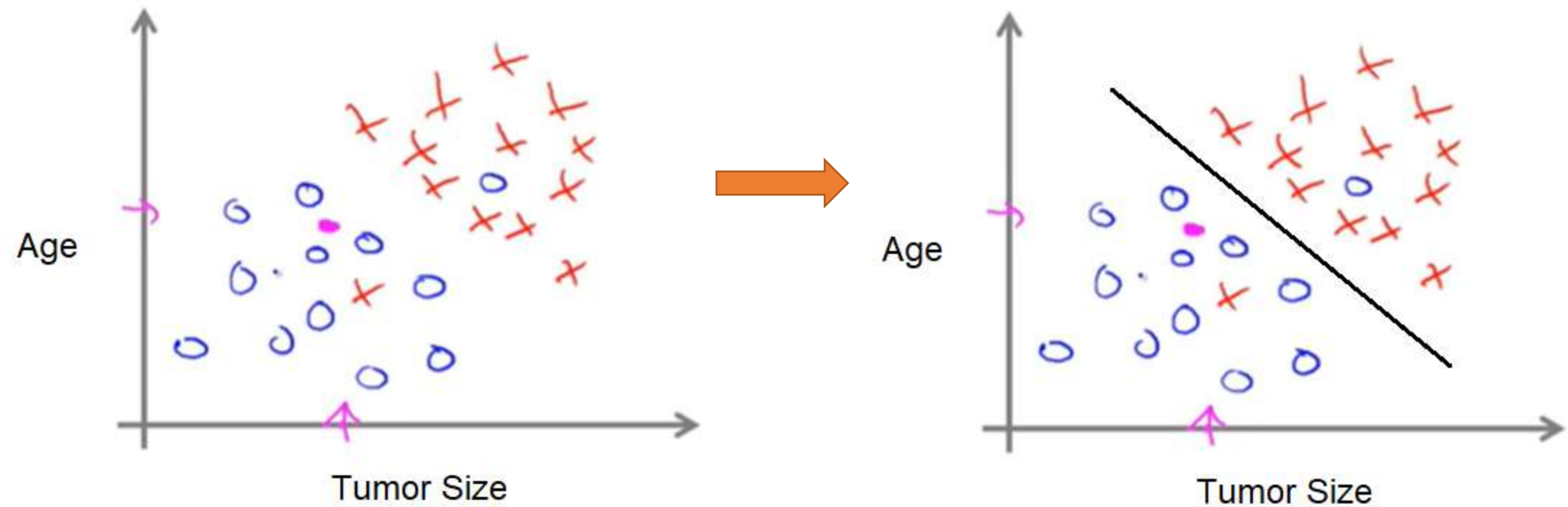
This problem is also called “**Classification Problem**”

Classification: Discrete valued output (0 or 1)

In this example, there is only one feature (tumor size) to predict whether the tumor is malignant or benign

A Stanford University case study:

In this example, there are two features (age, tumor size) to determine whether cancer is **malignant** or **benign**.



The algorithm fits a straight line to the data to try to separate out the malignant tumors from the benign ones.

* Other features used were clump thickness, uniformity of cell size, shape, etc.

You're running a company, and you want to develop learning algorithms to address each of two problems.

Problem 1:

You have a large inventory of identical items. You want to predict how many of these items will sell over the next 3 months.

Problem 2: You'd like software to examine individual customer accounts, and for each account decide if it has been hacked/compromised. Should you treat these as classification or as regression problems?

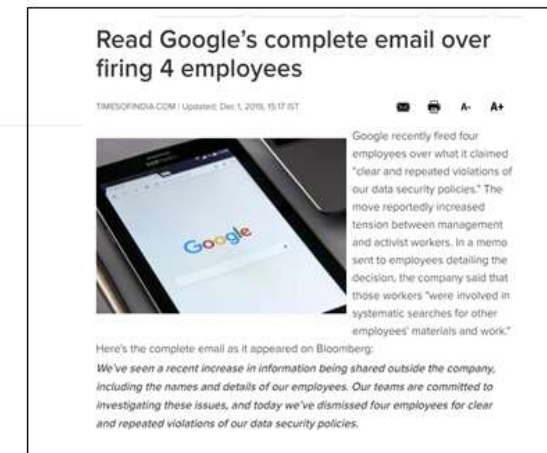
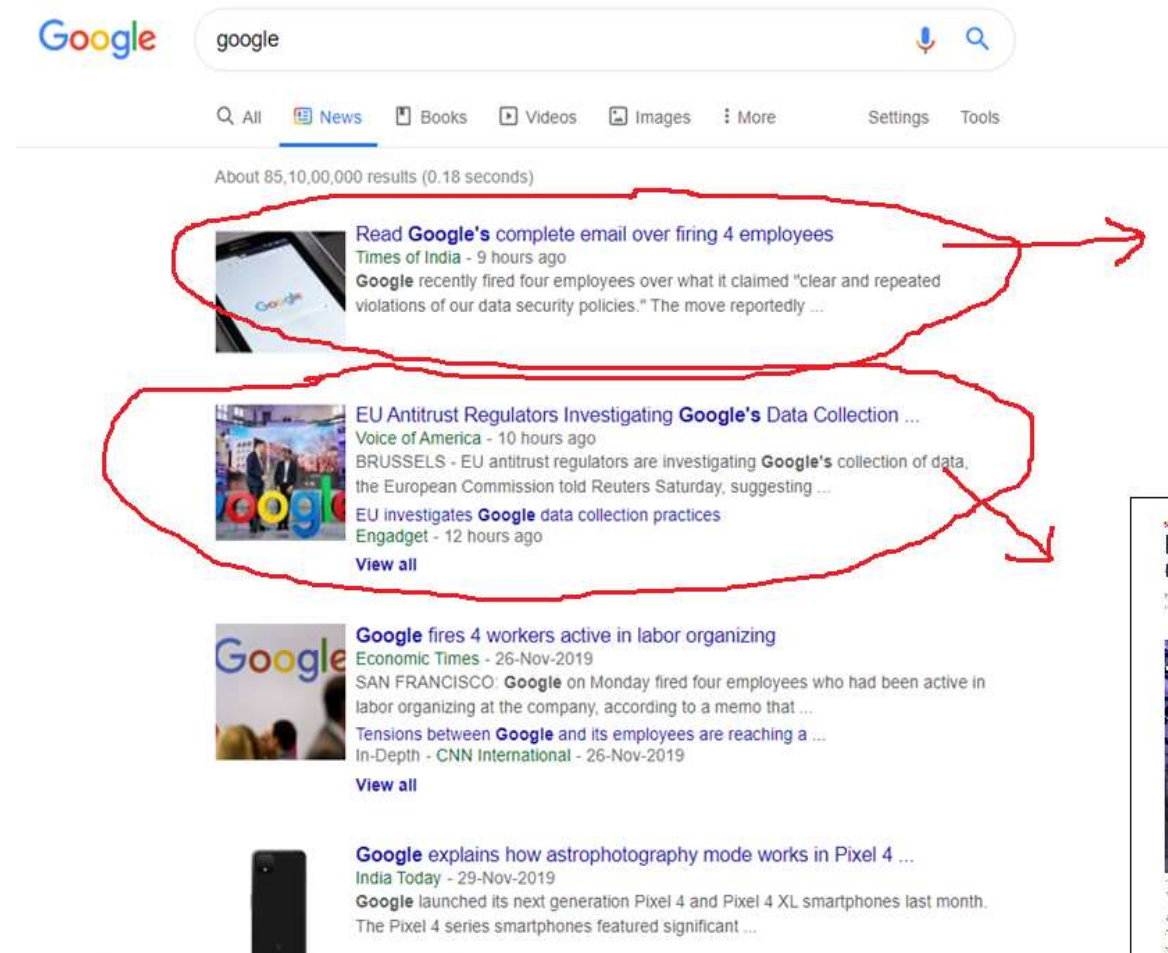
- Treat both as classification problems
- Treat problem 1 as a classification problem, problem 2 as a regression problem
- Treat problem 1 as a regression problem, problem 2 as a classification problem
- Treat both as regression problems

Unsupervised Learning

- Unsupervised learning allows us to approach problems with little or no idea what our results should look like. We can derive structure from data where we don't necessarily know the effect of the variables.
- We can derive this structure by clustering the data based on relationships among the variables in the data.
- With unsupervised learning there is no feedback based on the prediction results.

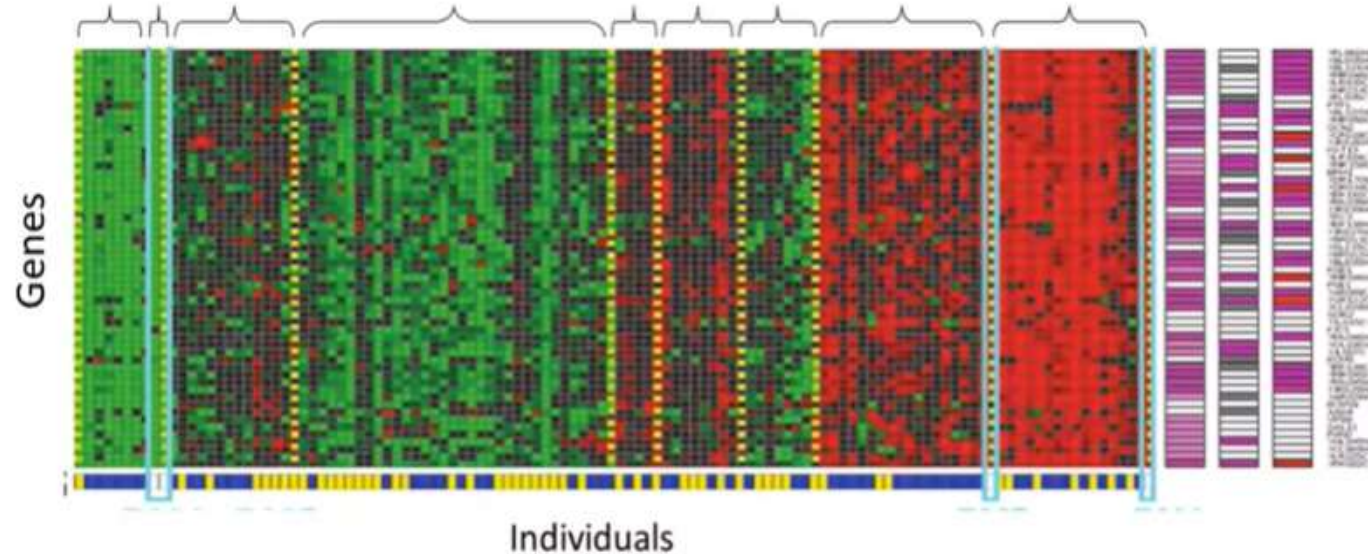
Example: Google News

Google News looks for thousands of news stories and automatically cluster them together. So, news about the same topic get displayed together.



Example: DNA Microarray Data (Clustering)

Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables, such as lifespan, location, roles, and so on.



Run a clustering algorithm to group individuals into different categories or into different types of people.

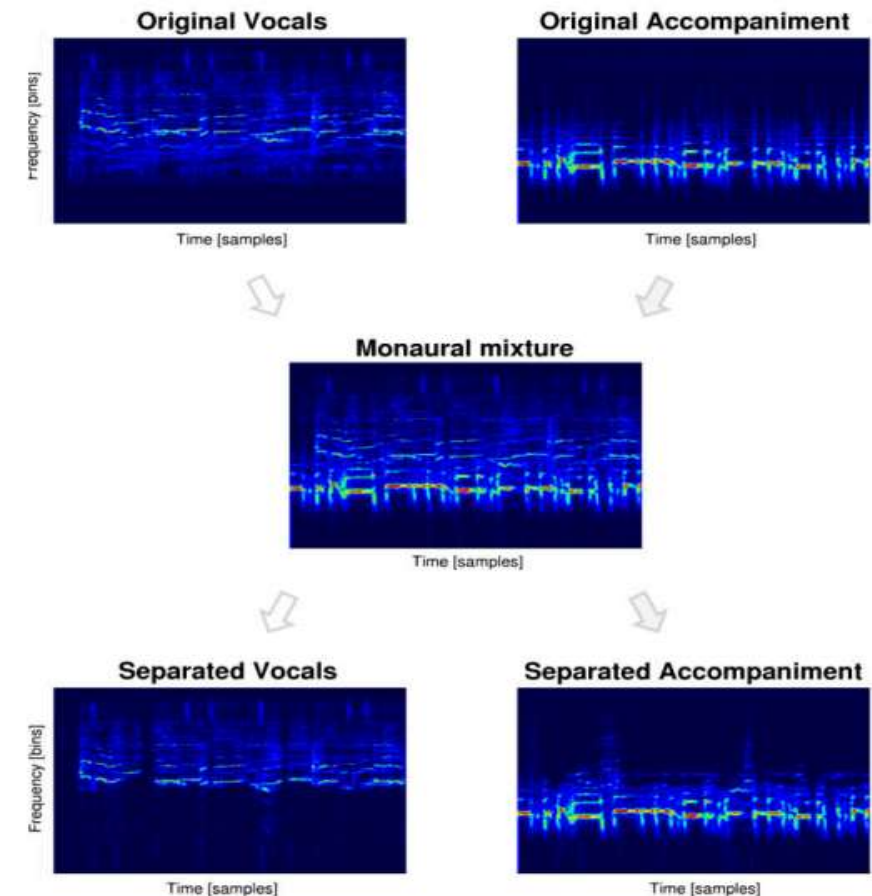
So this is Unsupervised Learning as we are not telling the algorithm in advance that these are Type 1, Type 2, ... Type n person

[Source: Su-In Lee, Dana Pe'er, Aimee Dudley, George Church, Daphne Koller]

Example: Cocktail Party Problem (Non-Clustering)

The cocktail party effect is the ability to focus on a specific human voice while filtering out other voices or background noise.

A particularly challenging cocktail party problem is in the field of music, where humans can easily concentrate on a singing voice superimposed on a musical background that includes a wide range of instruments. By comparison, machines are poor at this task.



To learn more about Cocktail Party Problem, refer the following links:

<https://www.technologyreview.com/s/537101/deep-learning-machine-solves-the-cocktail-party-problem/>:

<https://www.youtube.com/watch?v=T0HP9cxri0A>

<https://www.youtube.com/watch?v=2CD-3WLY8zq>

Of the following examples, which would you address using an unsupervised learning algorithm? (Check all that apply.)

- Given email labelled as spam/not spam, learn a spam filter.
- Given a set of news articles found on the web, group them into sets of articles about the same stories.
- Given a database of customer data, automatically discover market segments and group customers into different market segments.
- Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not.

A computer program is said to learn from experience E with respect to some task T and some performance measure P if its performance on T , as measured by P , improves with experience E .

Suppose we feed a learning algorithm a lot of historical weather data, and have it learn to predict weather. In this setting, what is T ?

- The process of the algorithm examining a large amount of historical weather data.
- The probability of it correctly predicting a future date's weather.
- The weather prediction
- None of these

Suppose you are working on weather prediction, and your weather station makes one of three predictions for each day's weather:

Sunny, Cloudy or Rainy. You'd like to use a learning algorithm to predict tomorrow's weather.

Would you treat this as a classification or a regression problem?

- Classification
- Regression

Suppose you are working on stock market prediction, Typically tens of millions of shares of Microsoft stock are traded (i.e., bought/sold) each day. You would like to predict the number of Microsoft shares that will be traded tomorrow.

Would you treat this as a classification or a regression problem?

- Classification
- Regression

Some of the problems below are best addressed using a supervised learning algorithm, and the others with an unsupervised learning algorithm. Which of the following would you apply supervised learning to? (Select all that apply.) In each case, assume some appropriate dataset is available for your algorithm to learn from.

- Take a collection of 1000 essays written on the US Economy, and find a way to automatically group these essays into a smaller number of groups of essays that are somehow “similar” or “related”.
- Examine a large collection of emails that are known to be spam email, to discover if there are sub-types of spam mail.
- Given genetic (DNA) data from a person, predict the odds of him/her developing diabetes over the next 10 years.
- Examine the statistics of two football teams, and predict which team will win tomorrow’s match (given historical data of teams’ wins/losses to learn from)

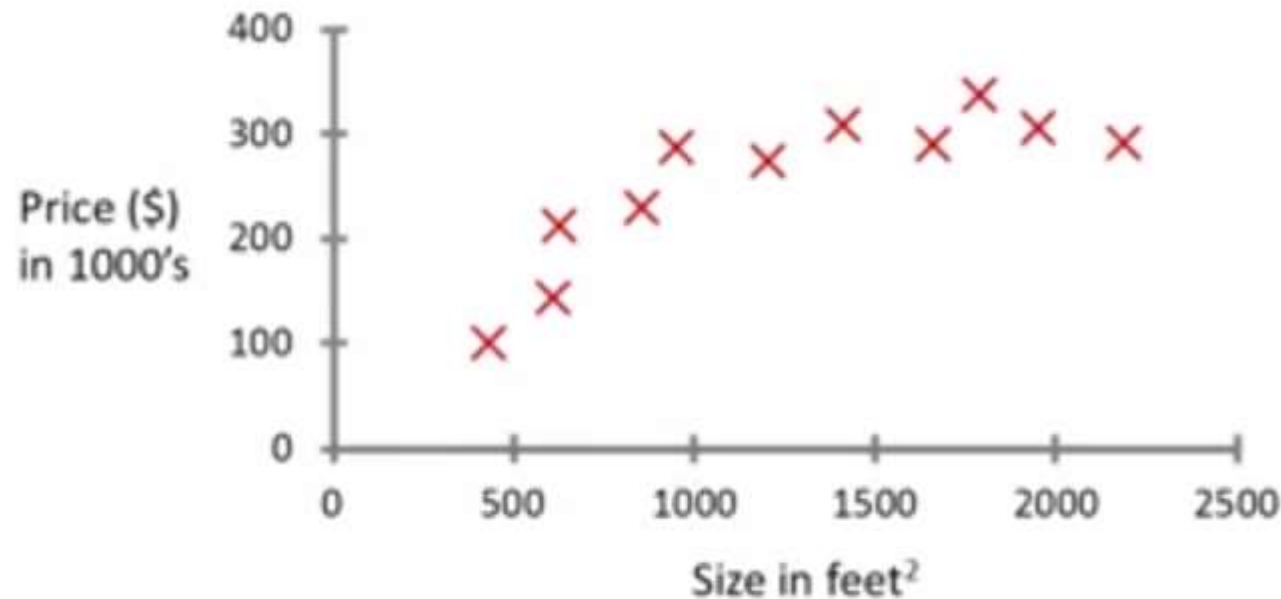
Which of these is a reasonable definition of machine learning?

- Machine Learning is the science of programming computers.
- Machine Learning learns from labelled data.
- Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.
- Machine Learning is the field of allowing robots to act intelligently.

Linear Regression with one variable

Given data about the size of houses on the real estate market, try to predict their price.

House Price Prediction



Case Study: Training set of housing prices at Portland, Oregon

Size in sq. feet (x)	Price (\$) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

Notation:

m = no. of training samples

x = input variables/features

y = output/target variable

(x, y) represents a training example

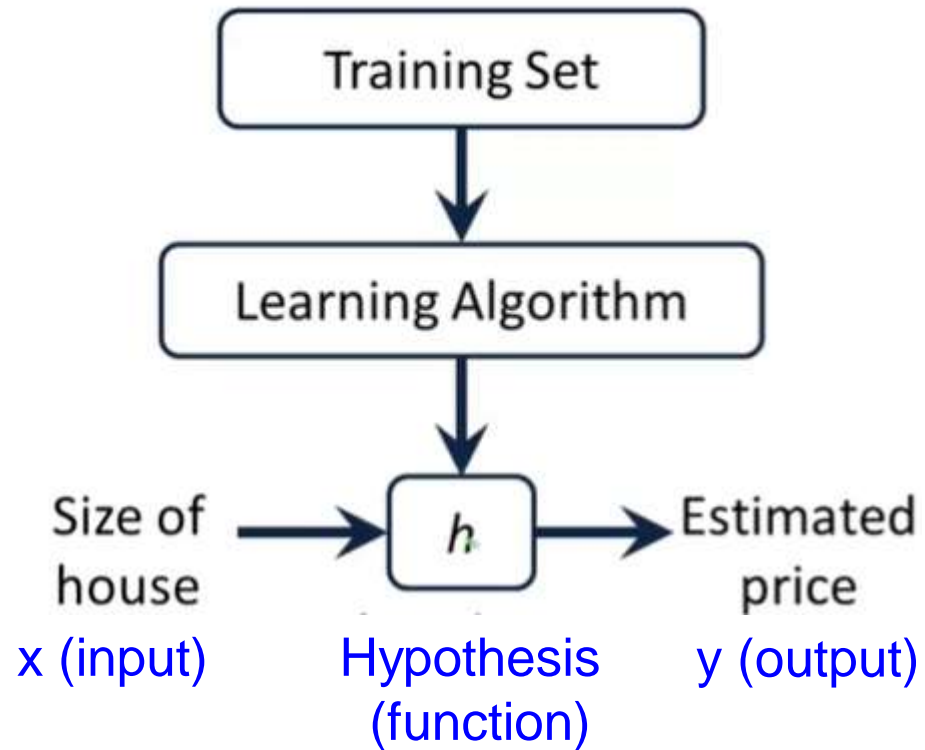
(x^i, y^i) represents i^{th} training example

Consider the training set shown below. $(x^{(i)}, y^{(i)})$ is the i^{th} training example. What is $y^{(3)}$?

Size in sq. feet (x)	Price (\$) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

- 1416
- 1534
- 315
- 0

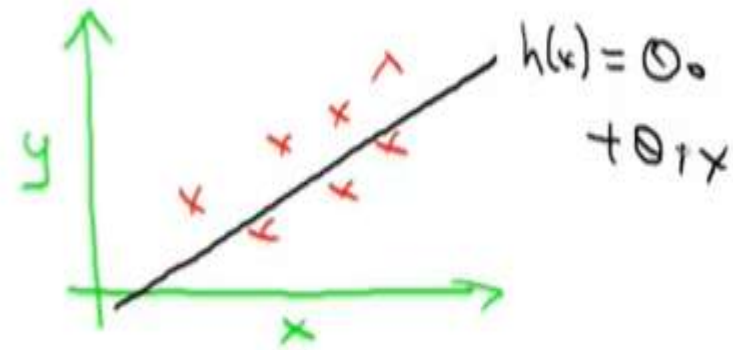
Model Representation



Function h maps from x to y
 $h: x \rightarrow y$

How do we represent h ?

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



This model is called “linear regression with one variable”

Cost Function

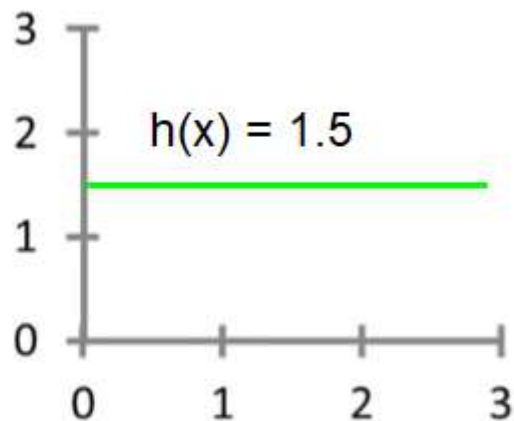
Cost function lets us figure out how to fit the best possible straight line to our data.

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

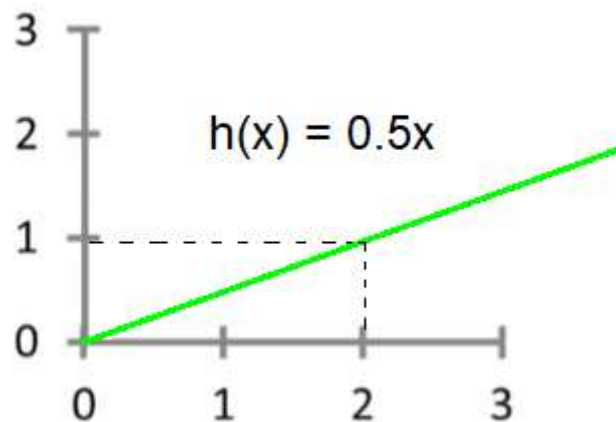
θ_0 , θ_1 are called parameters. How to choose them?

Using the above relation, plot graph for the given values of θ_0 , θ_1

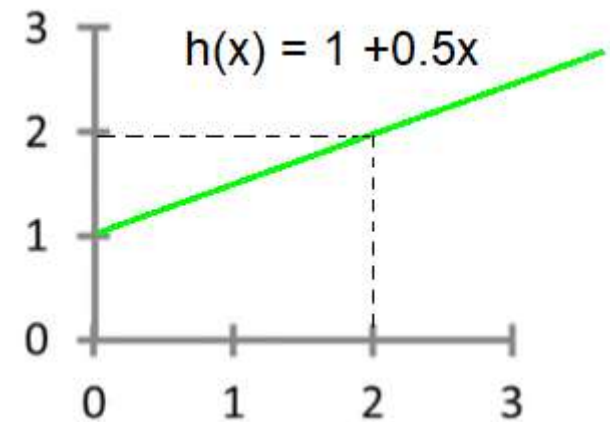
(i) $\theta_0 = 1.5$, $\theta_1 = 0$



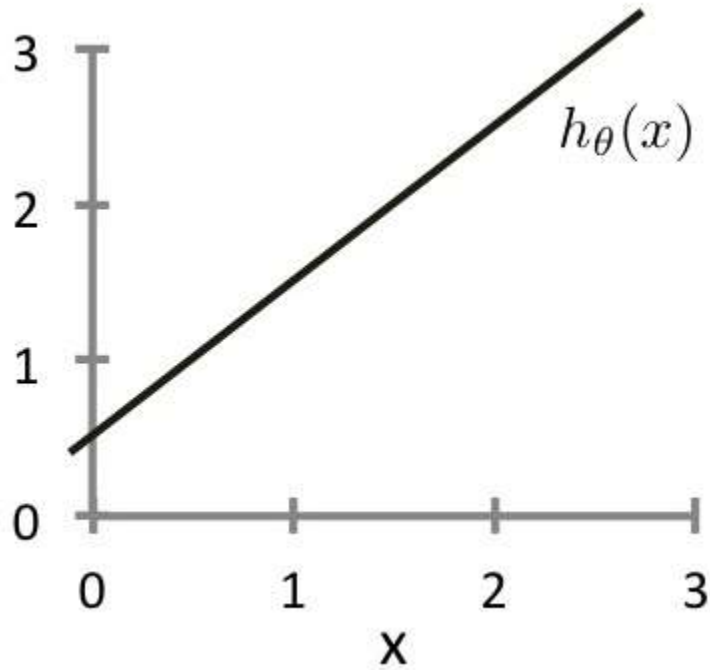
(ii) $\theta_0 = 0$, $\theta_1 = 0.5$



(iii) $\theta_0 = 1$, $\theta_1 = 0.5$

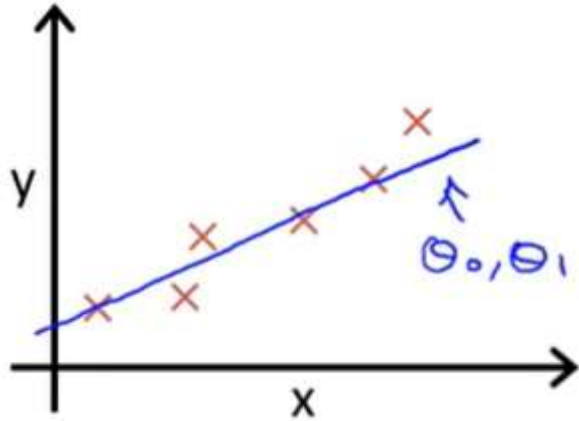


Consider the plot below of $h_{\theta}(x) = \theta_0 + \theta_1 x$
What are θ_0 and θ_1 ?



- $\theta_0 = 0, \theta_1 = 1$
- $\theta_0 = 0.5, \theta_1 = 1$
- $\theta_0 = 1, \theta_1 = 0.5$
- $\theta_0 = 1, \theta_1 = 1$

How to choose values of parameters?



Choose θ_0 and θ_1 such that $h_{\theta}(x)$ is close to y for our training examples (x, y)

Hence, the difference between y and $h_{\theta}(x)$ should be as small as possible for accurate predictions.

The general formula of **cost function (squared error function)** is given as follows:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m h_{\theta}((x^{(i)}) - y^{(i)})^2 \quad \text{and it should be as minimum as possible.}$$

Summary

Hypothesis:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

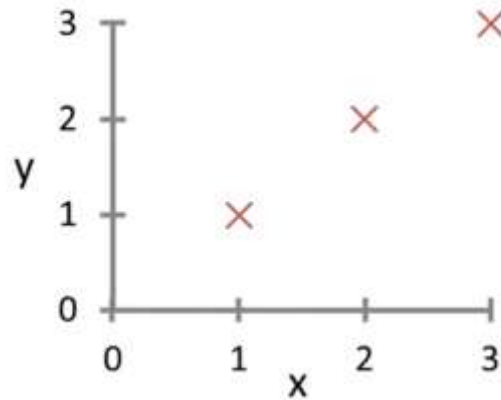
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m h_{\theta}((x^{(i)}) - y^{(i)})^2$$

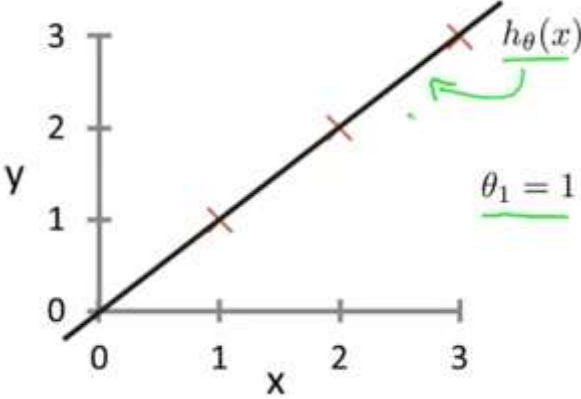
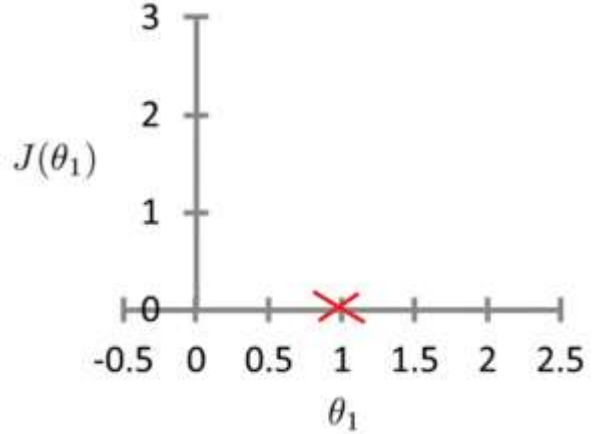
Goal:

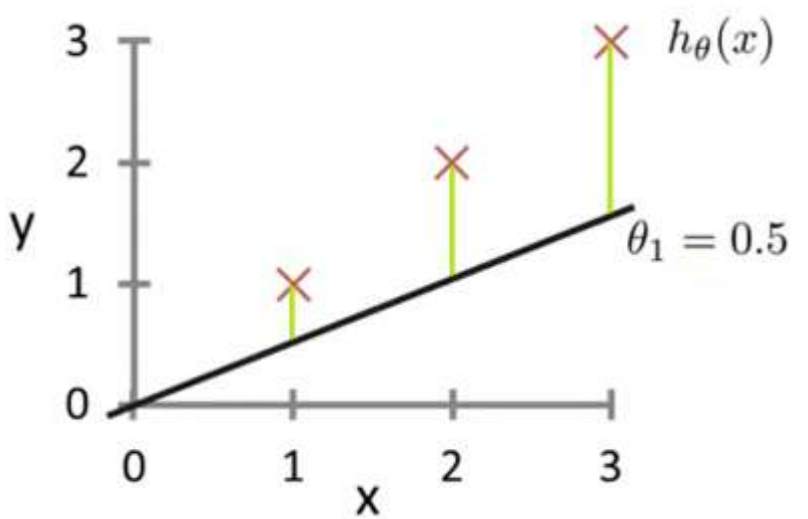
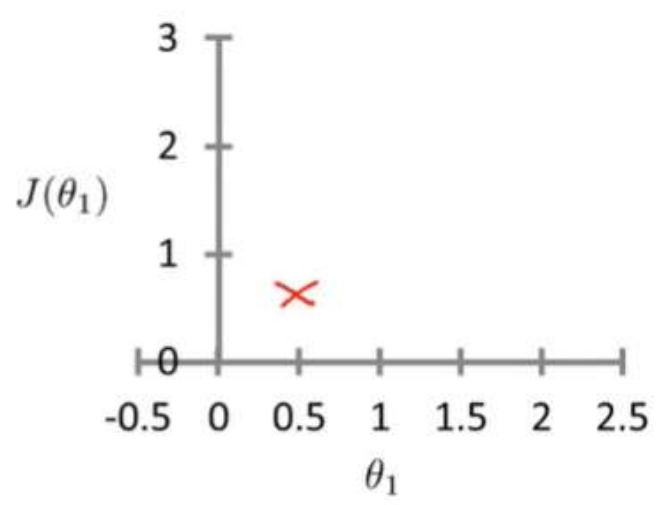
Minimise $J(\theta_0, \theta_1)$

Cost Function – Intuition I

Consider a given training set – (1,1), (2,2), (3,3)

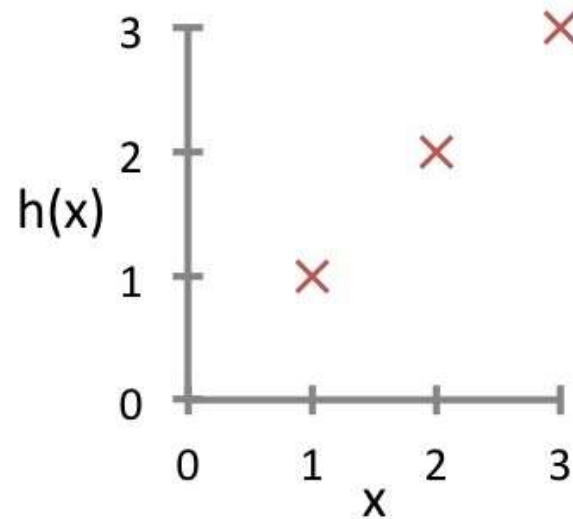


$h_{\theta}(x)$ (for fixed θ_1 , this is a function of x)	$J(\theta_1)$ (function of the parameter θ_1)
<p>For $\theta_0 = 0$ and $\theta_1 = 1$, hypothesis looks like this: $h_{\theta}(x) = x = y$</p>  <p>A scatter plot with the same three data points as the first figure. A solid black line, labeled $h_{\theta}(x)$, passes through all three points. A green arrow points to the line with the label $\theta_1 = 1$.</p>	<p>This means that, $J(1) = 0$</p>  <p>A graph of the cost function $J(\theta_1)$ versus θ_1. The y-axis is labeled $J(\theta_1)$ and ranges from 0 to 3. The x-axis is labeled θ_1 and ranges from -0.5 to 2.5 with tick marks at -0.5, 0, 0.5, 1, 1.5, 2, and 2.5. A red 'x' marks the point (1, 0) on the x-axis, indicating that the cost is zero when $\theta_1 = 1$.</p>

$h_{\theta}(x)$ (for fixed θ_1 , this is a function of x)	$J(\theta_1)$ (function of the parameter θ_1)
<p>For $\theta_0 = 0$ and $\theta_1 = 0.5$, hypothesis looks like this: $h_{\theta}(x) = 0.5x$</p> 	<p>This means that, $J(0.5) = 0.58$</p> 

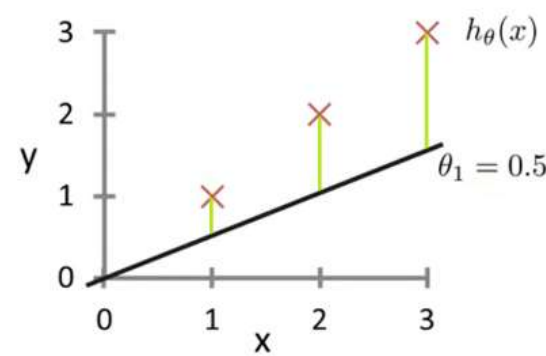
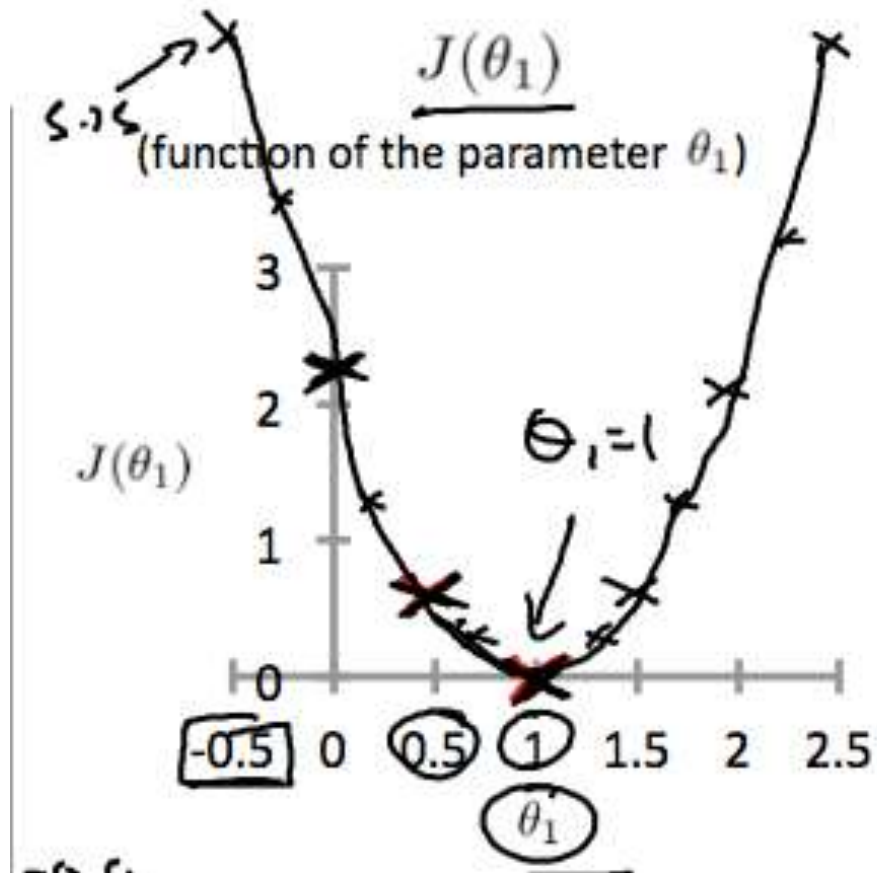
Suppose we have a training set with $m=3$ examples, plotted below. Our hypothesis representation is $h_{\theta}(x) = \theta_1 x$, with parameter θ_1 .

The cost function $J(\theta_1)$ is $J(\theta_1) = \frac{1}{2m} \sum_{i=1}^m h_{\theta}((x^{(i)}) - y^{(i)})^2$ What is $J(0)$?

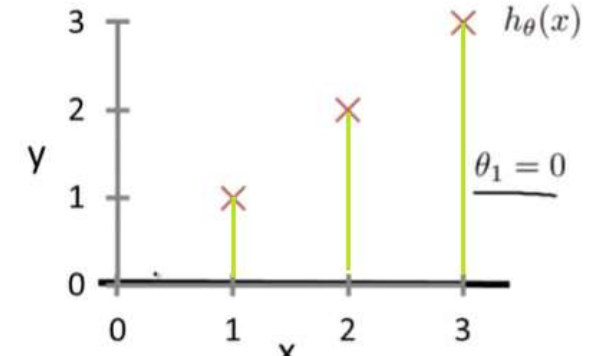


- 0
- 1/6
- 1
- 14/6

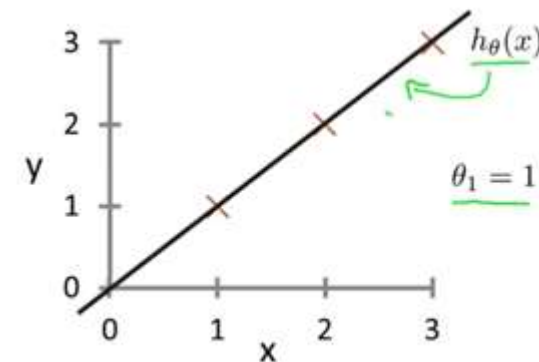
Plotting several other points yields to the following graph:



Case I: $J(0.5) = 0.58$



Case III: $J(0) = 2.33$



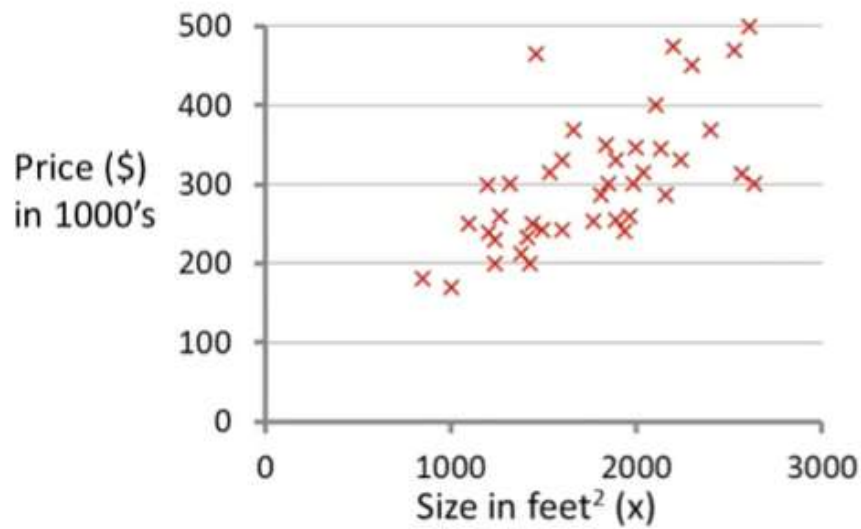
Case II: $J(1) = 0$

Best Solution

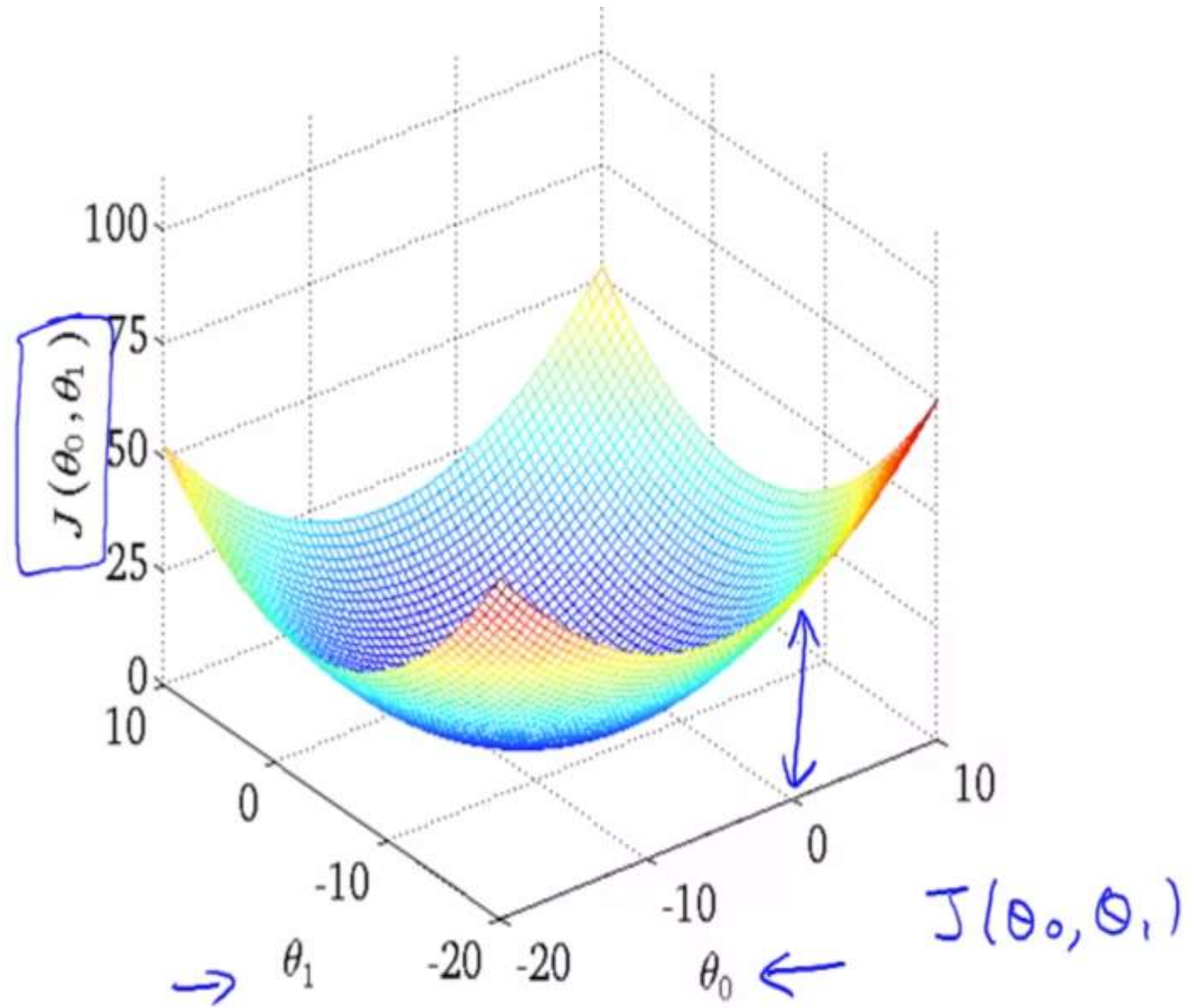
Thus as a goal, we should try to minimize the cost function.
In this case, $\theta_1 = 1$ is our global minimum which is the best possible fit.

Cost Function – Intuition II

Unlike Intuition I, in Intuition II, we retain both the parameters θ_0 , θ_1
Consider a given training set as follows:

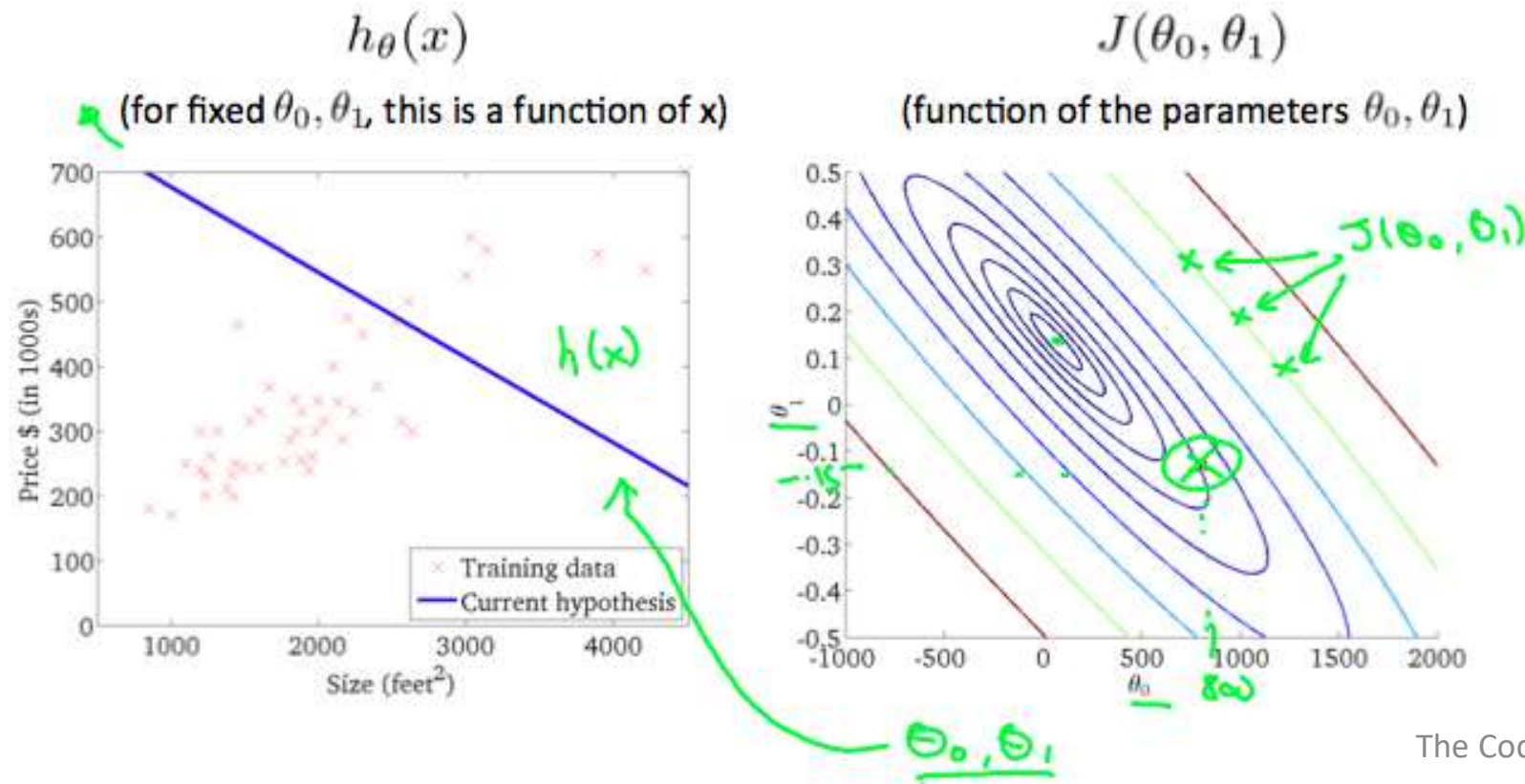


$h_{\theta}(x)$ (for fixed θ_0 , θ_1 , this is a function of x)	$J(\theta_0, \theta_1)$ (function of the parameters θ_0 , θ_1)
<p>A scatter plot similar to the one on the left, but with a black line representing the linear hypothesis $h_{\theta}(x) = 50 + 0.06x$ drawn through the data points. The axes are the same as the previous plot.</p> <p>$h_{\theta}(x) = 50 + 0.06x$</p> <p>For $\theta_0 = 50$, $\theta_1 = 0.06$</p>	<p>A 3D surface plot of the cost function $J(\theta_0, \theta_1)$. The vertical axis represents the cost value, ranging from 0 to 100. The horizontal axes represent the parameters θ_0 and θ_1, both ranging from -20 to 10. The surface is a bowl-shaped paraboloid, colored with a gradient from blue at the bottom to yellow at the top. A blue arrow points to the surface, and a blue box highlights the label $J(\theta_0, \theta_1)$.</p> <p>A complex 3D surface. In 2D, we represent it in contours</p>

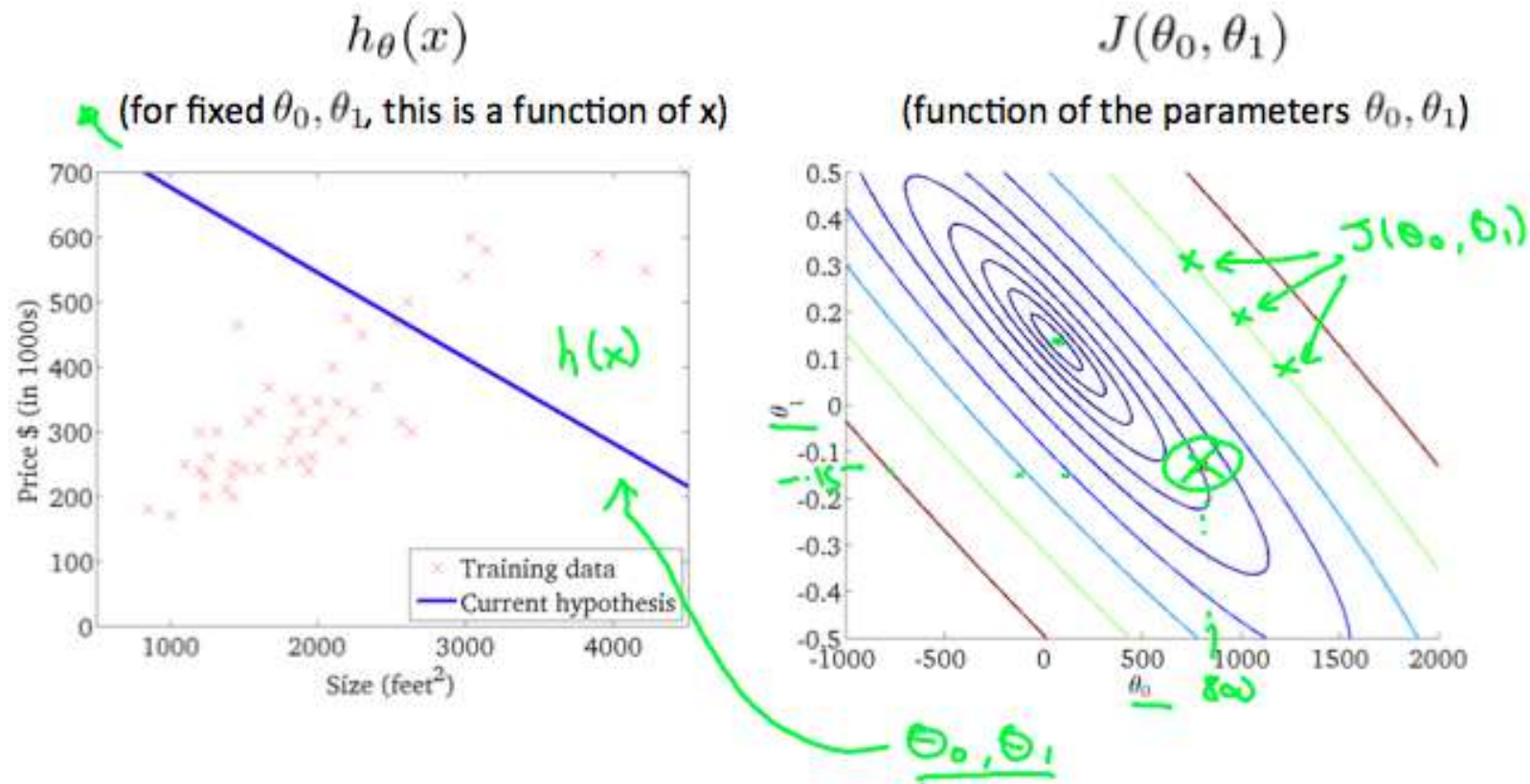


Contours

- A contour plot is a graph that contains many contour lines.
- A contour line of a two variable function has a constant value at all points of the same line.
- An example of such a graph is the one to the right below.



- Taking any color and going along the 'circle', one would expect to get the same value of the cost function.
- For example, the three green points found on the green line above have the same value for $J(\theta_0, \theta_1)$ and as a result, they are found along the same line.

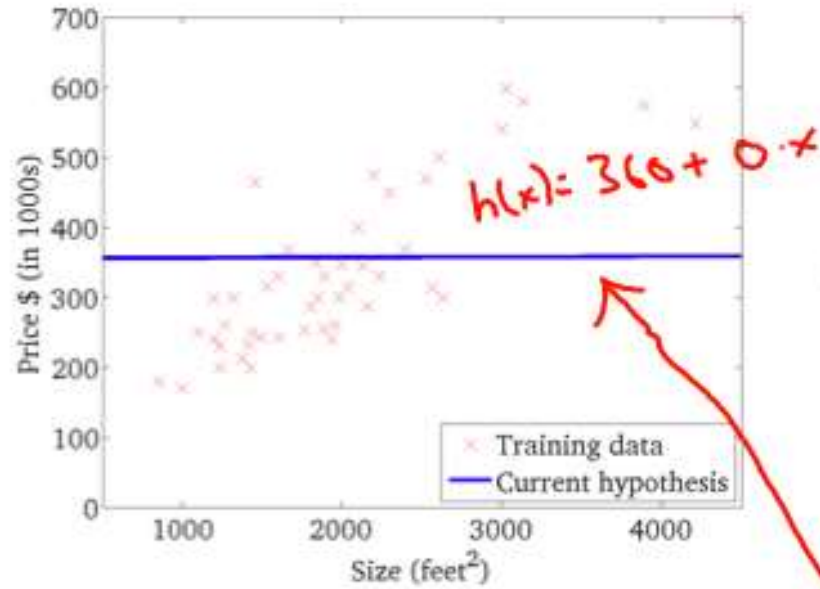


$$h(x) = 50 + 0.06x$$

$$\text{For } \theta_0 = 50, \theta_1 = 0.06$$

$$h_{\theta}(x)$$

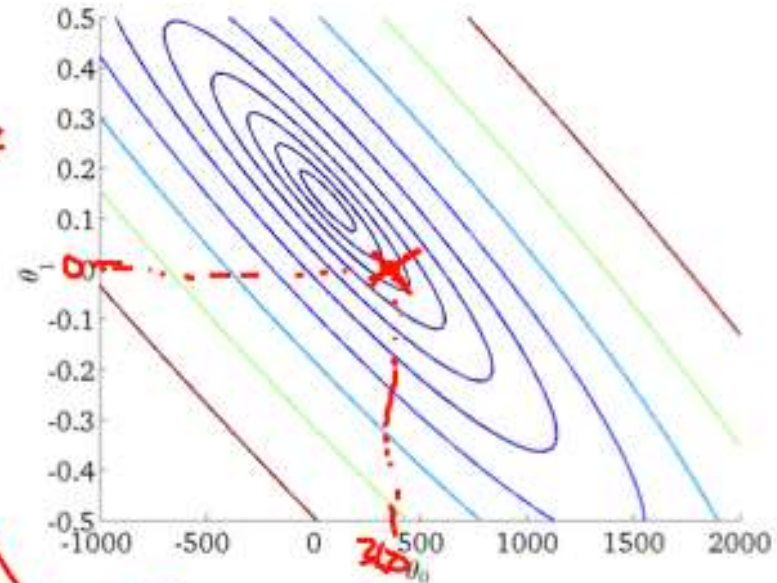
(for fixed θ_0, θ_1 , this is a function of x)



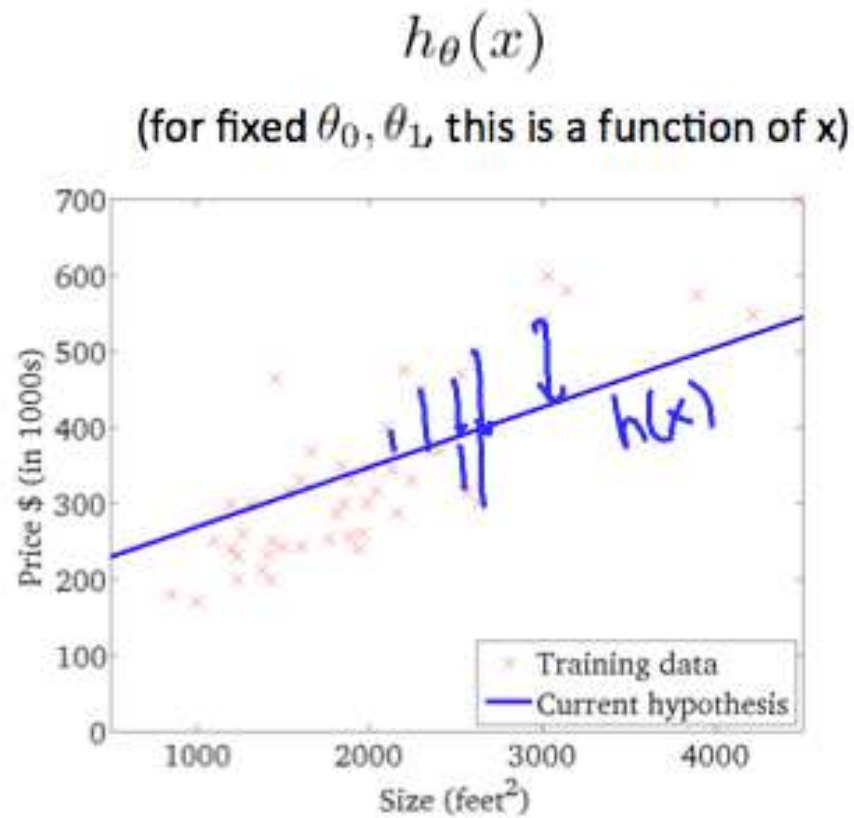
$$h(x) = 360$$

$$J(\theta_0, \theta_1)$$

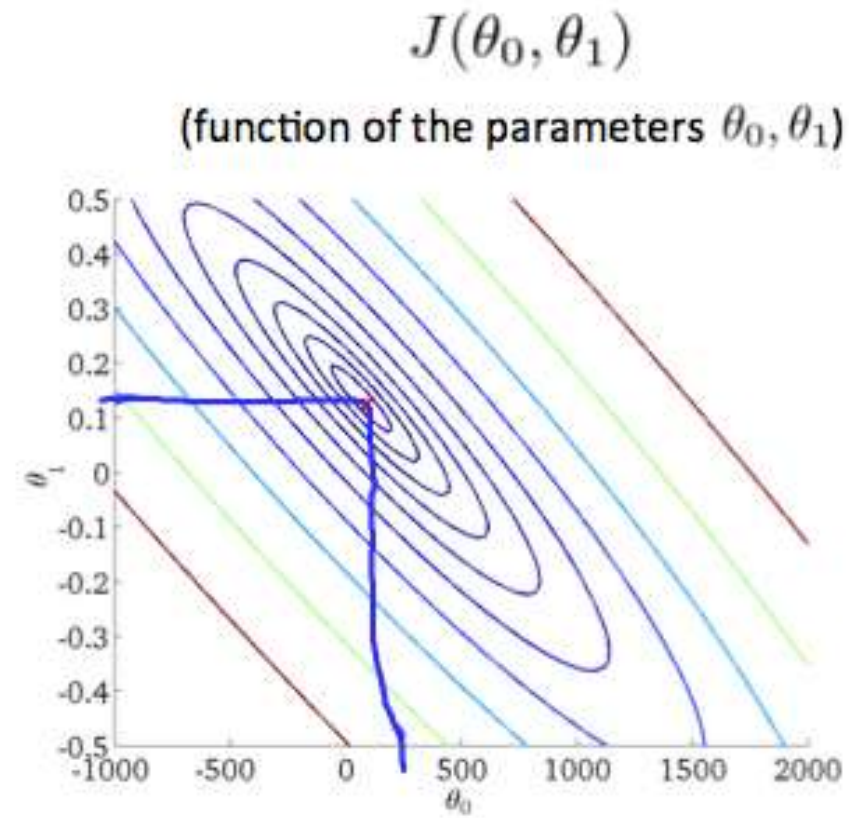
(function of the parameters θ_0, θ_1)



$$\text{For } \theta_0 = 360, \theta_1 = 0$$



$$h(x) = 250 + 0.12x$$

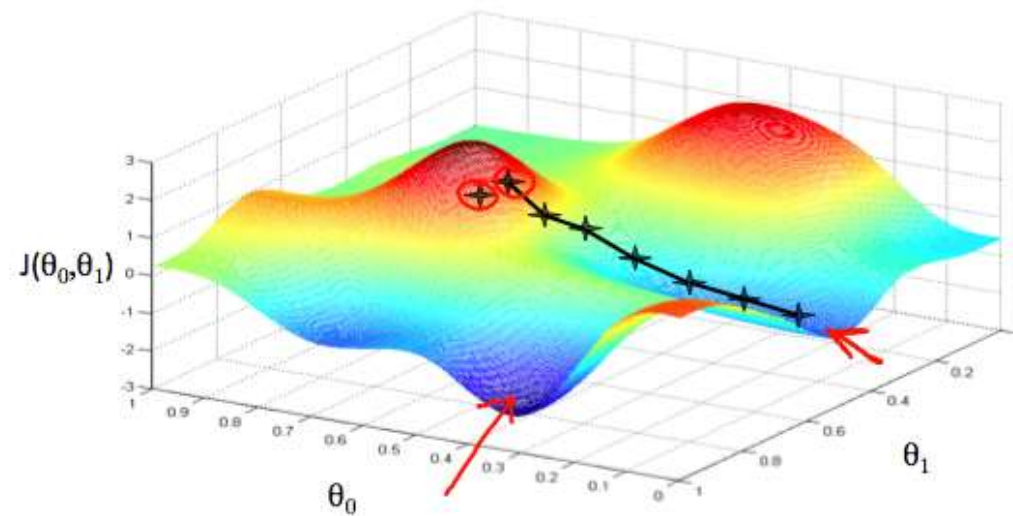


$$\text{For } \theta_0 = 250, \theta_1 = 0.12$$

The graph above minimizes the cost function as much as possible. Plotting those values on our graph to the right seems to put our point in the center of the inner most 'circle'. Hence, best possible solution.

Gradient Descent

- It's not possible to manually find out values of θ_0 , θ_1 to minimise $J(\theta_0, \theta_1)$
- Gradient Descent is an algorithm used to do this task.



- In the figure, the height of surface represents value of $J(\theta_0, \theta_1)$
- Deeper the surface, smaller is the height of $J(\theta_0, \theta_1)$
- The goal is to reach to the deepest possible surface.

Gradient Descent

- Imagine you are standing at a point on a hill.
- You look all around and find the best direction that will take you down the hill **most quickly** and you take a step towards that direction.
- Now you are at a new point (at a lower point).
- From this new point, you look around, decide what direction would take you downhill most quickly. Take a step towards it.
- Repeat the procedure until you converge down to the local minimum.



The Algorithm

repeat until convergence

{

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad (\text{for } j=0, 1)$$

}

α = Learning Rate (How big a step is taken down the hill)

At each iteration, values of θ_0, θ_1 , must be simultaneously updated.

Correct: Simultaneous update

→ $\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$
→ $\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$
→ $\theta_0 := \text{temp0}$
→ $\theta_1 := \text{temp1}$

Incorrect:

→ $\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$
→ $\theta_0 := \text{temp0}$
→ $\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$
→ $\theta_1 := \text{temp1}$

Suppose $\theta_0 = 1$, $\theta_1 = 2$ and we simultaneously update θ_0 and θ_1 using the rule: $\theta_j := \theta_j + \sqrt{\theta_0 \theta_1}$ (for $j = 0$ and $j=1$). What are the resulting values of θ_0 and θ_1 ?

- $\theta_0 = 1, \theta_1 = 2$
- $\theta_0 = 1 + \sqrt{2}, \theta_1 = 2 + \sqrt{2}$
- $\theta_0 = 2 + \sqrt{2}, \theta_1 = 1 + \sqrt{2}$
- $\theta_0 = 1 + \sqrt{2}, \theta_1 = \sqrt{(1 + \sqrt{2}) \cdot 2}$

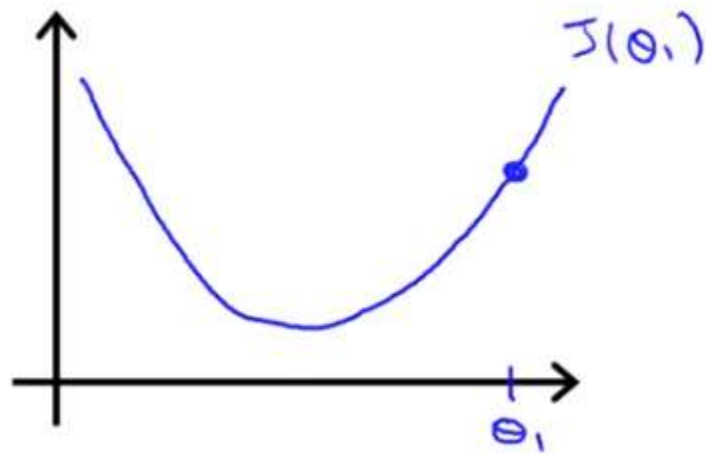
Gradient Descent - Intuition

For one parameter θ_1 , the gradient descent is given by –

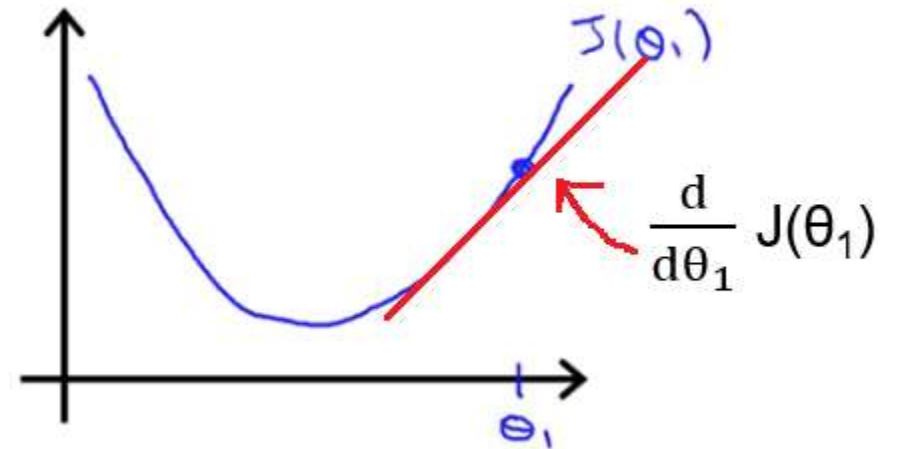
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

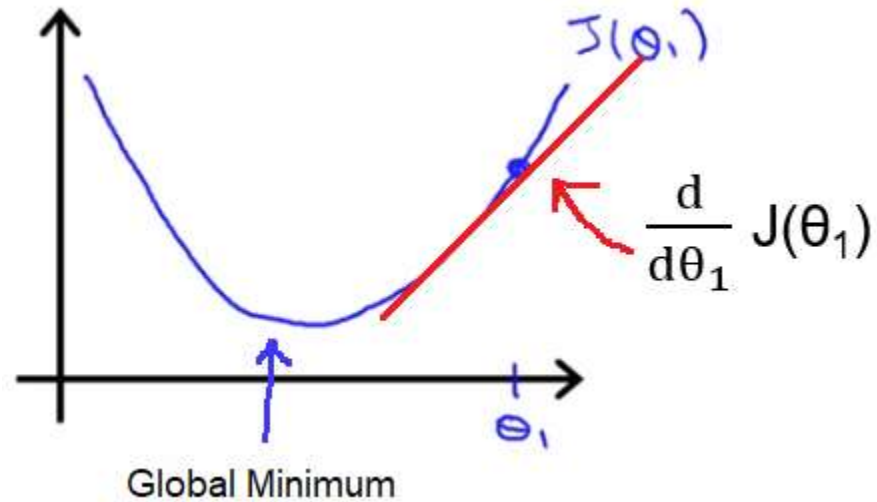
$$\frac{d}{d\theta_1} J(\theta_1)$$

Consider a plot as shown below – Case I:



Derivative of θ_1 gives a tangent at that point





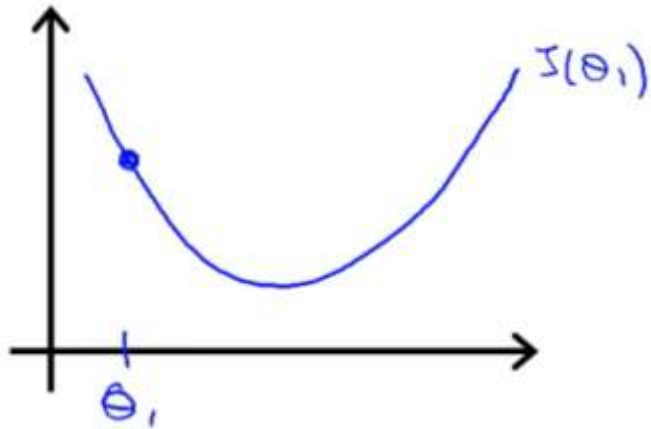
The tangent at θ_1 is a positive slope

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

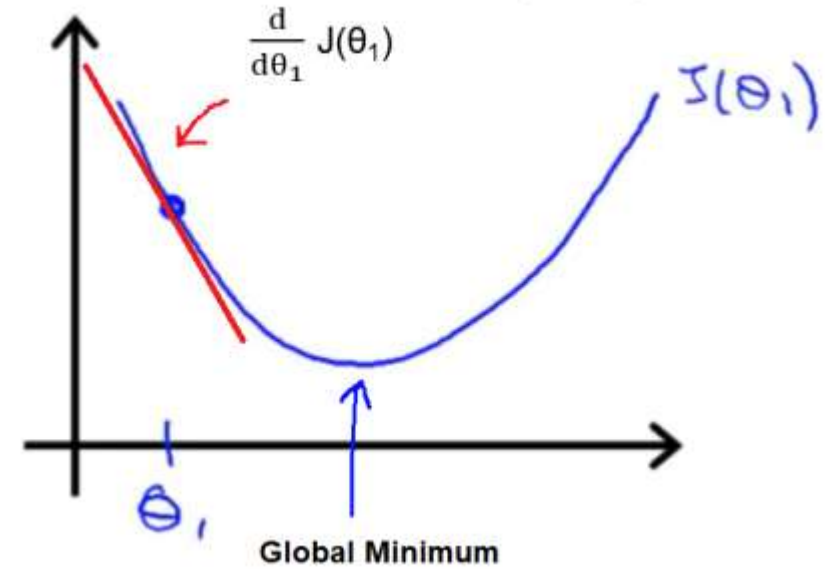
Positive Quantity

Thus, value of θ_1 **reduces** gradually (moves towards left as you can see in the figure) to reach the global minimum.

Case II:



Derivative of θ_1 gives a tangent at that point



The tangent at θ_1 is a negative slope

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

Negative Quantity

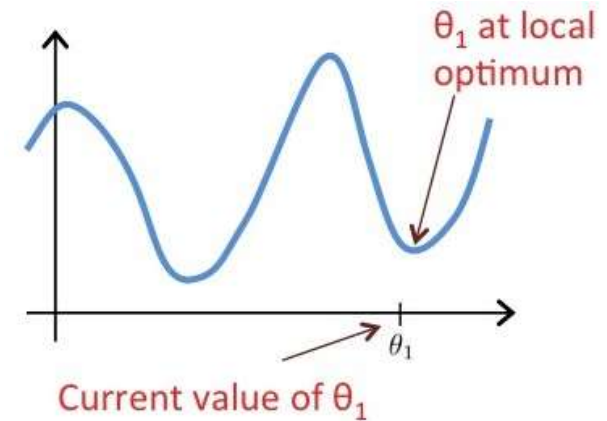
Thus, value of θ_1 **increases** gradually (moves towards right as you can see in the figure) to reach the global minimum.

Case III:

Suppose θ_1 is at a local optimum of $J(\theta_1)$ such as shown in the figure.

What will one step of gradient descent $\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$ do?

- Leave θ_1 unchanged
- Change θ_1 in a random direction
- Move θ_1 in the direction of global minimum of $J(\theta_1)$
- Decrease θ_1



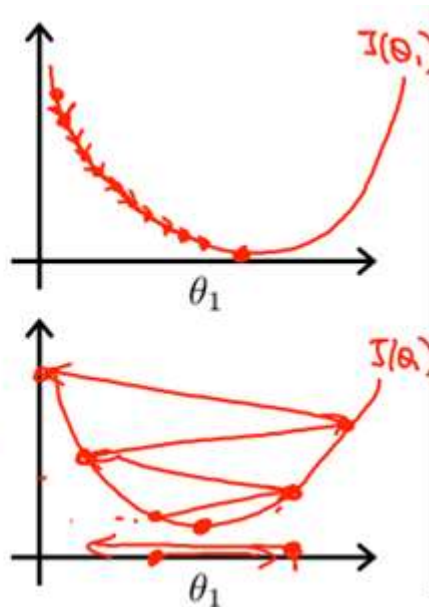
α should be properly adjusted

- ensure that the gradient descent algorithm converges in a reasonable time.
- failure to converge or too much time to obtain the minimum value imply that our step size is wrong.

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

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.



How does gradient descent converge with a fixed step size α ?

The intuition behind the convergence is that $\frac{d}{d\theta_1} J(\theta_1)$ approaches 0 as we approach the bottom of our convex function.

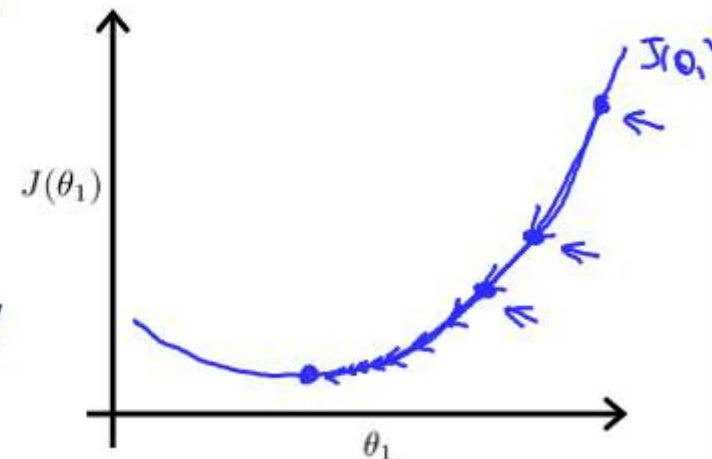
At the minimum, the derivative will always be 0 and thus we get:

$$\theta_1 := \theta_1 - \alpha * 0$$

Gradient descent can converge to a local minimum, even with the learning rate α fixed.

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

As we approach a local minimum, gradient descent will automatically take smaller steps. So, no need to decrease α over time.



Gradient Descent with Linear Regression

Gradient descent algorithm

repeat until convergence {
 $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$
 (for $j = 1$ and $j = 0$)
}

Linear Regression Model

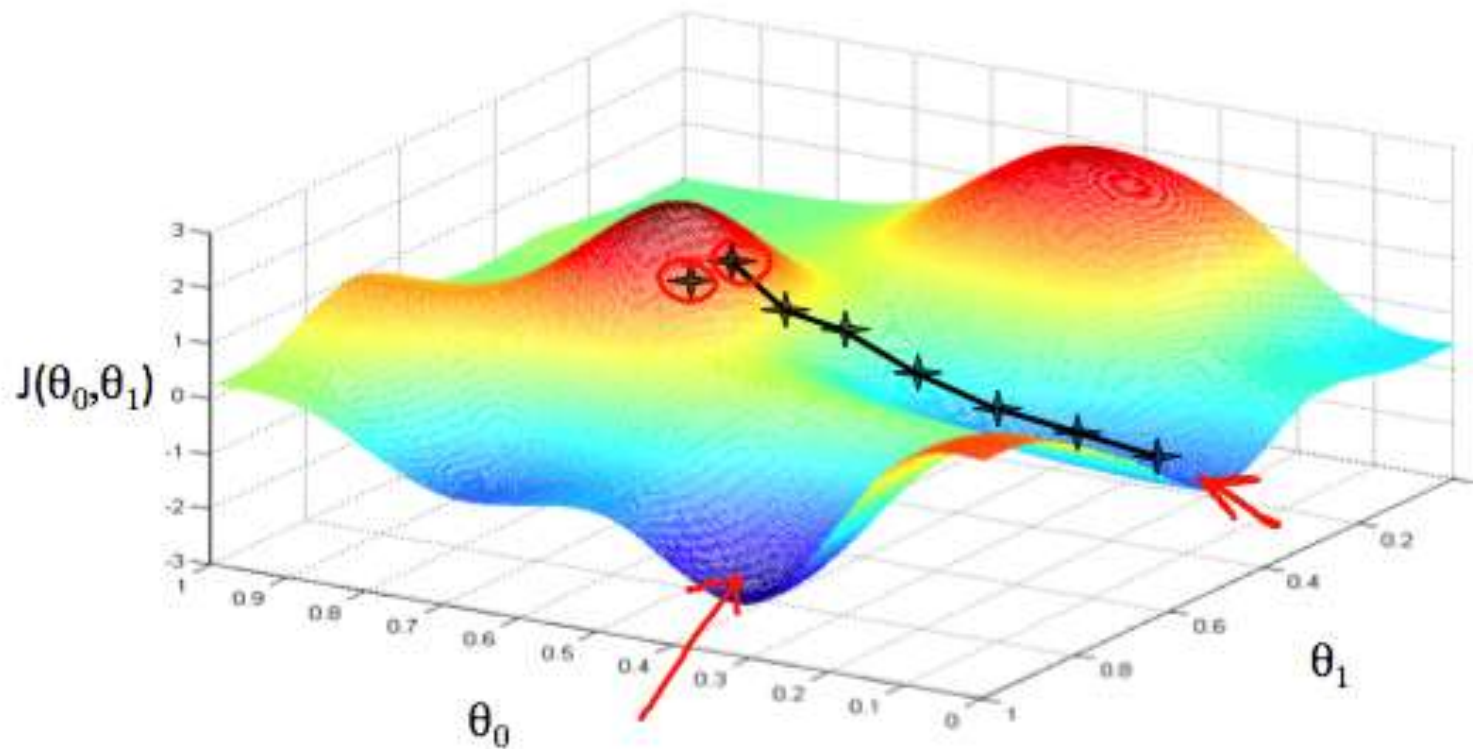
$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

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

We combine the two together and partially derivate $J(\theta_0, \theta_1)$ w.r.t. θ_0 & θ_1

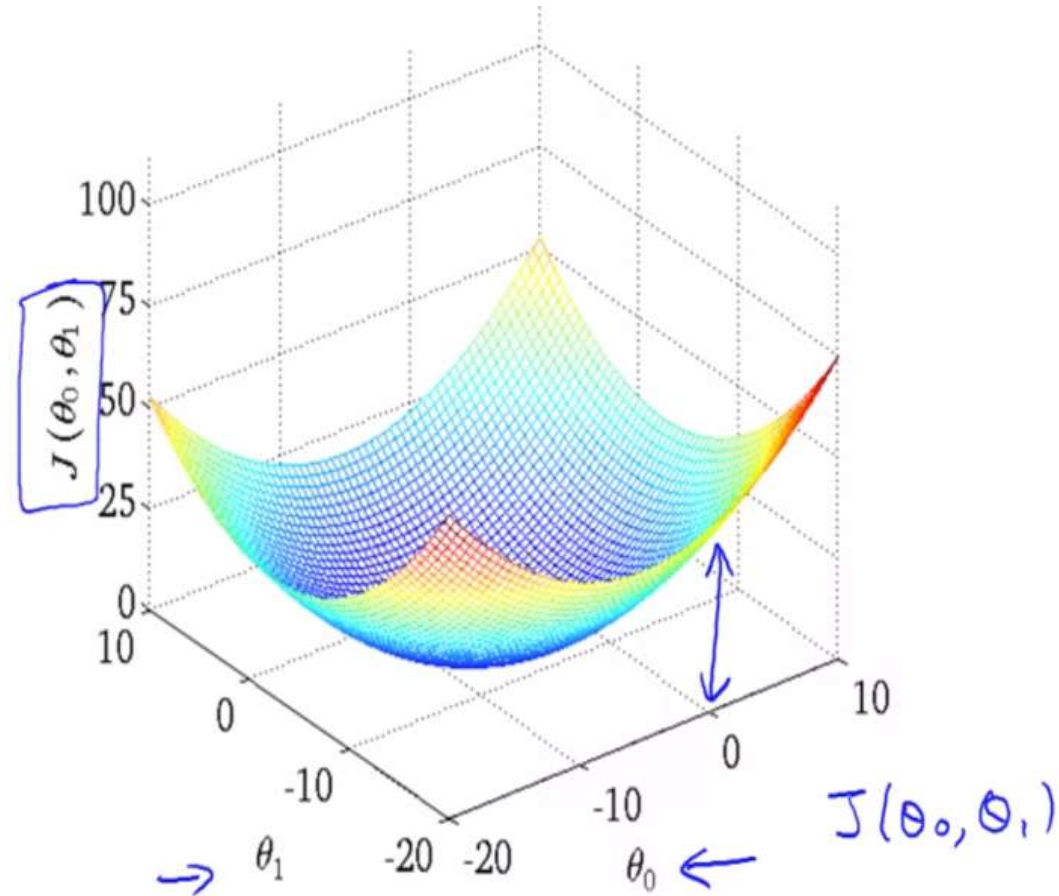
repeat until convergence {
 $\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})$
 $\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x^{(i)}$
}

Now, we keep substituting the values of θ_0 , θ_1 in the function (figure shown below) until we reach the **global optima** or simply, converge.



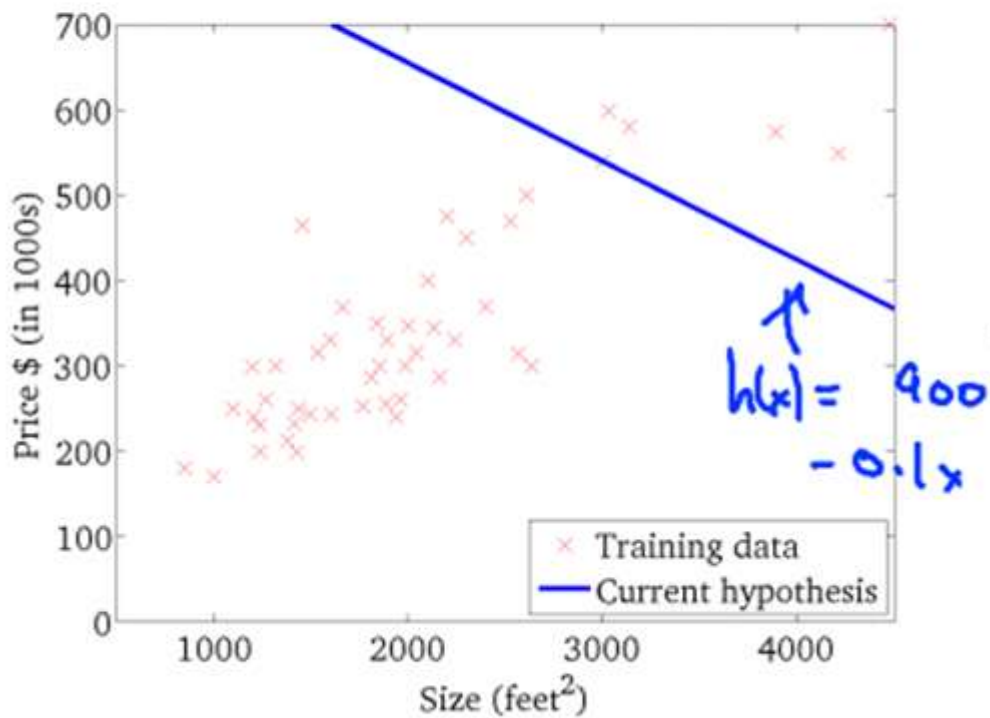
Note that, while gradient descent can be susceptible to local minima in general, the optimization problem we have posed here for linear regression has only one global, and no other local optima; thus gradient descent always converges.

In other words, the cost function for linear regression is always a “**bowl-shaped function**”. This is also called as a “**convex function**”.



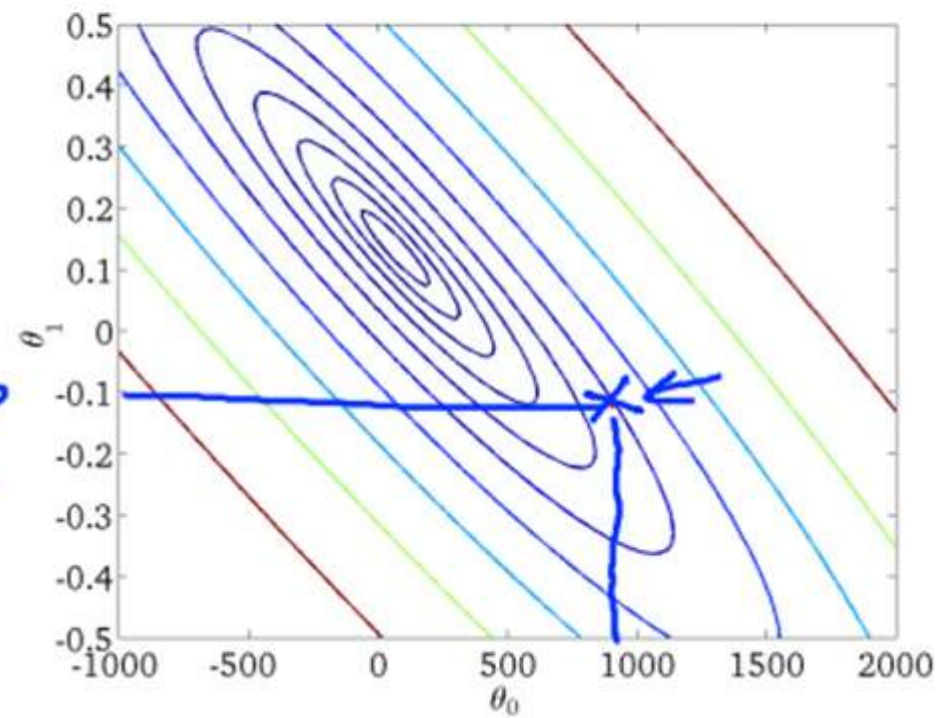
$$\underline{h_{\theta}(x)}$$

(for fixed θ_0, θ_1 , this is a function of x)



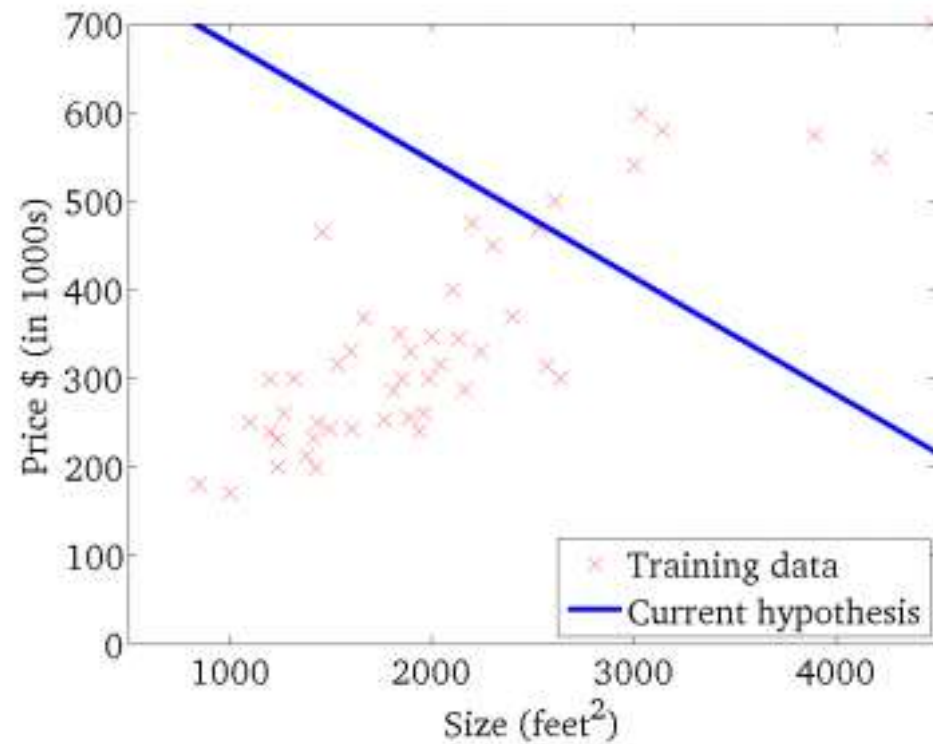
$$\underline{J(\theta_0, \theta_1)}$$

(function of the parameters θ_0, θ_1)



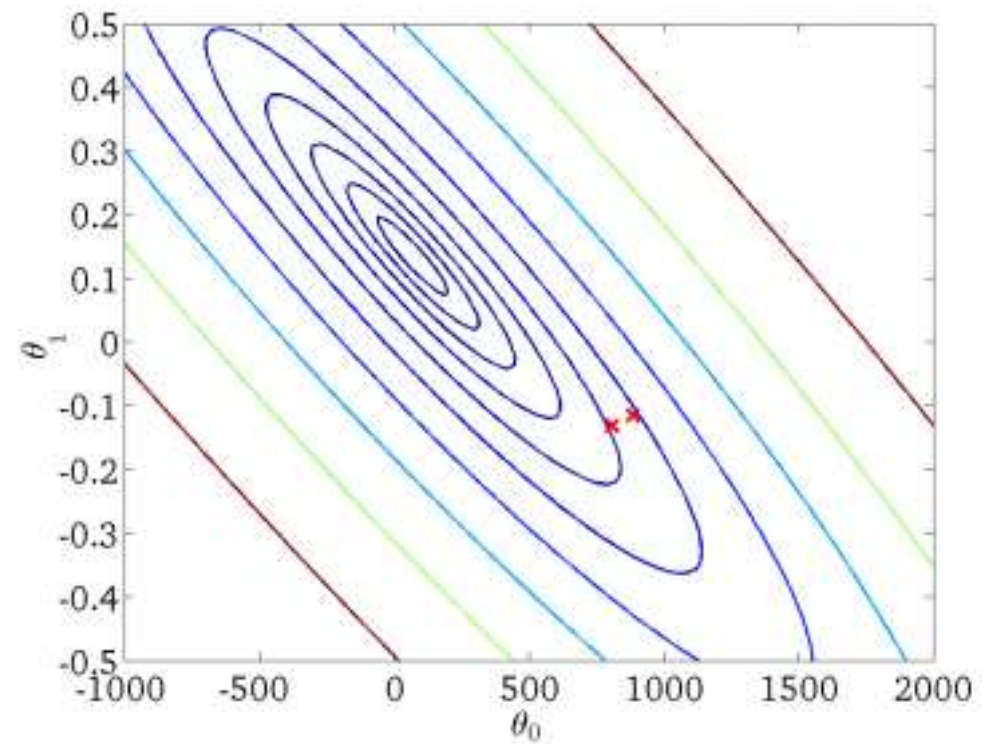
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



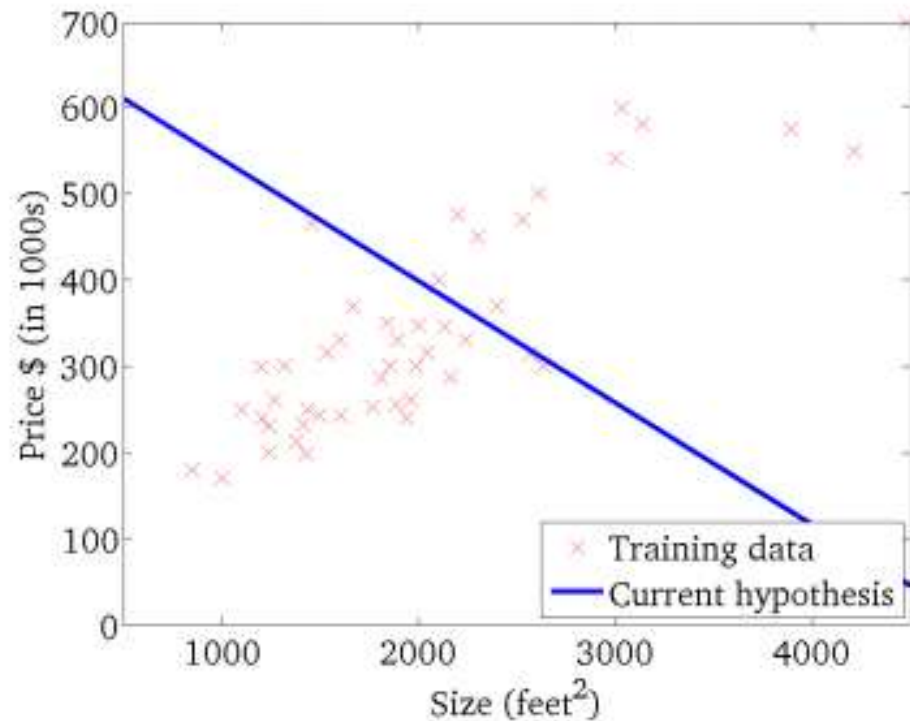
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



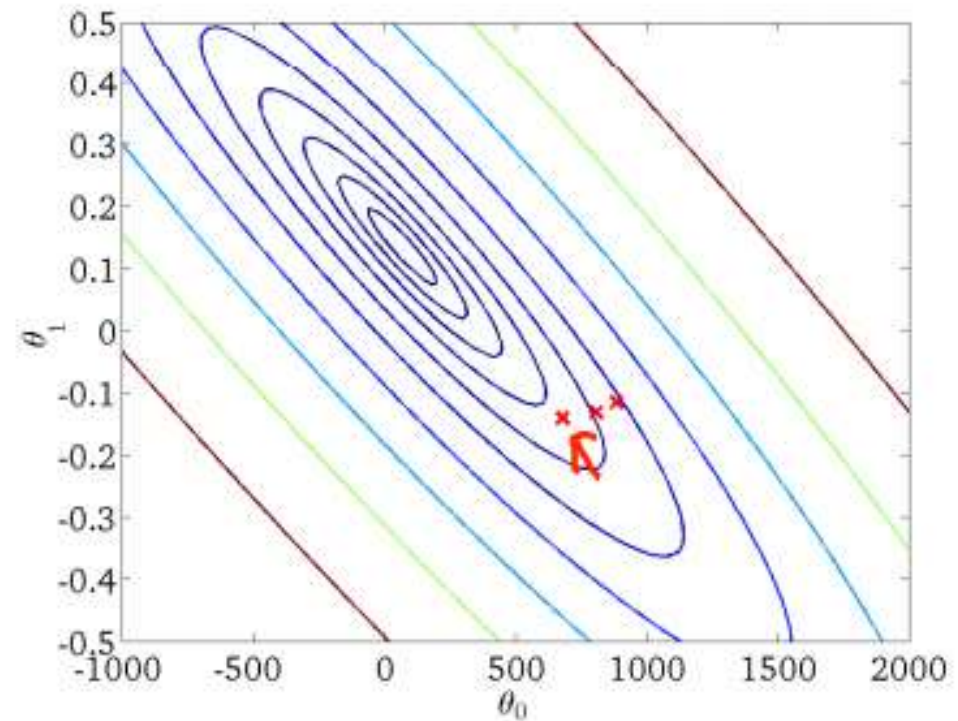
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



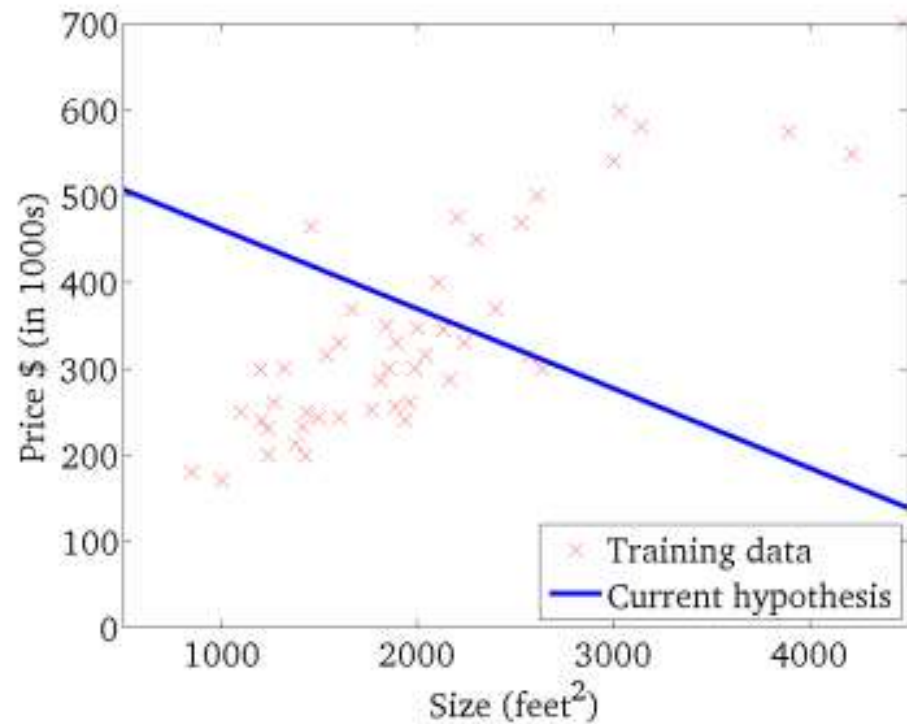
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



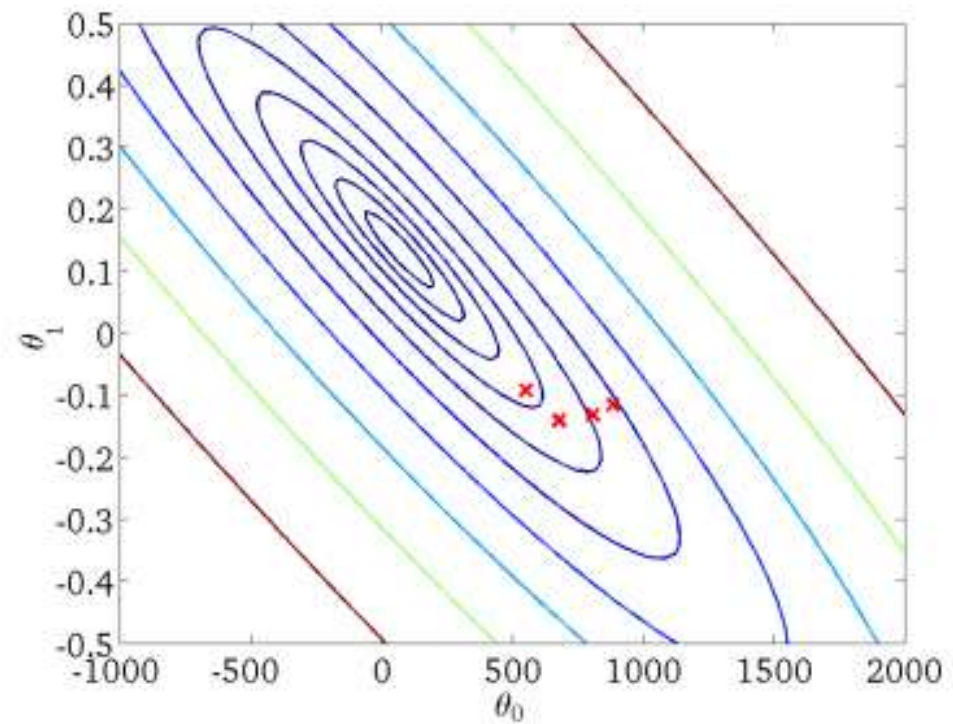
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



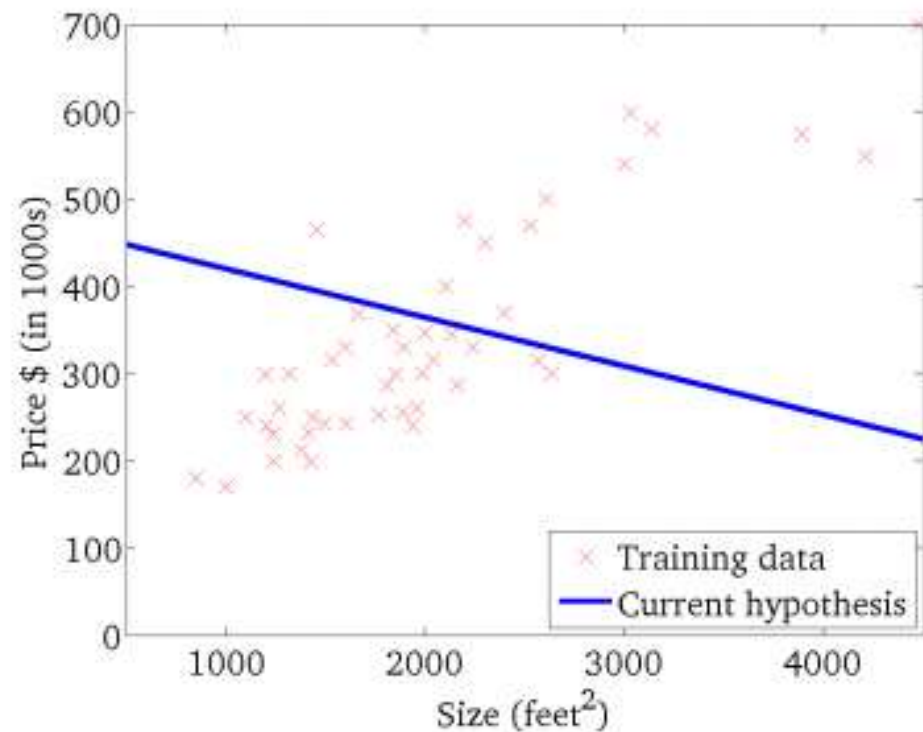
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



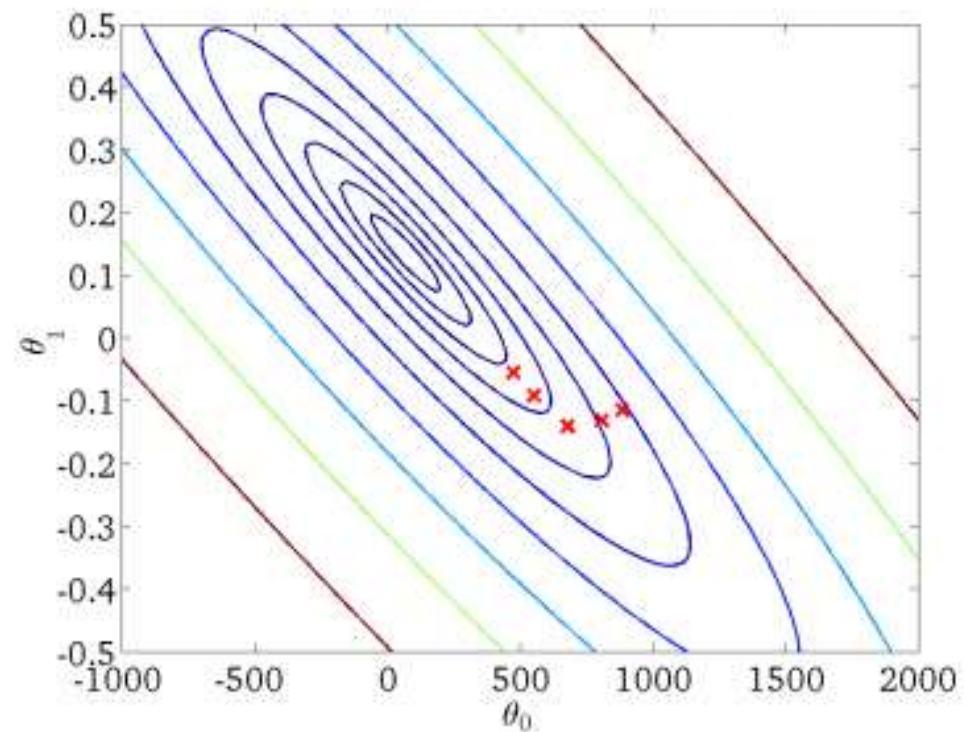
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



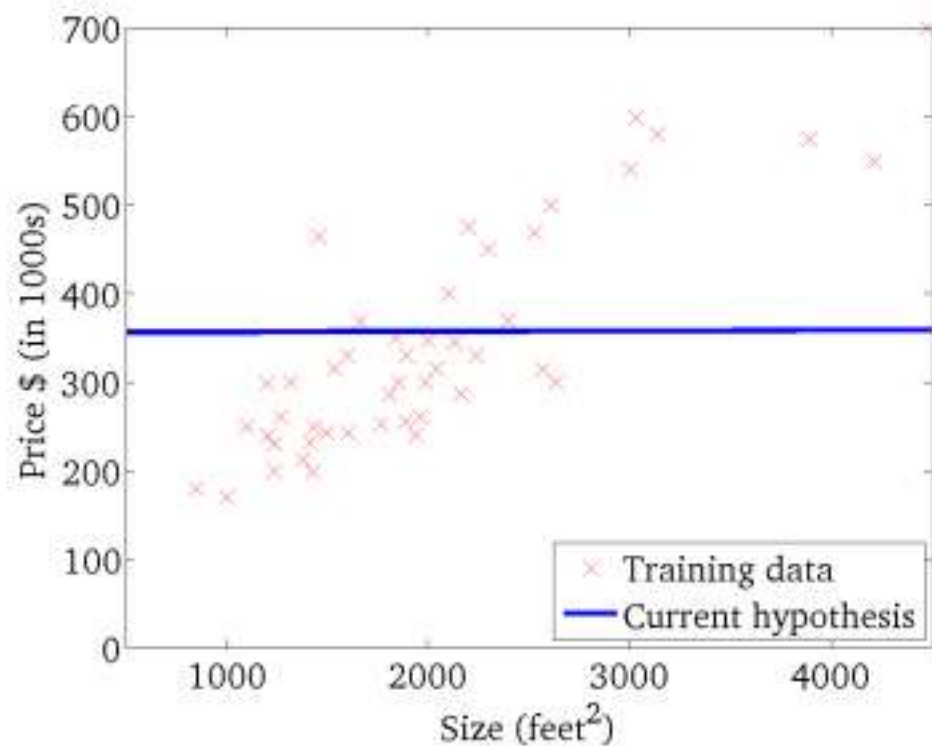
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



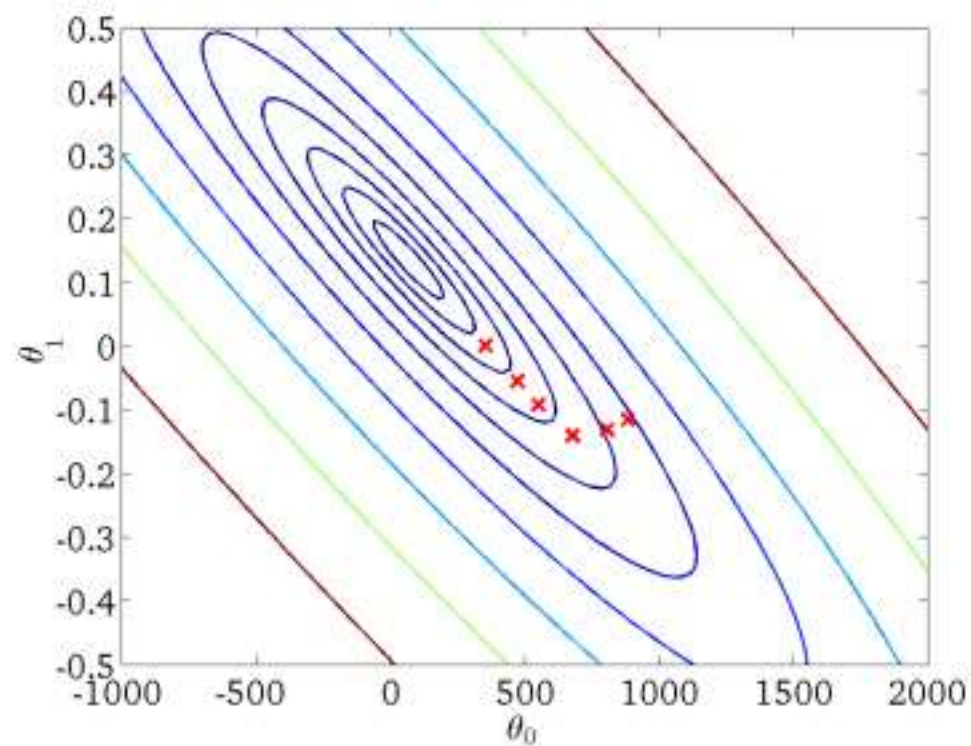
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



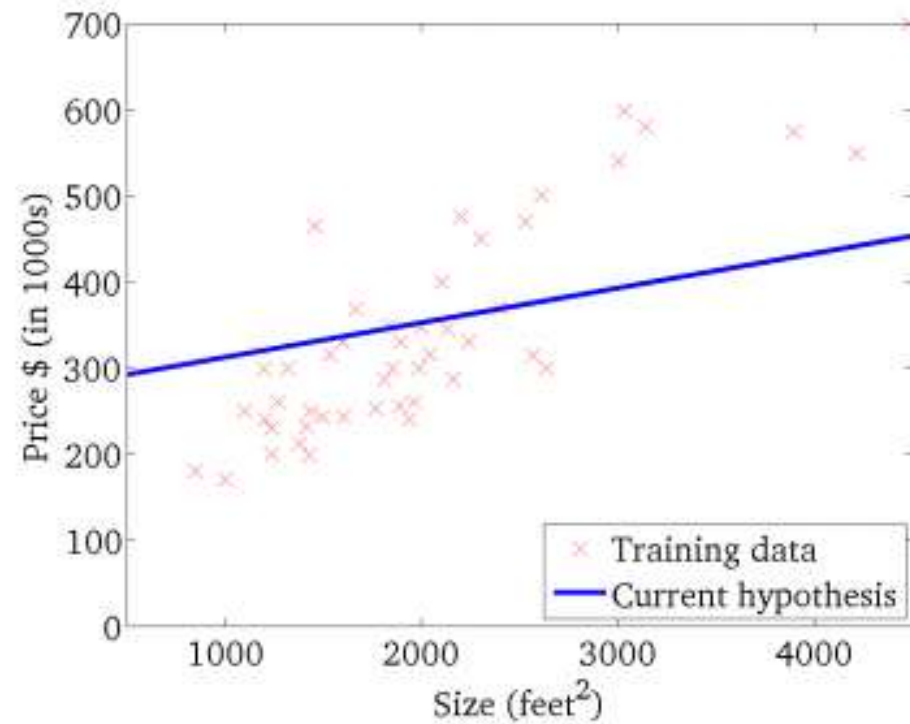
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



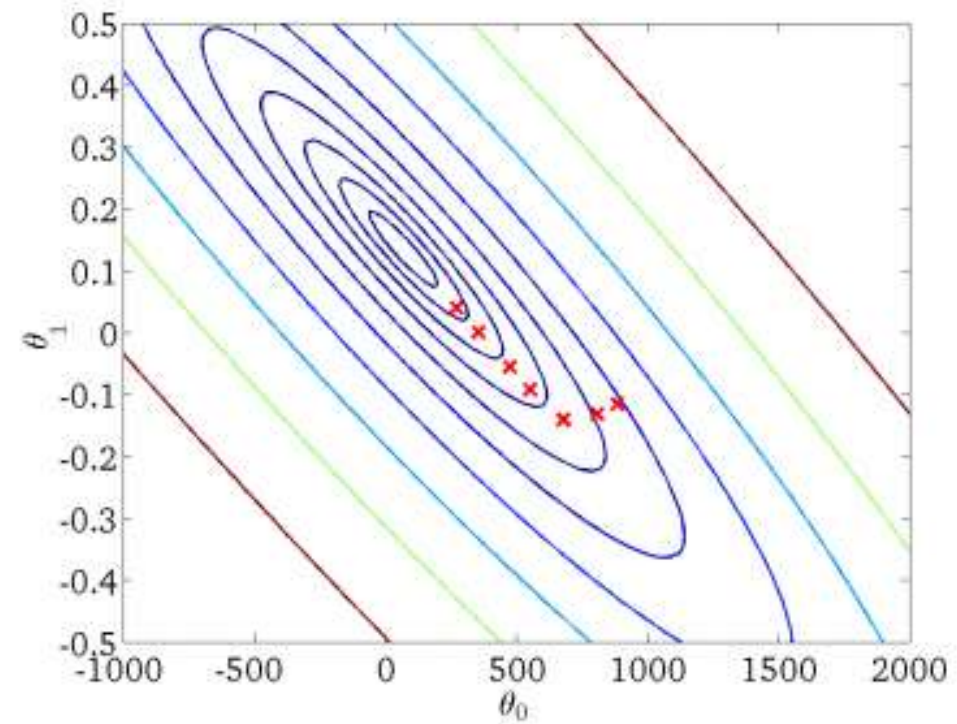
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



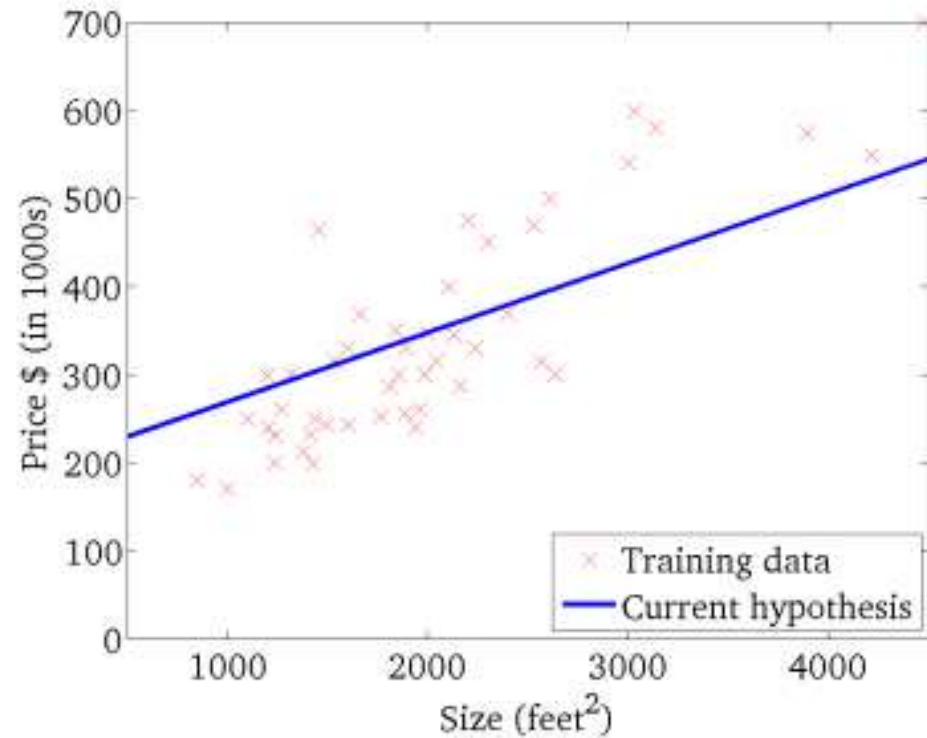
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



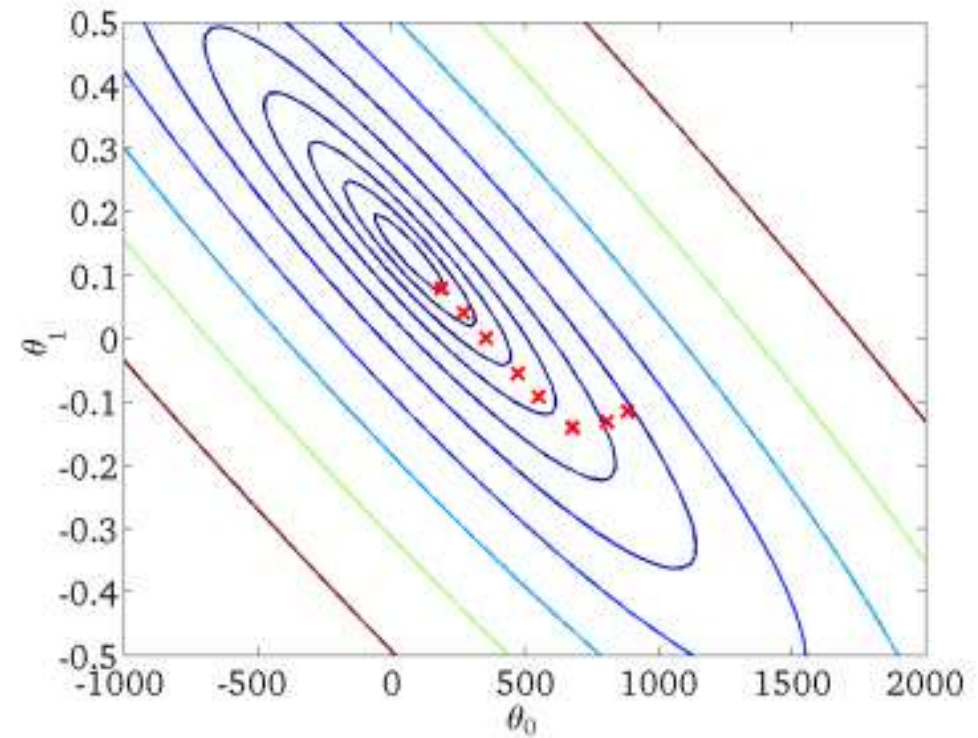
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



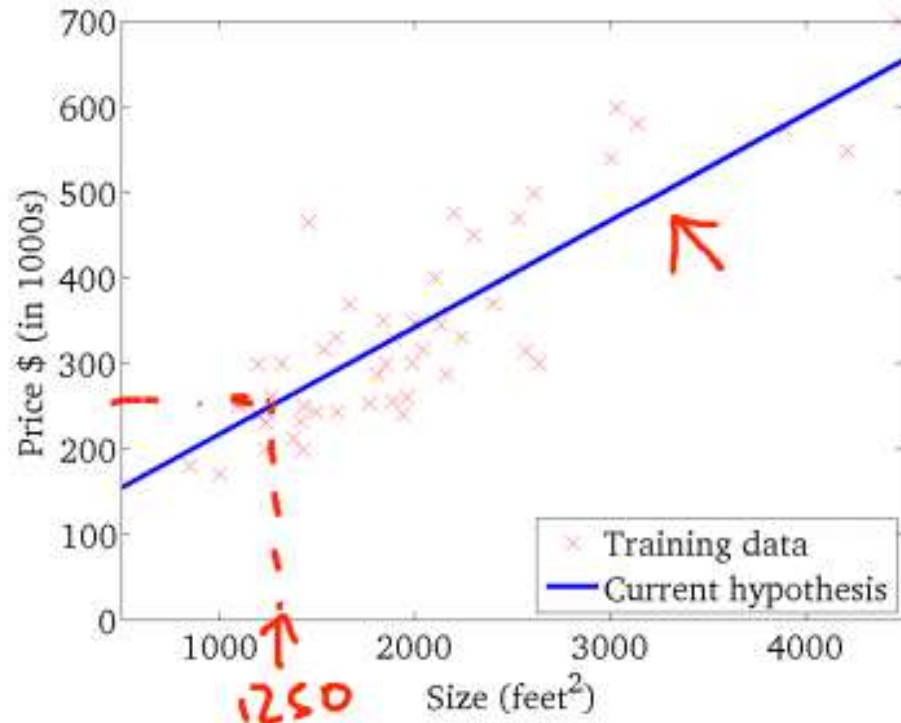
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



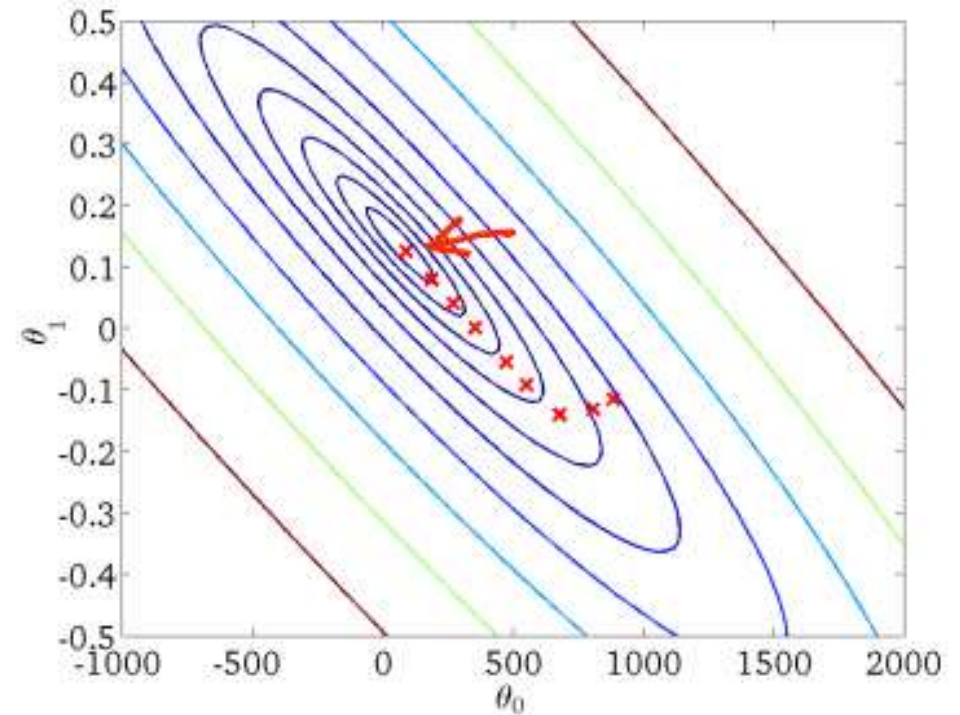
$$h_{\theta}(x)$$

(for fixed θ_0, θ_1 , this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



Best possible fit

This method looks at every example in the entire training set on every step, and is called **batch gradient descent**

Which of the following are true statements? Select all that apply.

- To make gradient descent converge, we must slowly decrease α over time.
- Gradient descent is guaranteed to find the global minimum for any function $J(\theta_0, \theta_1)$
- Gradient descent can converge even if α is kept fixed. (But α cannot be too large, or else it may fail to converge)
- For the specific choice of cost function $J(\theta_0, \theta_1)$ used in linear regression, there are no local optima (other than the global optimum)

Linear Algebra

Matrices:

Matrices are rectangular arrangement of elements.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \\ j & k & l \end{bmatrix}$$

The above matrix has four rows and three columns, so it is a 4 x 3 matrix.

Vectors:

A vector is a matrix with one column and many rows

$$\begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix}$$

The above matrix is a 4D vector.

Some Standard Notations

- A_{ij} refers to the element in the i^{th} row and j^{th} column of matrix A .
- A vector with 'n' rows is referred to as an 'n'-dimensional vector.
- v_i refers to the element in the i^{th} row of the vector.
- In general, all our vectors and matrices will be 1-indexed.
(Note that for some programming languages, the arrays are 0-indexed)
- Matrices are usually denoted by uppercase names while vectors are lowercase.
- "Scalar" means that an object is a single value, not a vector or matrix.
- R refers to the set of scalar real numbers
- R^n refers to the set of n-dimensional vectors of real numbers.

Addition and Scalar Multiplication

Addition and subtraction are **element-wise**, so you simply add or subtract each corresponding element:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} a+w & b+x \\ c+y & d+z \end{bmatrix} \qquad \begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} a-w & b-x \\ c-y & d-z \end{bmatrix}$$

To add or subtract two matrices, their dimensions must be **the same**.

In scalar multiplication, we simply multiply every element by the scalar value:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} * x = \begin{bmatrix} a*x & b*x \\ c*x & d*x \end{bmatrix}$$

In scalar division, we simply divide every element by the scalar value:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} / x = \begin{bmatrix} a/x & b/x \\ c/x & d/x \end{bmatrix}$$

Matrix-Vector Multiplication

We map the column of the vector onto each row of the matrix, multiplying each element and summing the result.

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a * x + b * y \\ c * x + d * y \\ e * x + f * y \end{bmatrix}$$

The result is a **vector**. The number of **columns** of the matrix must equal the number of **rows** of the vector.

An **m x n matrix** multiplied by an **n x 1 vector** results in an **m x 1 vector**.

Example:

Consider the following hypothesis: $h_{\theta}(x) = -40 + 0.25x$ for the given house sizes in sq. feet.

House	Size (in sq. feet)
H1	2104
H2	1416
H3	1534
H4	852

This can be represented as the product of matrix and vector.

$$\begin{bmatrix} 1 & 2104 \\ 1 & 1416 \\ 1 & 1534 \\ 1 & 852 \end{bmatrix} \begin{bmatrix} -40 \\ 0.25 \end{bmatrix} = \begin{bmatrix} -40 \times 1 + 0.25 \times 2104 \\ -40 \times 1 + 0.25 \times 1416 \\ -40 \times 1 + 0.25 \times 1534 \\ -40 \times 1 + 0.25 \times 852 \end{bmatrix}$$

Note: Prediction = Data Matrix x Parameters

Matrix-Matrix Multiplication

An **m x n matrix** multiplied by an **n x o matrix** results in an **m x o matrix**.

$$\begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} * \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} a * w + b * y & a * x + b * z \\ c * w + d * y & c * x + d * z \\ e * w + f * y & e * x + f * z \end{bmatrix}$$

To multiply two matrices, the number of **columns** of the first matrix must equal the number of **rows** of the second matrix.

Example:

Consider the following three hypothesis for given house sizes in sq. feet.

$$h_{\theta}(x) = -40 + 0.25x$$

$$h_{\theta}(x) = 200 + 0.1x$$

$$h_{\theta}(x) = -150 + 0.4x$$

House	Size (in sq. feet)
H1	2104
H2	1416
H3	1534
H4	852

This can be represented as the product of two matrices.

$$\begin{bmatrix} 1 & 2104 \\ 1 & 1416 \\ 1 & 1534 \\ 1 & 852 \end{bmatrix} \times \begin{bmatrix} -40 & 200 & -150 \\ 0.25 & 0.1 & 0.4 \end{bmatrix} = \begin{bmatrix} 486 & 410 & 692 \\ 314 & 342 & 416 \\ 344 & 353 & 464 \\ 173 & 285 & 191 \end{bmatrix}$$

Hence, with the help of only one matrix multiplication, we are able to make 12 predictions.

Properties

- Matrix multiplication is not commutative i.e. $A*B \neq B*A$
- Matrices are associative i.e. $(A*B)*C = A*(B*C)$
- An identity matrix of order 'n' is represented as I_n
- All the diagonal elements of an identity matrix are 1
- All the non-diagonal elements of an identity matrix are 0
- $A*I = I*A = A$
(Note that the orders of both the matrices, A and I must be equal)

Inverse and Transpose

- The inverse of a matrix A is denoted by A^{-1}
- $A \times A^{-1} = I$
- Inverse of only square matrices is possible
- If $|A| = 0$, then inverse of A does not exist
- Such a matrix is called “singular” or “degenerate” matrix
- The transpose of a matrix A is denoted by A^T or A'

$$A = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \longrightarrow A^T = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}$$

- $A_{ij} = A_{ji}^T$