Linear Regression

A number of slides and screenshots from : Andrew Ng's course on machine learning and Sebastian Raschka's course on deep learning
Both can be found for free on youtube

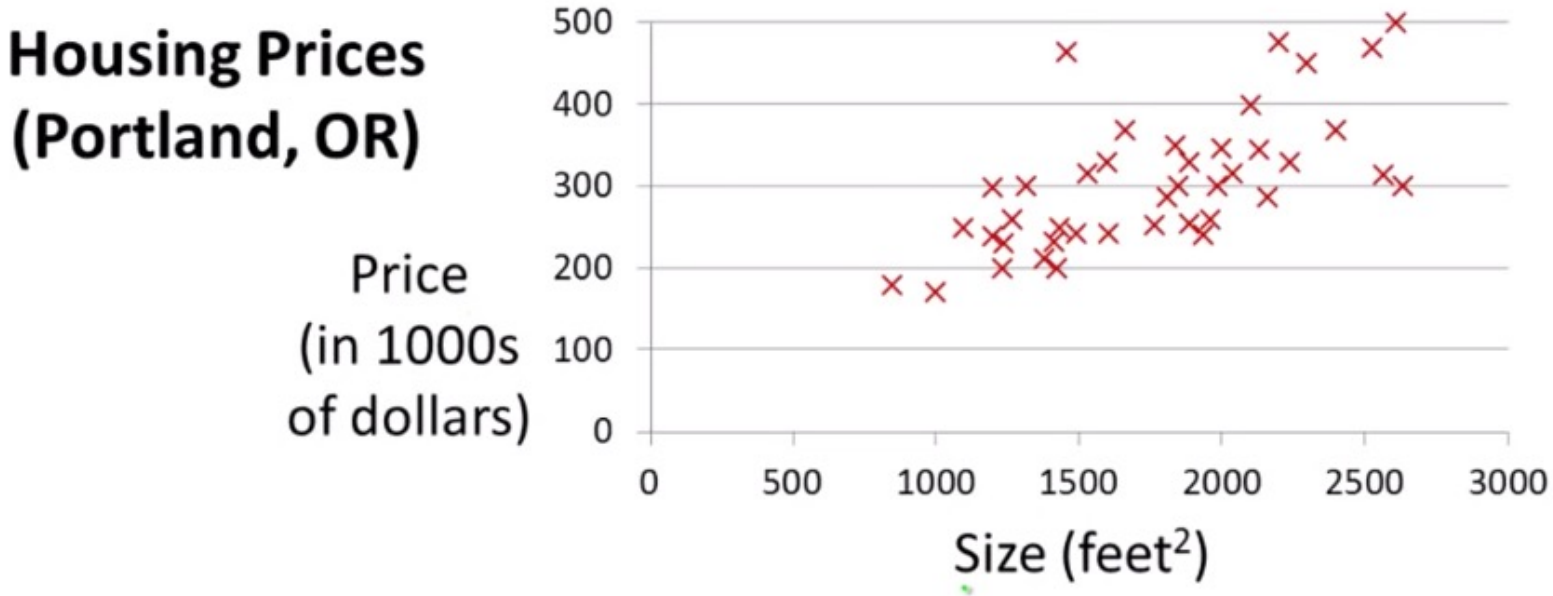
What does regression mean ?

- Seen in intro, but :
- Regression means predictiong real-valued outputs.
- An essential type of supervised machine learning task (trying to give the right « answer » for each example in the data).
- Often contrasted with classification.

- Example :
- Predicting height => many many real-valued outputs are possible…
- Vs. Predicting a « height class » : short medium-height tall

Dataset and problem example

- Imagine we want to create an ML algorithm to predict the price of a house, using only as information the size of the house. This is the dataset we can use to train our algorithm.



Training Set and Notation

Training set of housing prices (Portland, OR)

Size in feet^2 (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

Notation:

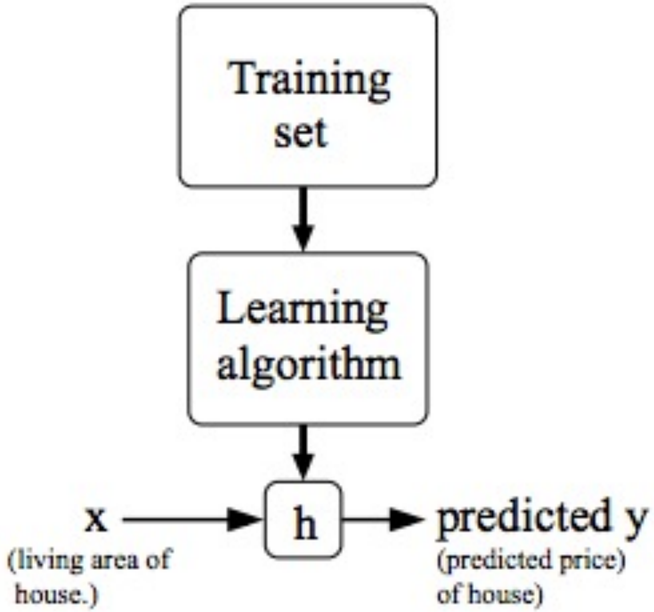
m = Number of training examples
x's = "input" variable / features
y's = "output" variable / "target" variable

The supervised learning workflow

- h: hypothesis
- h is a function which maps x's to y's
- Our goal will be to find the function which takes

x as input and predicts the correct y for that

x.

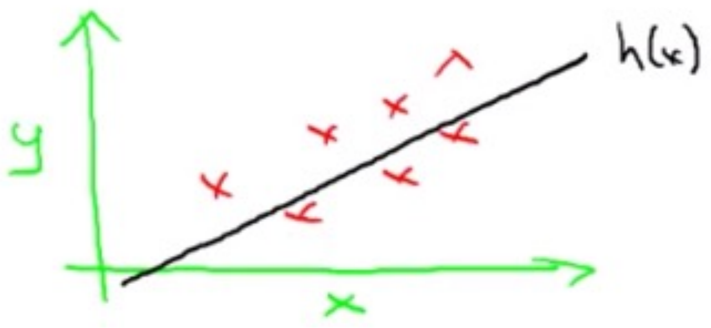


Model h

- To start with, we will use a simple model, a function which is the equation of a line (maybe you remember y = ax + b from school ?)

$$h(x) = \theta_0 + \theta_1 x$$

- This model will predict that y is some straight line function :



If this seems a bit odd to you…

- Remember we want our function to predict the examples we have in our training set correctly,
- which our simple model will probably not do very well….
- What if we can’t get to all the points using a straight line ?
- Don’t worry for now, this is still a good starting point !

Cost function

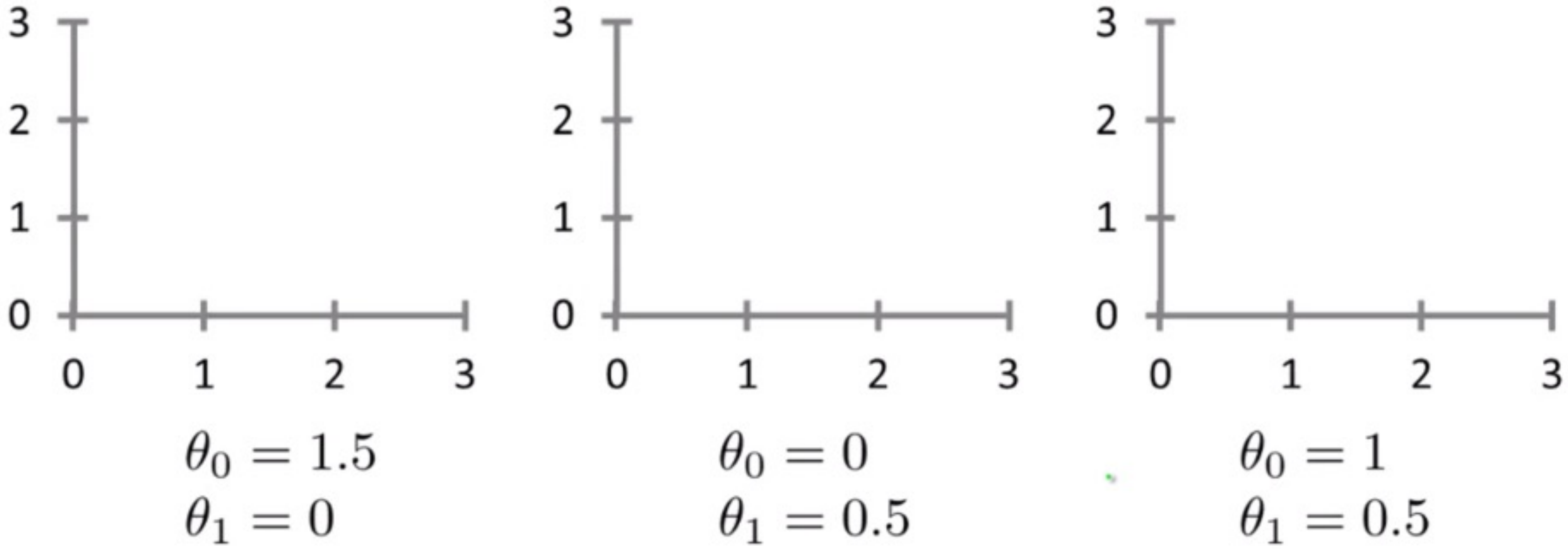
- This is a second function we will use to judge how well our straight line is fitting the data and to find the best possible straight line.

- $h(x) = \theta_0 + \theta_1 x$

- $\theta_{i's}$ are what we call **parameters** and we want to find the right combination of those parameters to get the best line.
- So how do we choose the right parameters ?

Different parameter choices/hypotheses

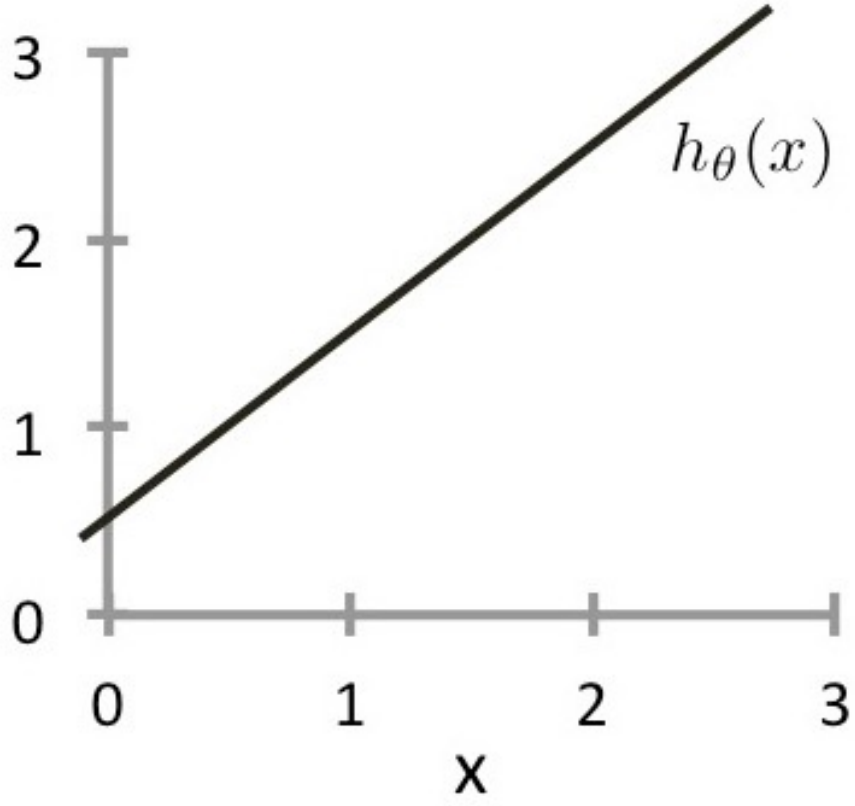
$$h_\theta(x) = \theta_0 + \theta_1 x$$



Exercise

- Look at the plot of $h(x) = \theta_0 + \theta_1 x$

- What are the values of θ_0 and θ_1 ?



Minimization Problem

- We want to choose θ_0 and θ_1 so that
- $h(x)$is close to y for out training examples (x, y)…
- So this is actually a **minimization problem**,
- where we want to minimize $(h(x) - y)^2$ by tweaking our parameters θ_0 and θ_1

Cost function = Quantifying the model's error

- The previous slide only took into account the error for a single example…
- So for all of our examples m the average error is :

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

The 2 is just there to make the math easier but doesn't change anything fundamentally, you can regard this as the average error.

- This function is known as the MSE (we'll see how it works in a few slides) and is the most commonly used:

Mean Squared Error

To recap

Hypothesis:

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

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

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

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$

https://www.coursera.org/learn/machine-learning/home/welcome

Cost Function Intuition

- Let's use a simplified model hypothesis to understand what's going on:

$$h(x) = \theta_1 x$$

- Our objective is now to minimize

$$J(\theta_1)$$

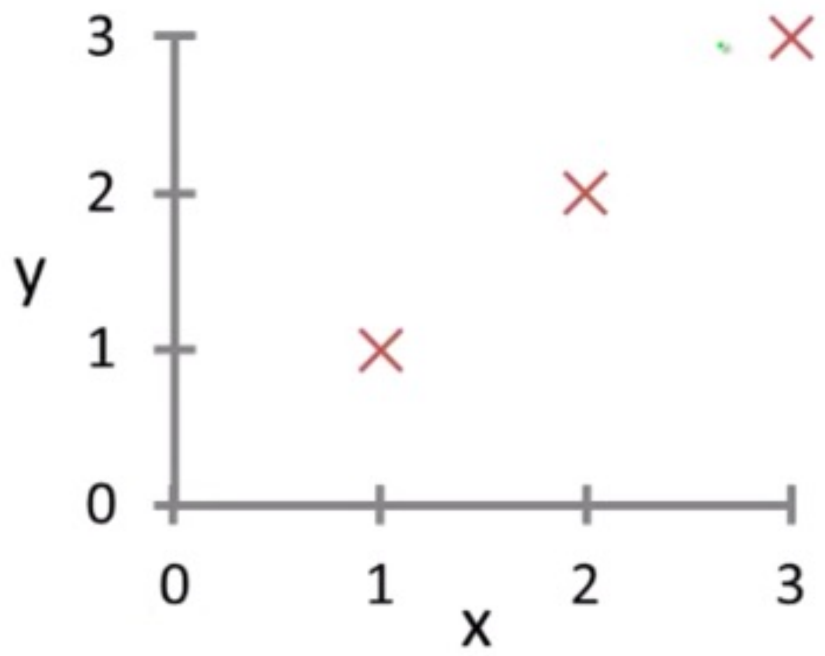
- And our cost function looks like

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

Hypothesis function vs. Cost function

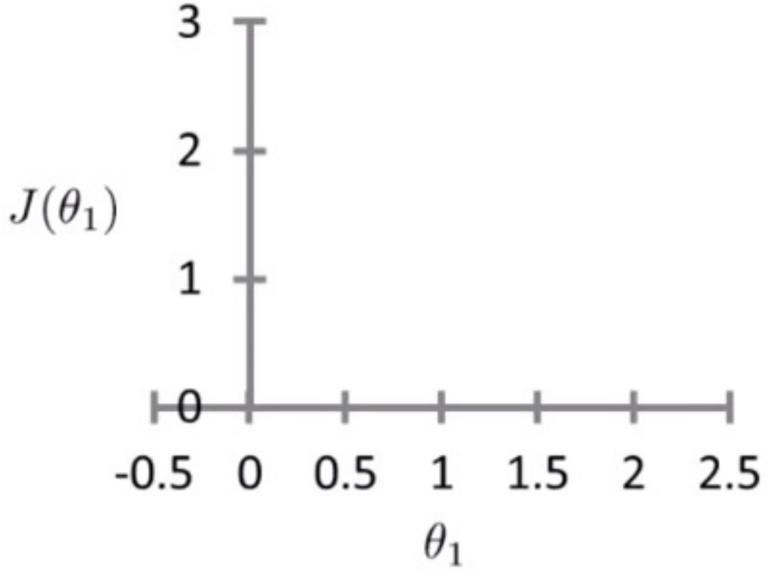
- If the points below represent our training data and $\theta = 1$, what does our hypothesis (line) look like ?
- What is the cost ? Let's find out !

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



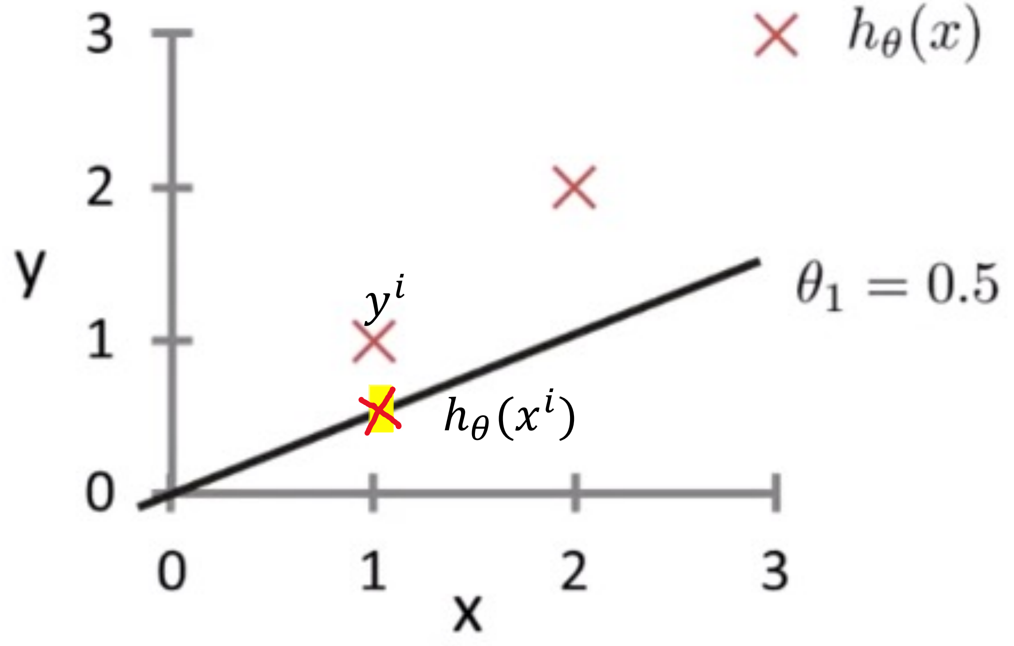
Hypothesis function vs. Cost function

- $J(\theta_1 = 1) = 0$
- We can now plot our error rate
- Notice that the values for θ_1 are on the horizontal axis. This is not the same graph as before !!



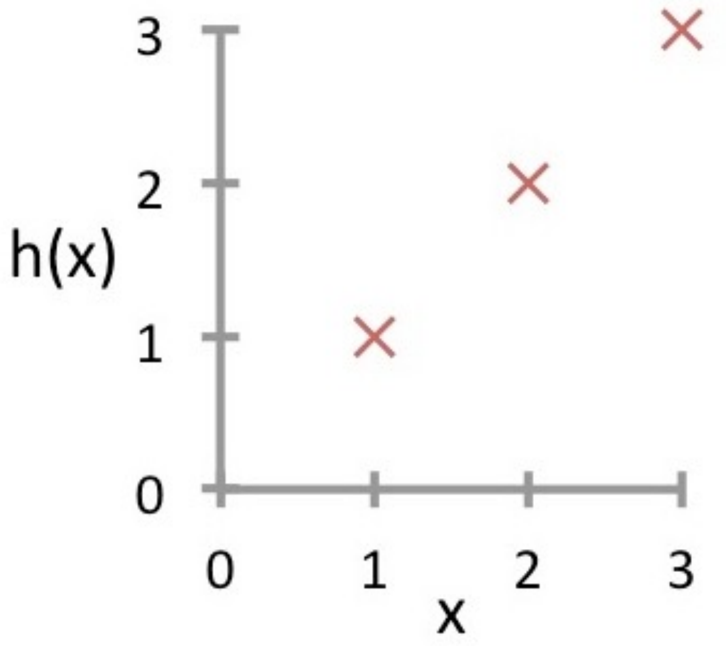
Hypothesis function vs. Cost function

- Now let's look at $\theta_1 = 0.5$
- And compute $J(\theta_1 = 0.5)$ (approx. 0.58)
- The error for each point is actually the height wich seperates the data point and the line for a given x.



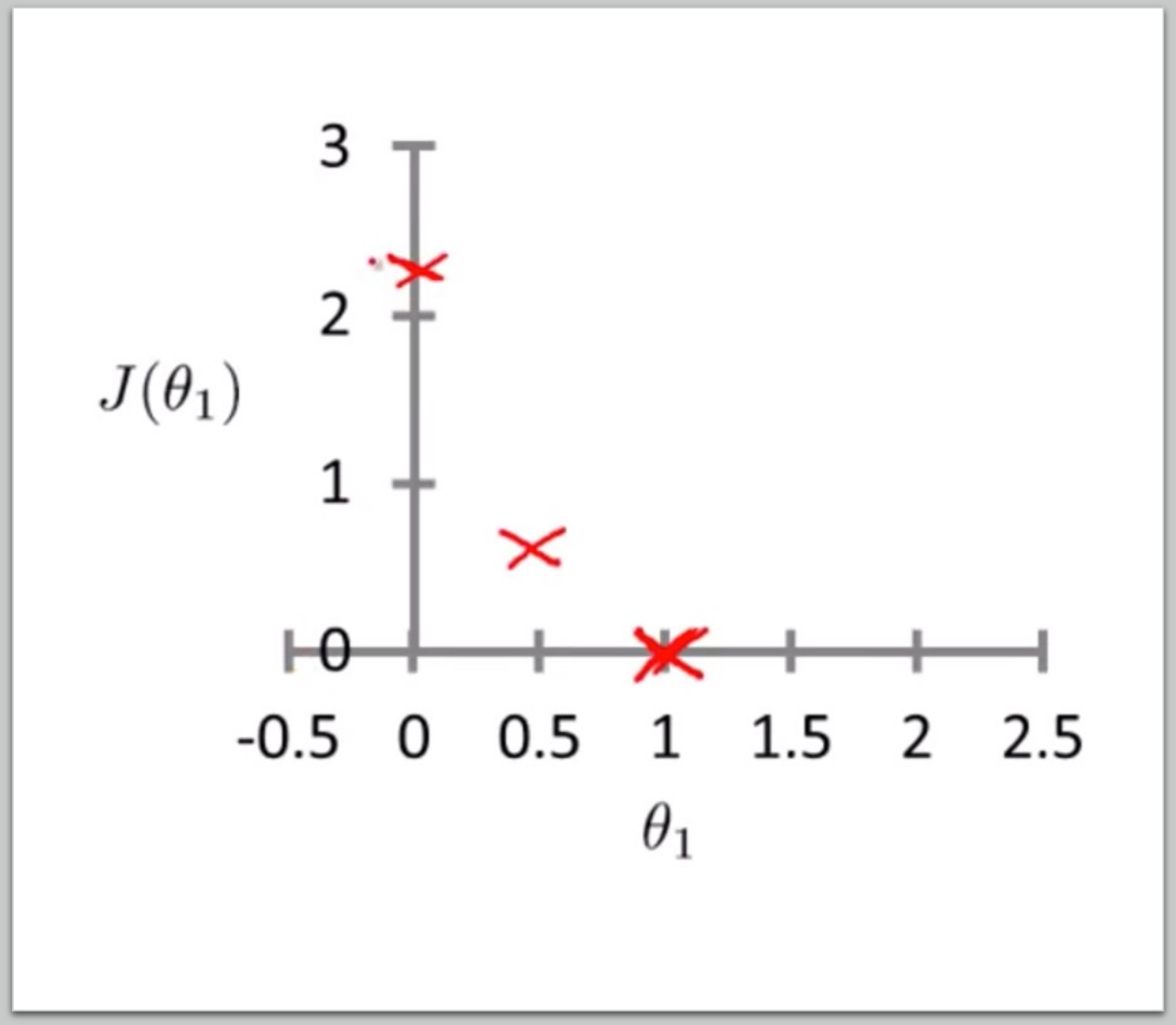
Your turn !

- Suppose this is our training set. $m = 3$.
- Given the same hypothesis and cost
- functions as before, what is $J(0)$?
- ie. $\theta_1 = 0$
- Should be approx. 2.3



Hypothesis function vs. Cost function

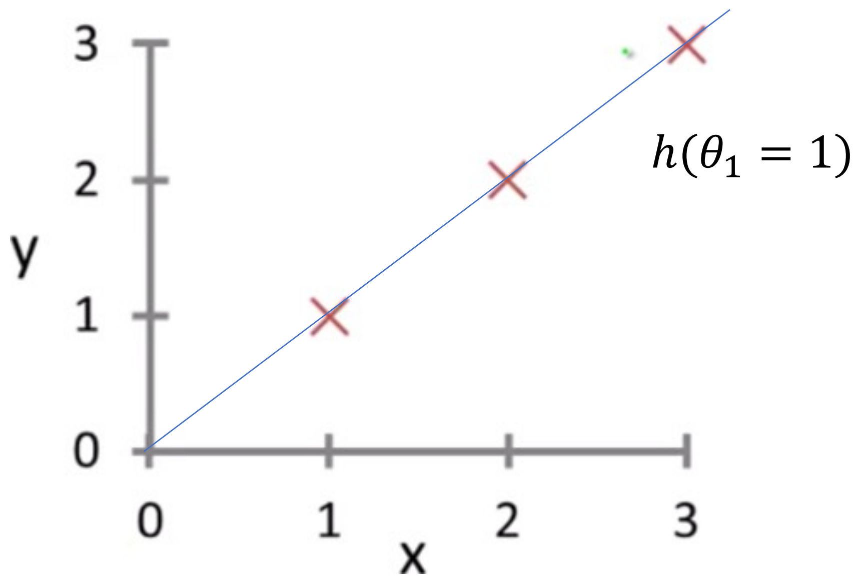
- We could continue plotting points but we'll stop here.
- With the error calculated for the different values of θ_1, we start to see part of the general shape of the function
- It turns out the function is convex/looks like a parabola.

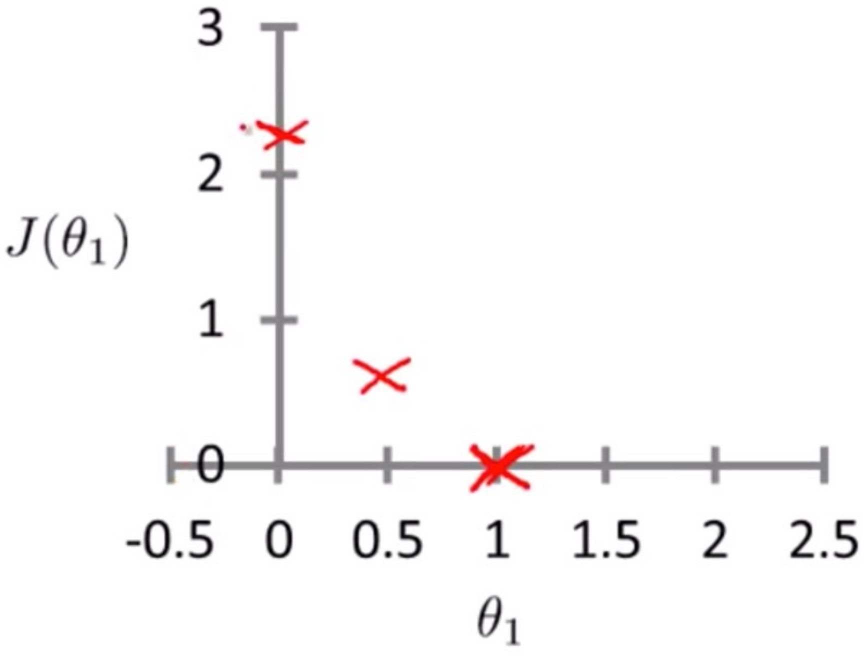


Quick recap

- Each value of θ_1 plotted corresponds to a different hypothesis / model / straight line on the data point graphs shown previously.
- For each value we can compute a value $J(\theta_1)$ to trace out the cost function.
- Now remember, we wanted to find the value of θ_1 which minimized $J(\theta_1)$… Looking at the graph we can now do so !

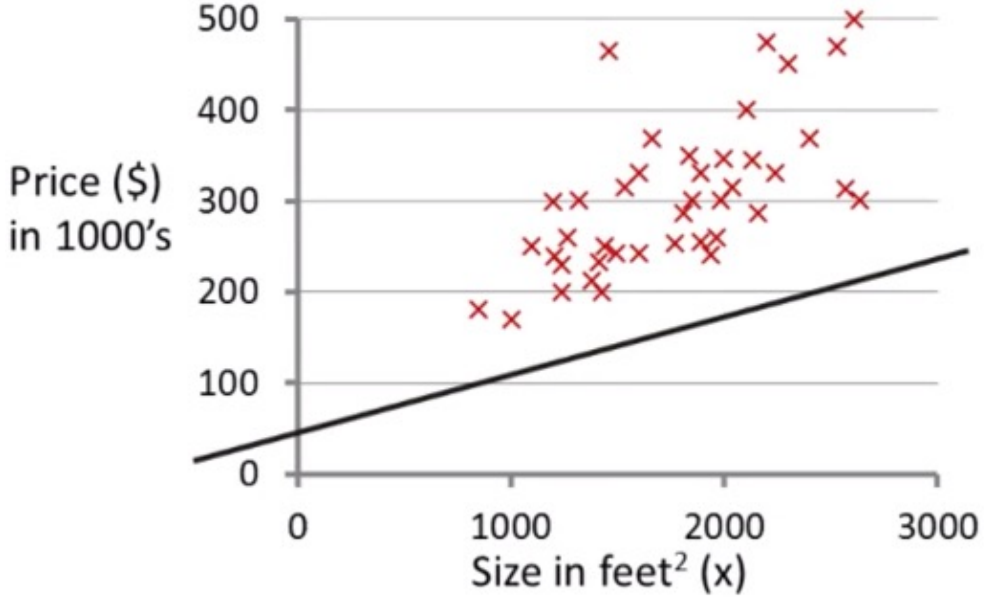
- No surprise, the value of θ_1 which minimizes the error, is associated with the model which fits the data perfectly





Back to 2 parameters

- Now we use our original, 2 parameter hypothesis to draw our line.
- For :
- $\theta_0 = 50$
- $\theta_1 = 0.06$
- We get this straight line as our model



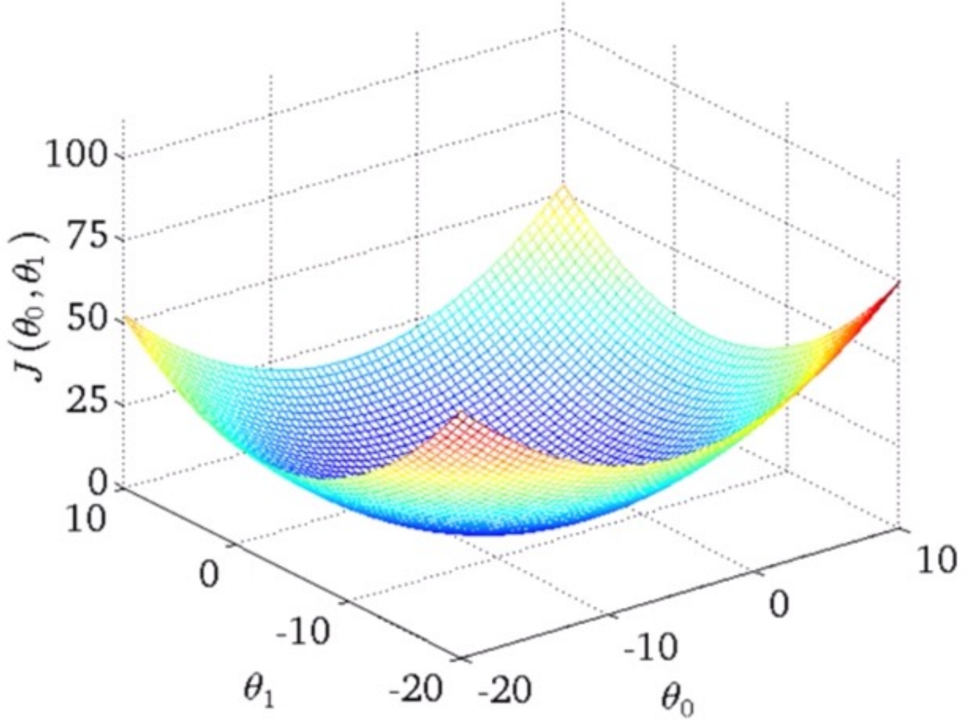
$$h_\theta(x) = 50 + 0.06x$$

Corresponding Cost function

- Now we have two parameters, the error graph will be slightly harder to plot as it has 3 dimensions:

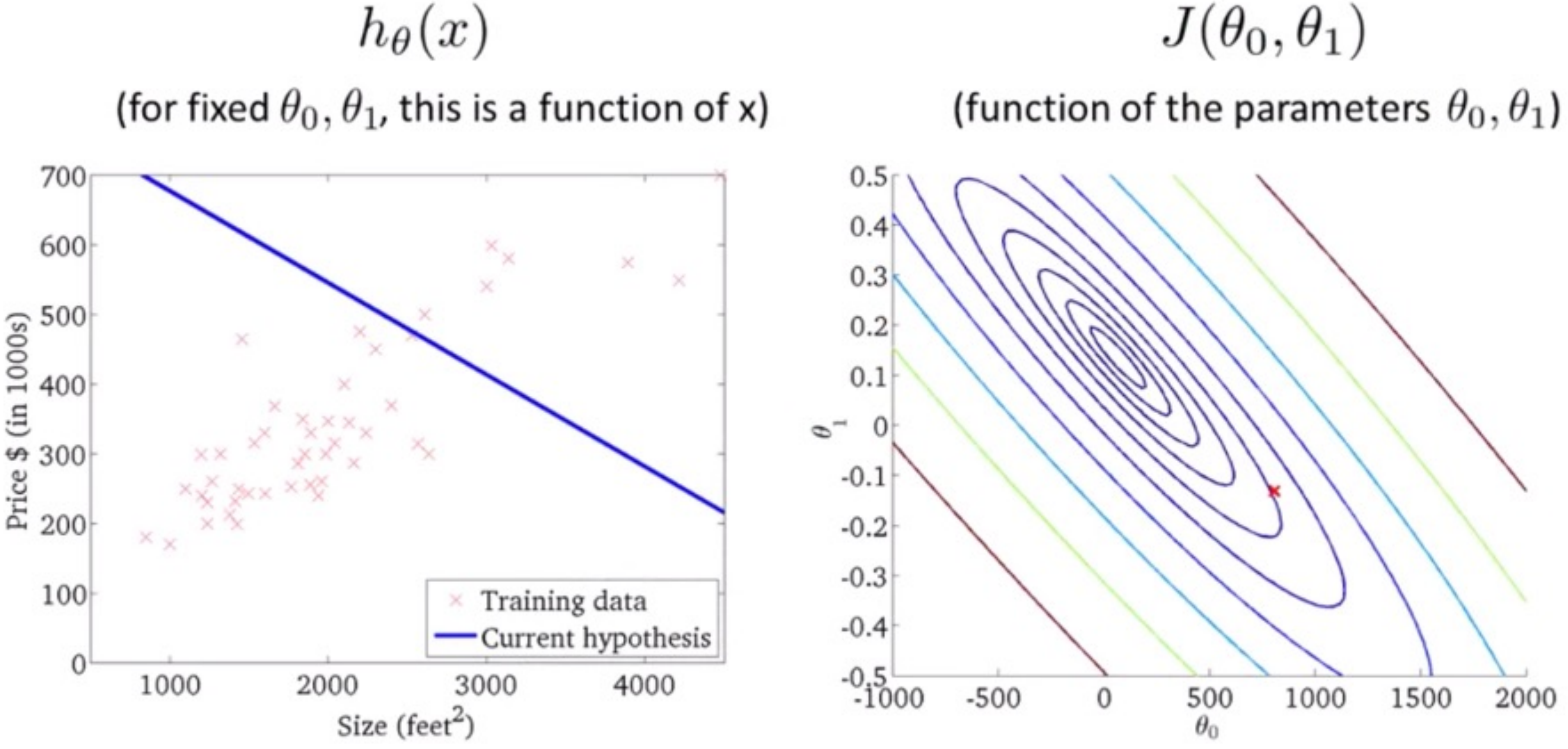
$$\theta_1, \theta_2, cost$$

- Indeed , $J(\theta_1, \theta_2)$ now has 2 inputs,
- So it will like this in 3D:



Contour Plots

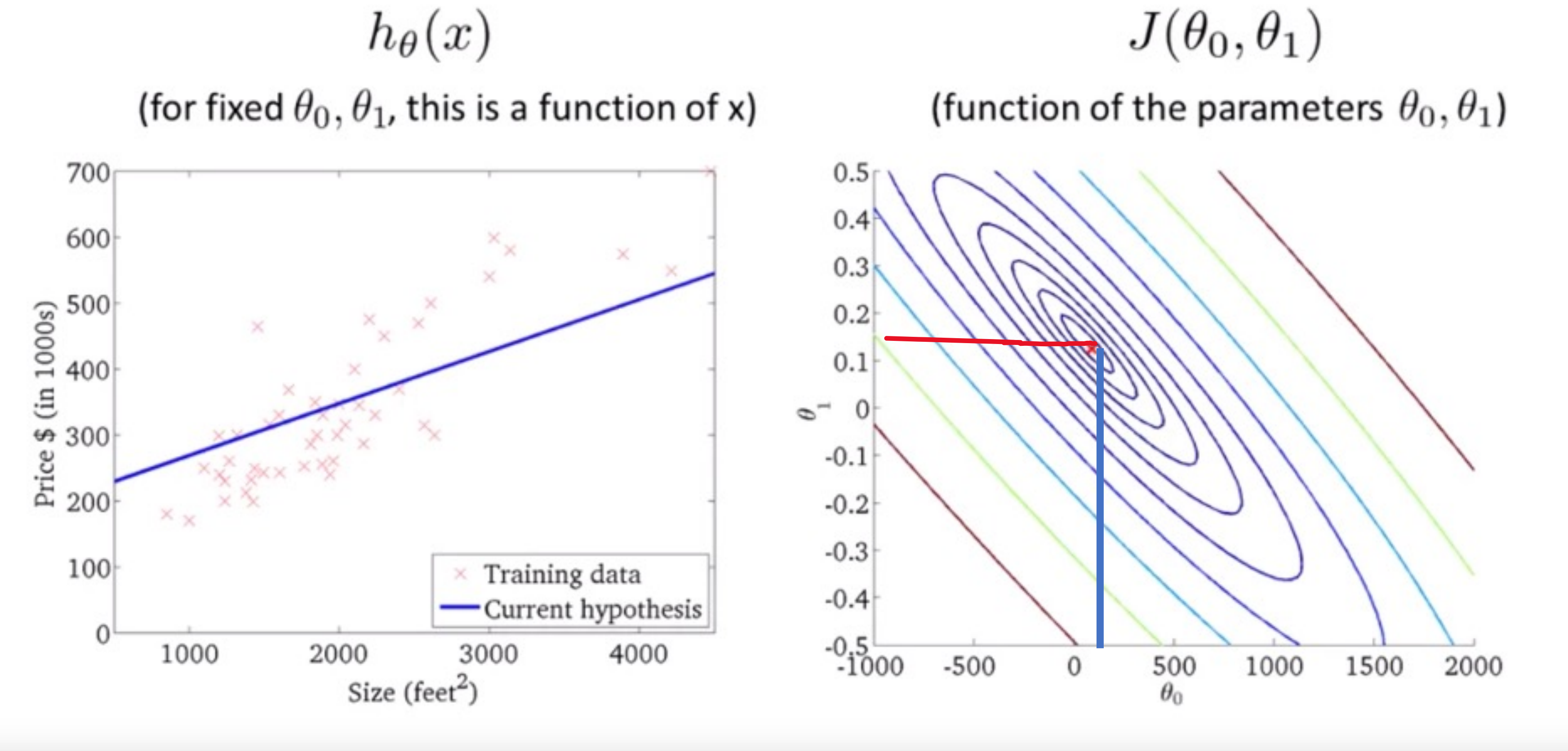
- To stay in 2D, you will see the cost function represented by a contour plot :



The ovals/ellipses show the set of points which take on the same value for given values of θ_0, θ_1

Countour Plots

- The minimum is at the center of all the « ellipses ».
- This shows a model very close to the minimum.



Gradient Descent

- Now we know how to evaluate a model, using a cost function, how do we make the model *learn* the optimal parameters ?
- In other words, how do we minimize the cost function without testing all the different possible models ?

- The algorithm used to do this is called *Gradient Descent*, and is essential to most machine learning algorithms, not just linear regression !

Gradient Descent

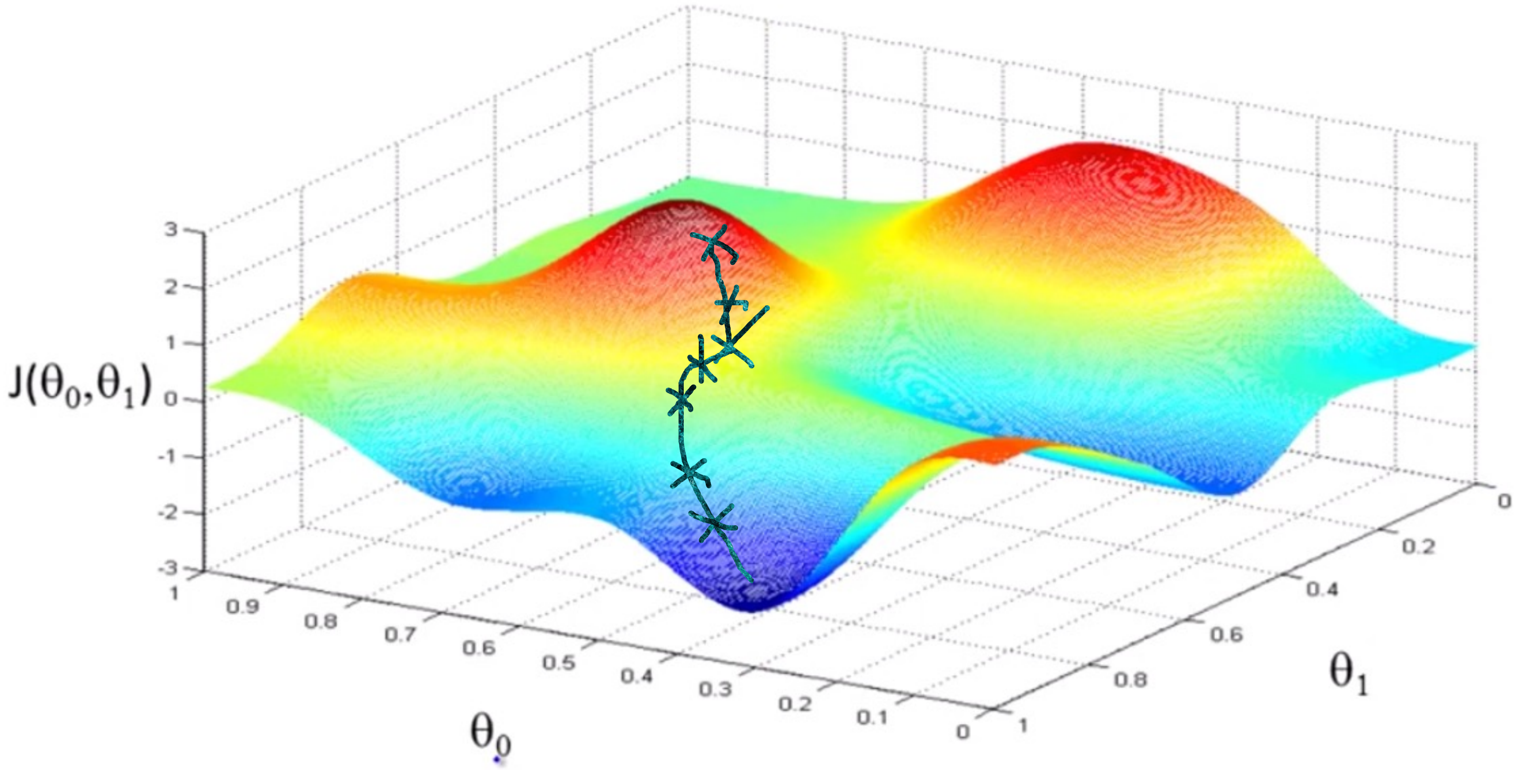
- We have some function $J(\theta_1, \theta_2)$
- Which we want to minimize…

- Outline :

  - Start with some inital guess, some random values for θ_1, θ_2
  - Keep updating θ_1, θ_2 a little bit to reduce $J(\theta_1, \theta_2)$ until we hopefully end up at a minimum

GD intuition

- This is your cost function in 3D
- Imagine you start somewhere near the top of one of the « hills » and your goal is to walk in the direction which will take you down the fastest.



GD formula

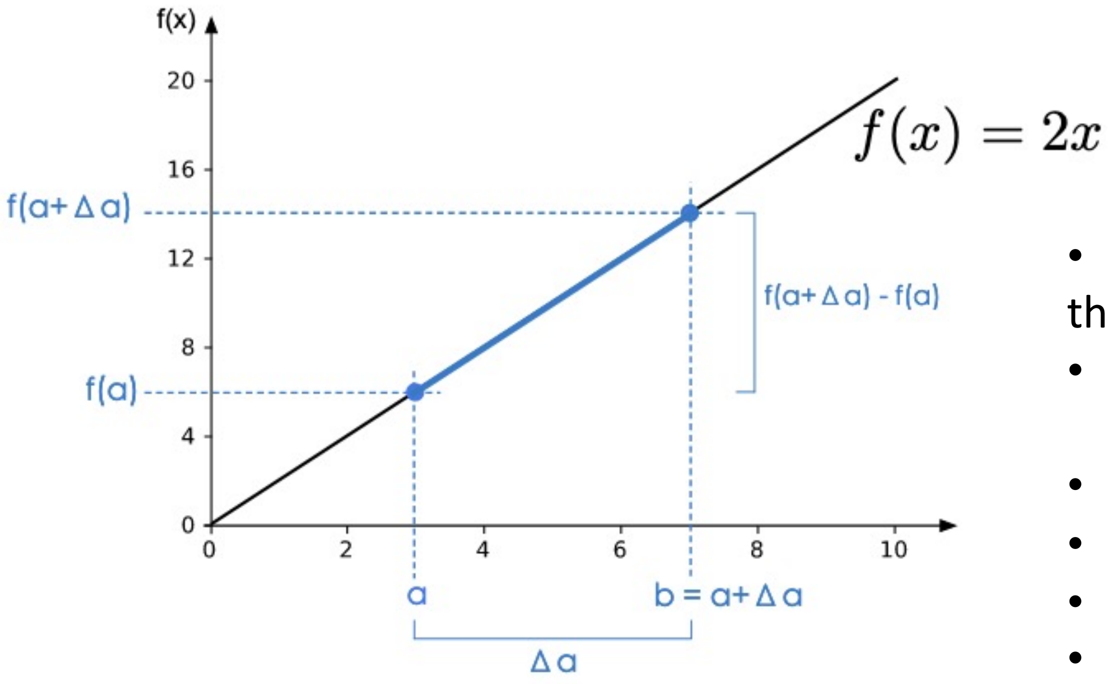
$$\text{repeat until convergence } \{$$
$$\theta_j := \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)} \quad (\text{for } j = 0 \text{ and } j = 1)$$
$$\}$$

- This is the update formula for each of the parameters
- := signifies assignment
- α is a number called the *learning rate.* If α is very large, then it corresponds to an aggressive learning procedure and big steps being taken « downhill » and vice versa.
- $\boxed{\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)}$ is a derivative term, for which we need to do a bit of calculus !

Calculus Refresher : Derivatives

- The derivative describes how the output of a function varies with regard to a tiny tiny tiny variation in input.
- To start, let's first look at a not so tiny change in input :

Derivative of a function = "rate of change" = "slope"



$$\text{Slope} = \frac{f(a+\Delta a) - f(a)}{a + \Delta a - a} = \frac{f(a+\Delta a) - f(a)}{\Delta a}$$

- Go through the calculation of the slope.
- Slope is equal to 2

- This means any change in input by
- Δx will result in a change in output
- Of 2 times Δx
- AKA : if we change the input by 1 unit,
- the output changes b 2 units

- $f(x + \Delta x) = 2x + 2\Delta x$
- $f(3 + 4) = 6 + 8 = 14$

See here for the original explanation

Calculus Refresher : Derivatives

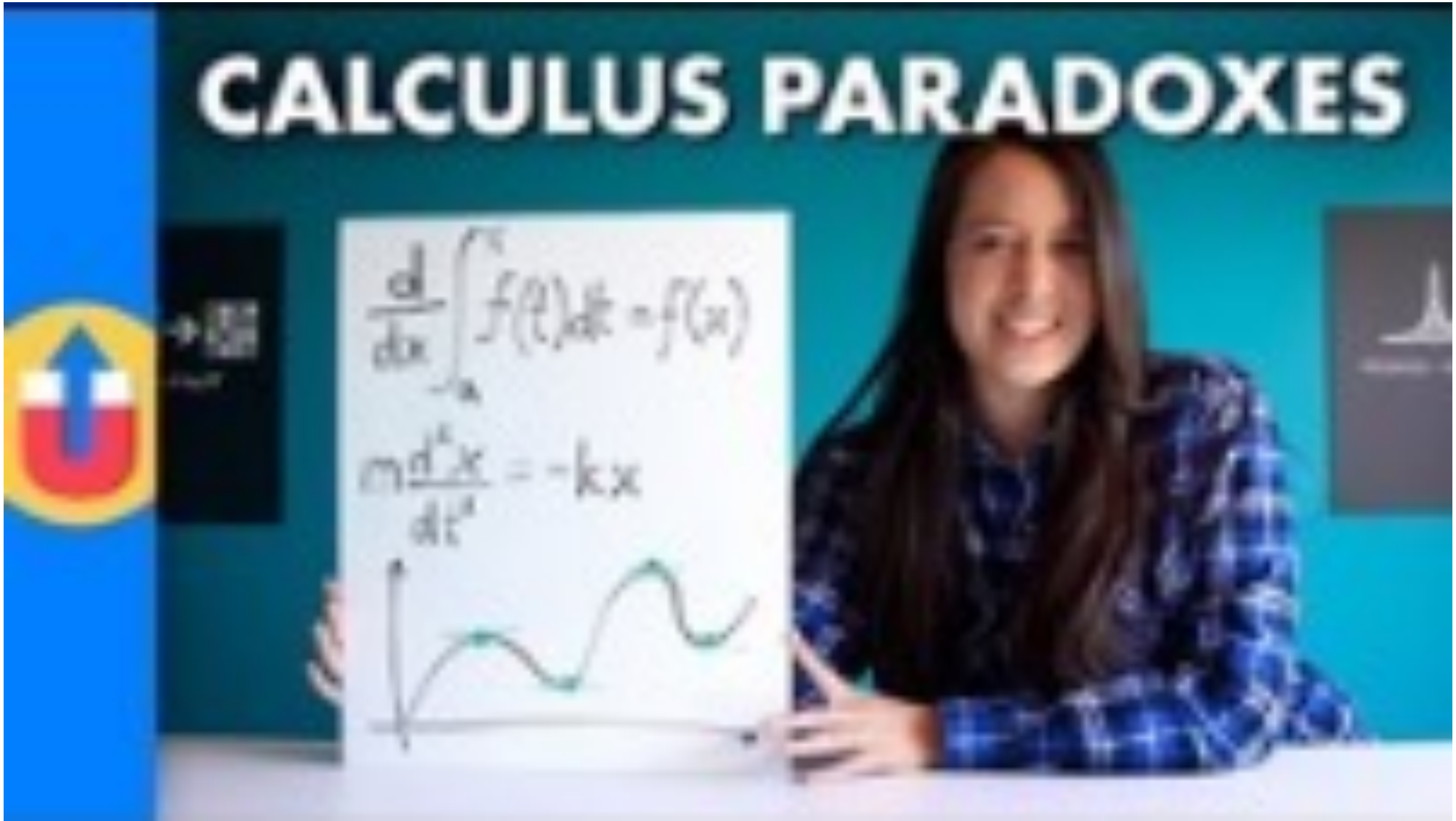
- But what happens as Δx becomes very tiny (ie. very very close to 0) ?

- This is referred to the « instantaneous rate of change ». In other words, how if we were to freeze time how fast would the car be traveling for example…?

- This notion is quite paradoxical…

Derivatives : Paradox

- Zeno's Nerf Gun (8:46)



Derivatives : notation and using the limit

- How does a tiny change in x affect the output ?
- Or, paradoxically, how is the output changing at a specific « instant »
- x ?

$$f'(x) = \frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

Lagrange notation

Leibniz Notation

<u>Example 1:</u> $f(x) = 2x$

$$\begin{aligned}
\frac{df}{dx} &= \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \\
&= \lim_{\Delta x \to 0} \frac{2x + 2\Delta x - 2x}{\Delta x} \\
&= \lim_{\Delta x \to 0} \frac{2\Delta x}{\Delta x} \\
&= \lim_{\Delta x \to 0} 2.
\end{aligned}$$

Derivatives: a more complicated example

$$f'(x) = \frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

<u>**Example 2:**</u> $f(x) = x^2$

$$\begin{aligned} \frac{df}{dx} &= \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \\ &= \lim_{\Delta x \to 0} \frac{x^2 + 2x\Delta x + (\Delta x)^2 - x^2}{\Delta x} \\ &= \lim_{\Delta x \to 0} \frac{2x\Delta x + (\Delta x)^2}{\Delta x} \\ &= \lim_{\Delta x \to 0} 2x + \Delta x. \end{aligned}$$

- So if we change the input by 1 unit (Δx = 1),
- The ouput changes by 2x +1 units

- $x = 2$
- $f(x+1) = f(2) + 2x + 1$
  $= 4 + 4 + 1 = 9$

- An this remains true as Δx approaches 0,
- Instead of being equal to 1.

- In fact, as Δx approaches 0, the derivative
- Approaches 2x.

- See videos:
  - https://www.youtube.com/watch?v=owI7zxCqNY0 (simple linear regreesion)
  - https://www.youtube.com/watch?v=HoqXask9cN8
  - https://www.youtube.com/playlist?list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF
  - https://www.youtube.com/watch?v=TSFMepJbHa0 (polynomial regression)