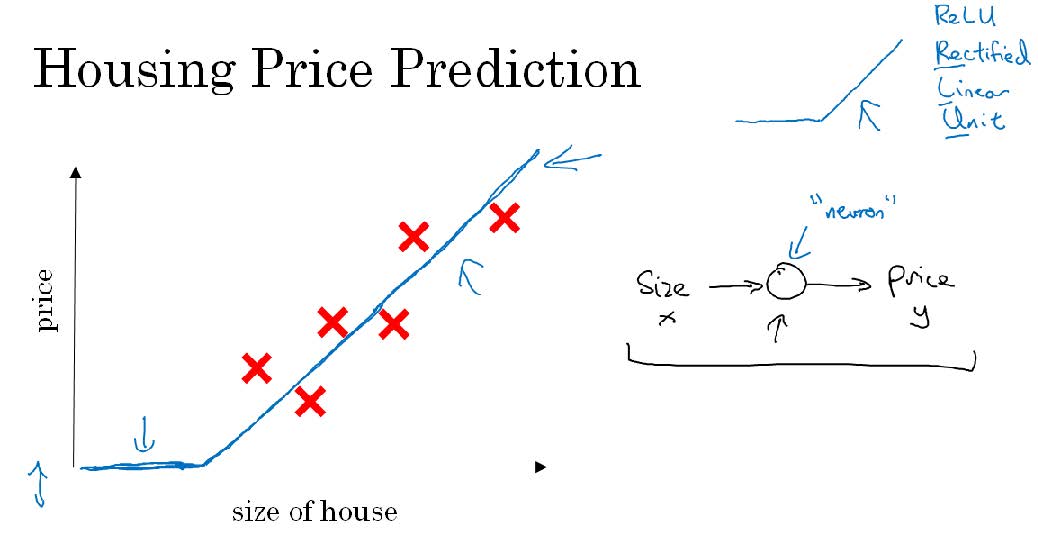
# What is NN?

* It is a powerful learning algorithm inspired by how the brain works
* Why deep learning is taking off now this fast?



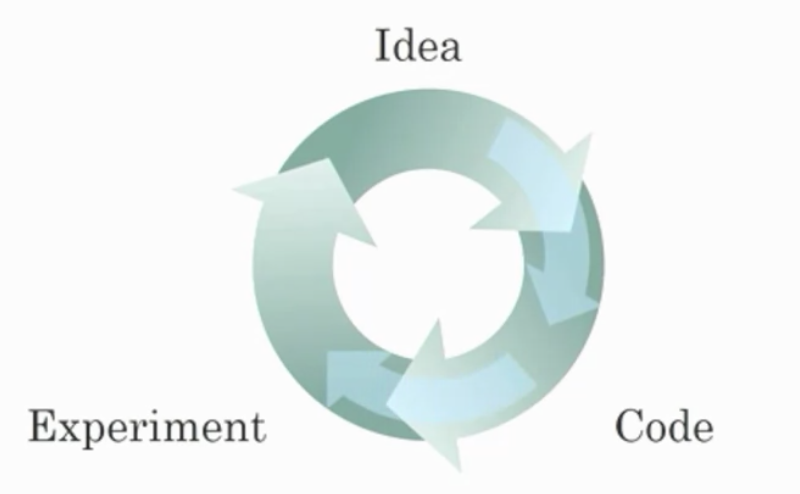
# Supervised Learning

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**.

# Structured vs unstructured data

* Structured data refers to things that has a **defined meaning** such as price, age whereas unstructured data refers to thing like pixel, raw audio, text.

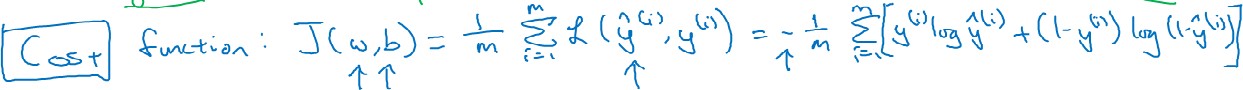
# Going about any project



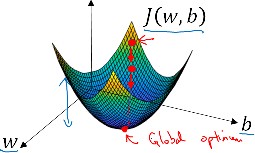
# Logistic Regression

* Logistic regression is a learning algorithm used in a supervised learning problem when the output 𝑦 are all **either zero or one**.
* The goal of logistic regression is to **minimize** the error between its predictions and training data.
* The **parameters** used in Logistic regression are:
  + The input features vector
  + The training label
  + The weights
  + The threshold
  + The output
  + Sigmoid function
* Some observations from the graph:
  + If 𝑧 is a large positive number, then 𝜎(𝑧) = 1
  + If 𝑧 is small or large negative number, then 𝜎(𝑧) = -10
  + If 𝑧 = 0, then 𝜎(𝑧) = 0.5
* X is a nx x M dim matrix

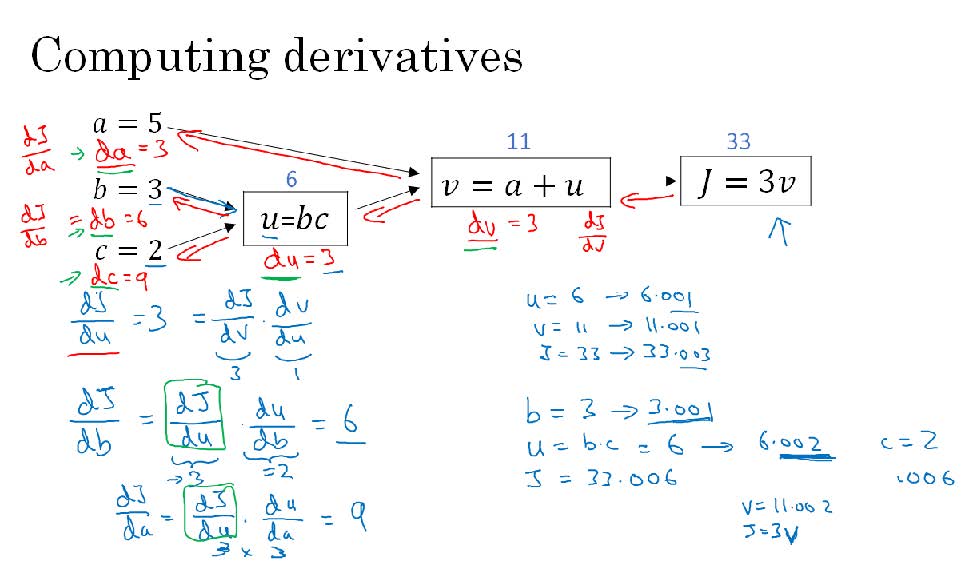
# Loss function v/s cost function

* **Loss (error) function:** The loss function measures the discrepancy between the prediction (𝑦̂ (𝑖) ) and the desired output (𝑦 (𝑖) ). In other words, the loss function computes the error for a single training example. 
* **Cost function:** The cost function is the average of the loss function of the entire training set. We are going to find the parameters 𝑤 𝑎𝑛𝑑 𝑏 that minimize the overall cost function.

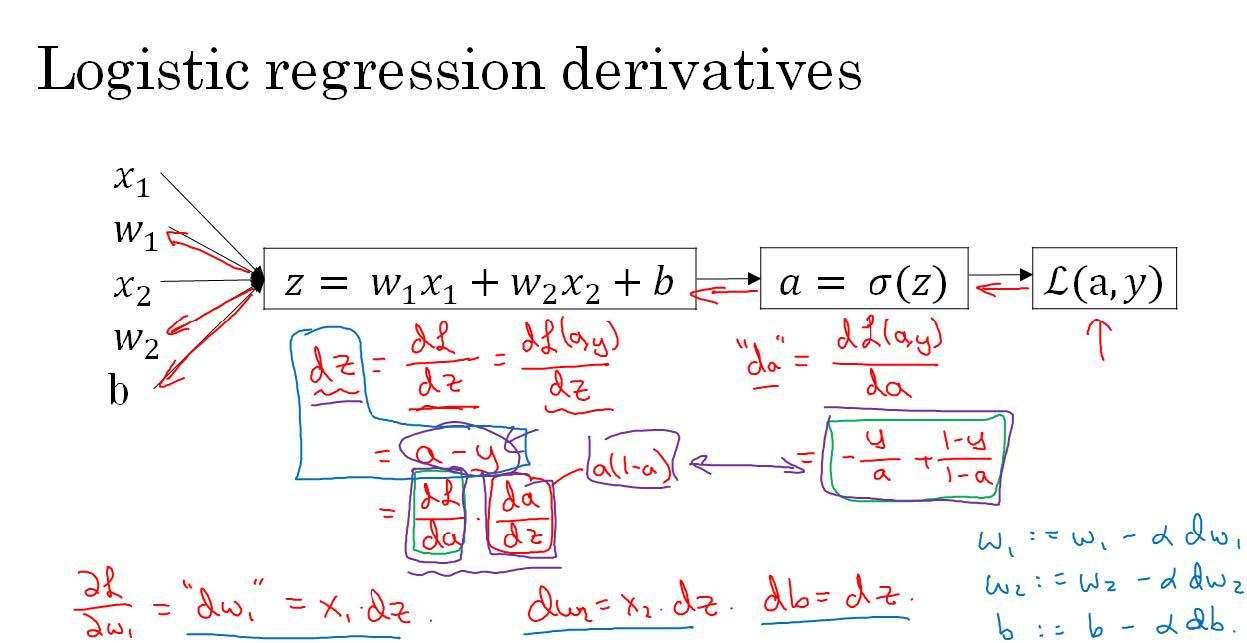
# Gradient Descent

* Gradient descent algorithm is used to **train** the parameters w and b. While training we want to find w and b that minimize cost function J(w,b).
* Generally w and b are single row no values plotted in x-axis, J is a convex function a single big bow. To find the good parameters value for w and b
* What we'll do is to initialize w and b to some initial value, we can initialize to zero. **Random initialization can also work but people usually don't** so thatfor logistic regression so what the gradient descent does is to start at this initial point and take a step in the steepest downhill direction or say descent as quickly as possible. This is a one iteration of the gradient descent.
* We can go on with the multiple iteration till it converses to the global optima / close to global optima.
* 
* Repeat { w := w - alpha dJ(w)/dw }
* Alpha here is the learning rate and controls how bigger step we take on each iteration of gradient descent.
* dJ(w)/dw is the derivative (slope of a function at a point), the update of the change you want to make to the parameter w.

# Computation Graph



# Logistic Regression Derivative



NB: Logistic Regression Vectorization (see pdf)

# Why this choice is made?

