Table of Contents

**Type chapter title (level 1)1**

Type chapter title (level 2)2

Type chapter title (level 3)3

**Logistic Regression as a Neural Network4**

Binary Classification5

Logistic Regression5

Logistic Regression Cost Function6

Gradient6

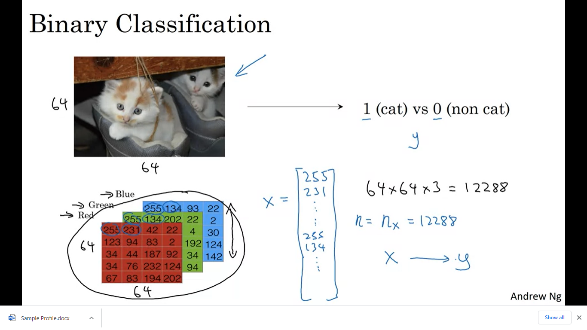
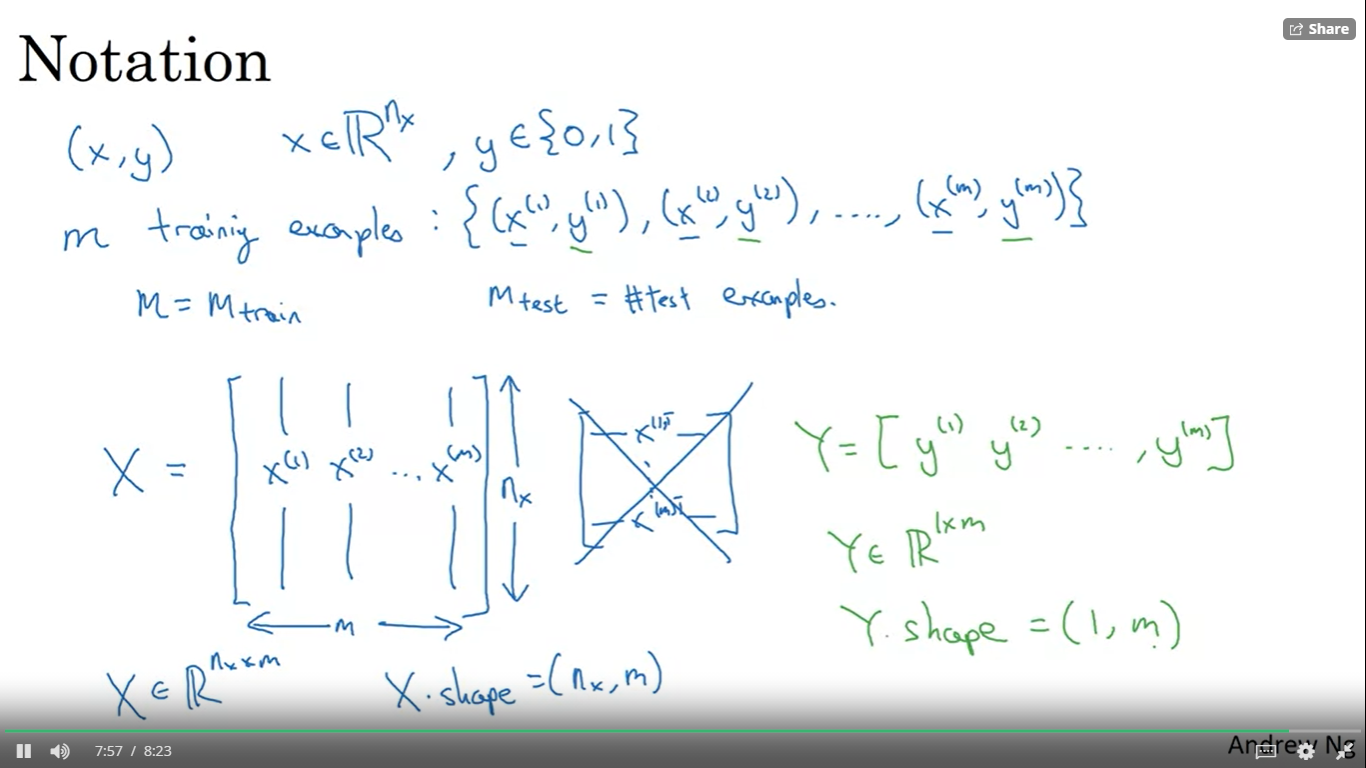
Computation Graph6

Logistic Regression Gradient Descent6

# Logistic Regression as a Neural Network

## Binary Classification:

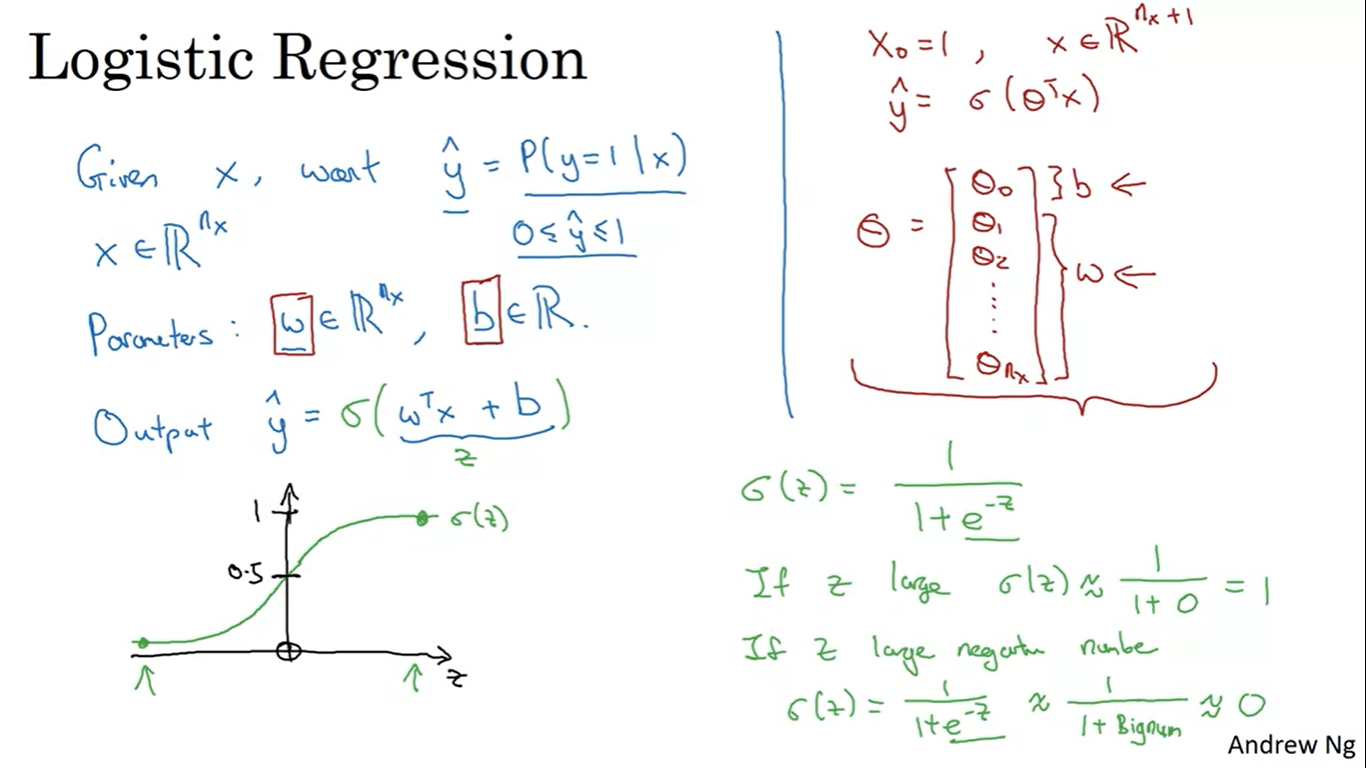
For a given image, classify if it is a cat or non-cat. If the image is 64\*64 pixel, then each image is represented in three colours Red, Green and Blue. Each colour is numerically represented by a 64\*64 matrix (Here we only see a very small representation of 5\*4 of actual value). All the three matrices corresponding to RGB will be converted into a input vector X, which when feed



## Logistic Regression:

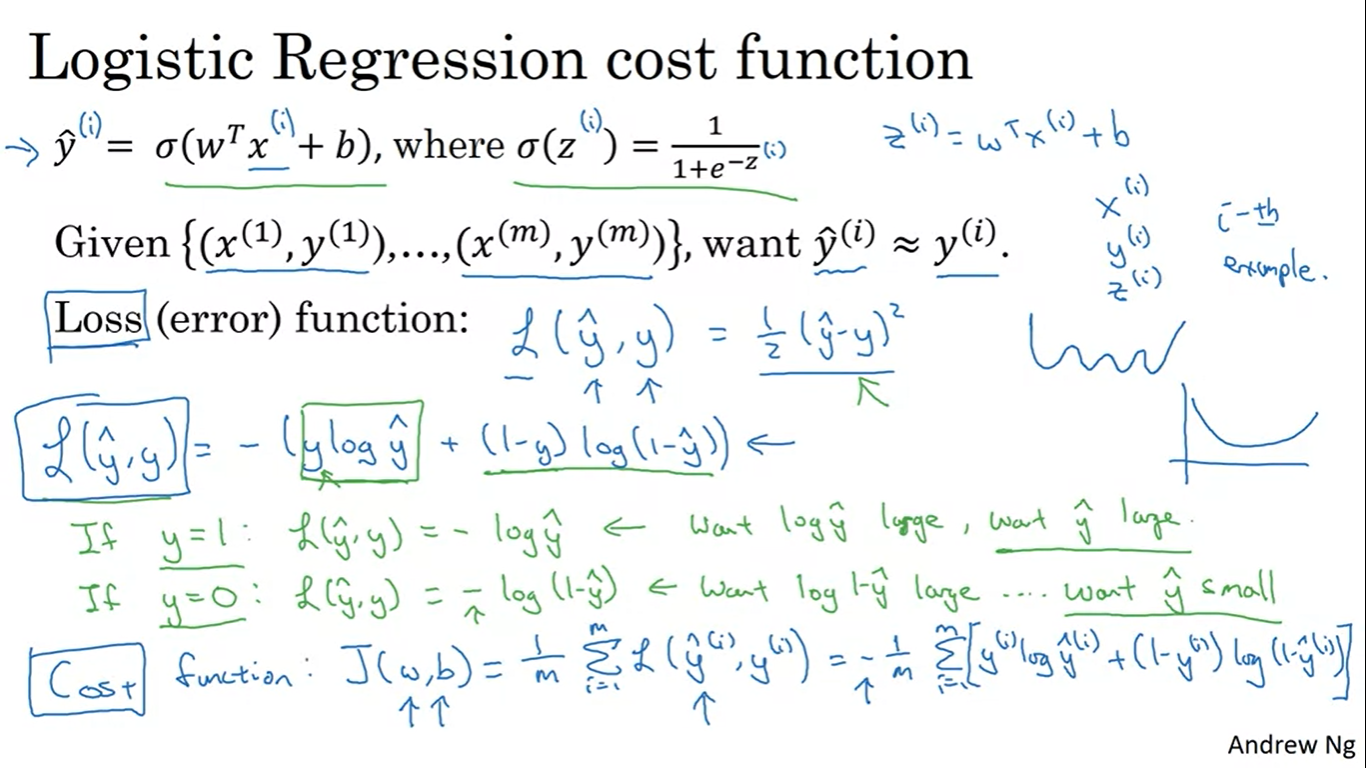
We know that the value of Y is a binary output i.e for a given image(x), what are the chances it could be a cat(y = 1). Suppose if we consider to build a model using a simple linear regression, then out y could be from (-infinity, infinity), which contradicts to our hypothesis, hence we take a sigmoid transformation of linear regression to get output y. The output of sigmoid function lies in between (0,1).

In this curve we have two parameter w and b. We calibrate these values separately when fitting in a NN, because w is a n-dimensional vector and b is constant for given x =1.



### Logistic Regression Cost Function:

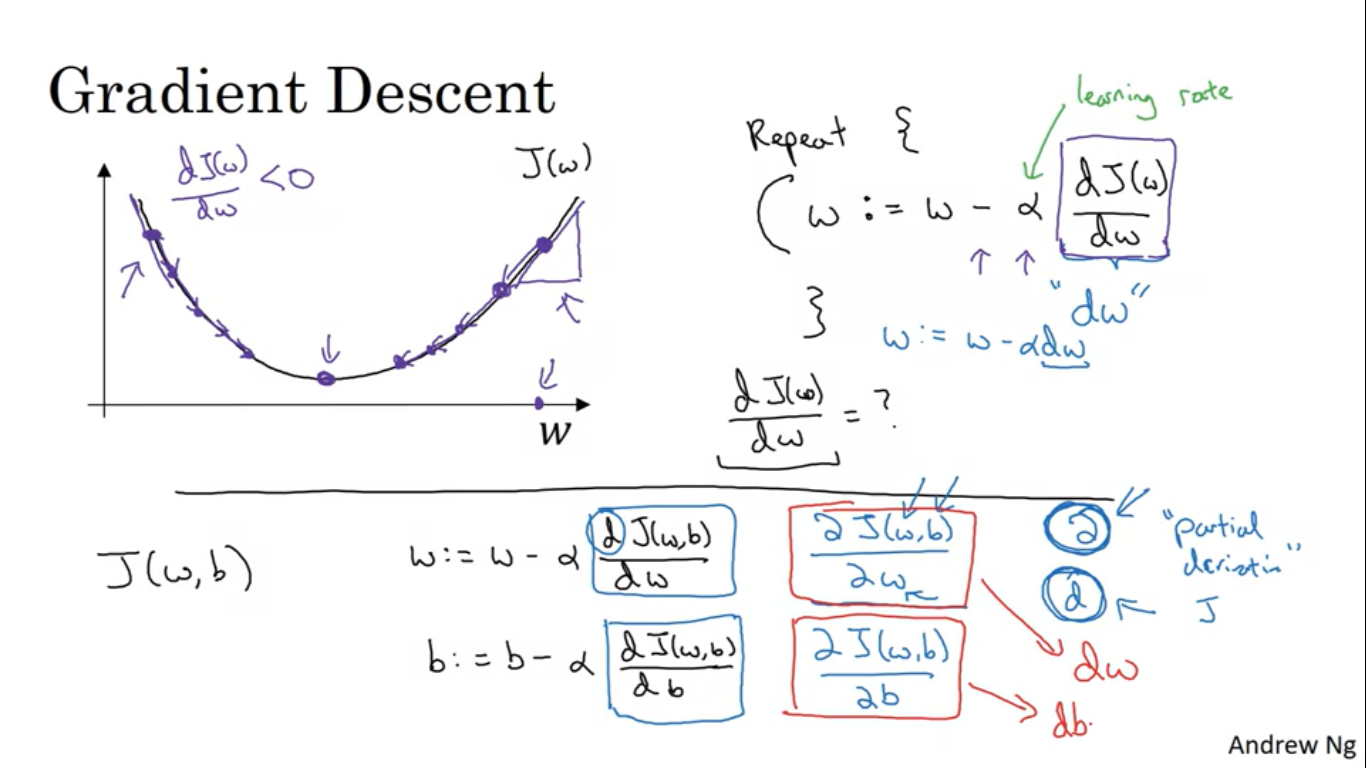
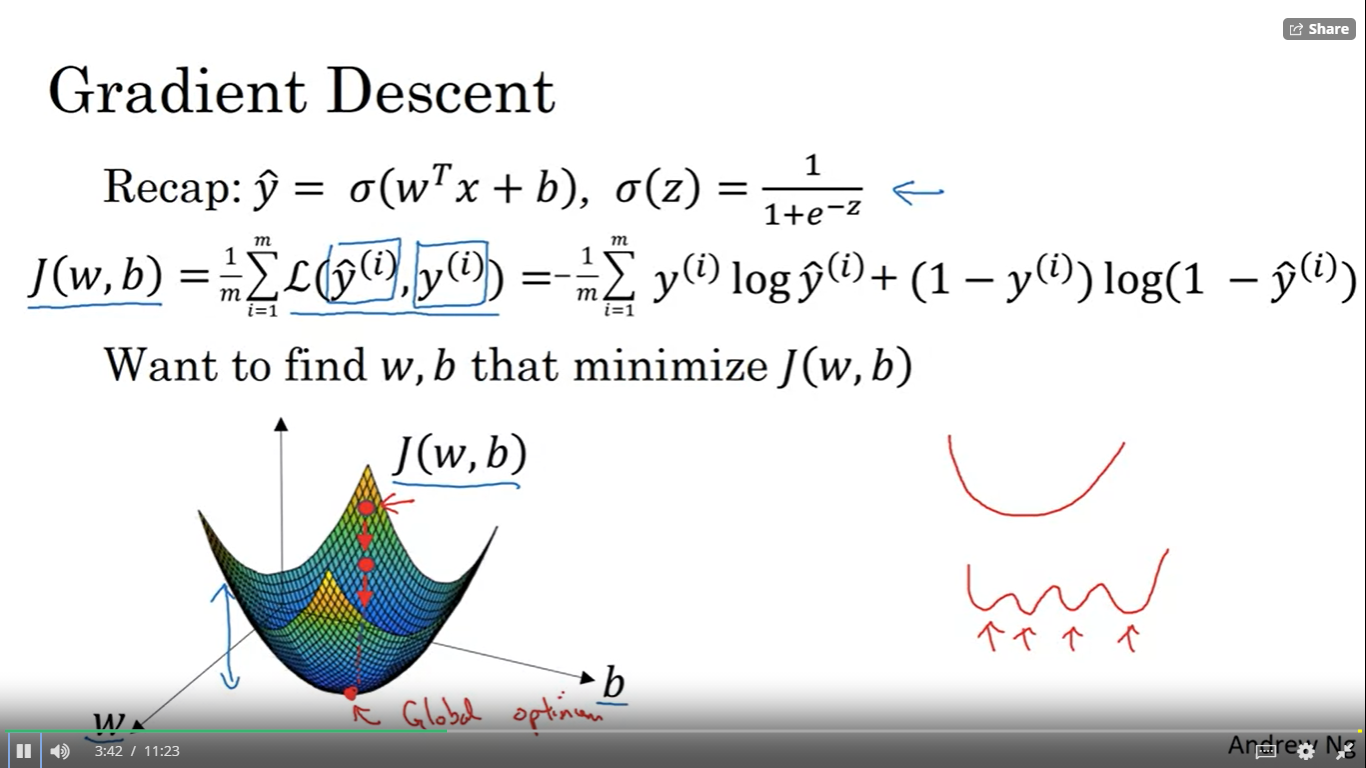
To validate the performance of a model we have two methods the loss function and the cost functions. Loss functions is square of difference from predicted and actual for each event, while cost function is average of all the events. We use cost function to calibrate w and b.



### Gradient:

We consider the cost function follows a simple convex plot between w, b, and y(predicted) – y(actual). To minimise the convex function, we need to calibrate the global minimum. This global minimum is calibrated by initializing w with a value and a learning rate. To simplify we will convert the 3D graph to a 2D graph by removing b value from the graph and calibrate the optimal w. Slope at ‘w’ will be derivate of ‘w’.

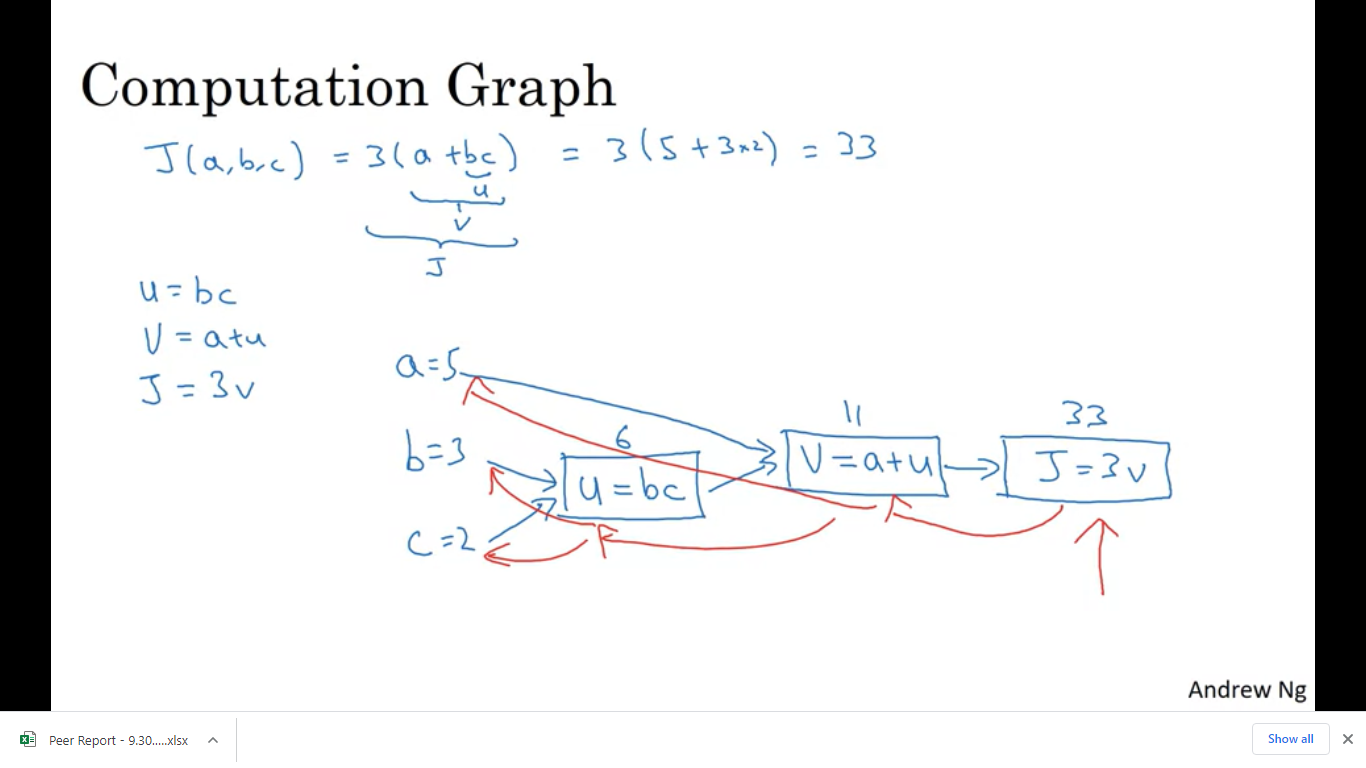
\*The slope of a straight line doesn’t change as the learning rate increases/decreases but the slope of curved line changes



### Computation Graph:

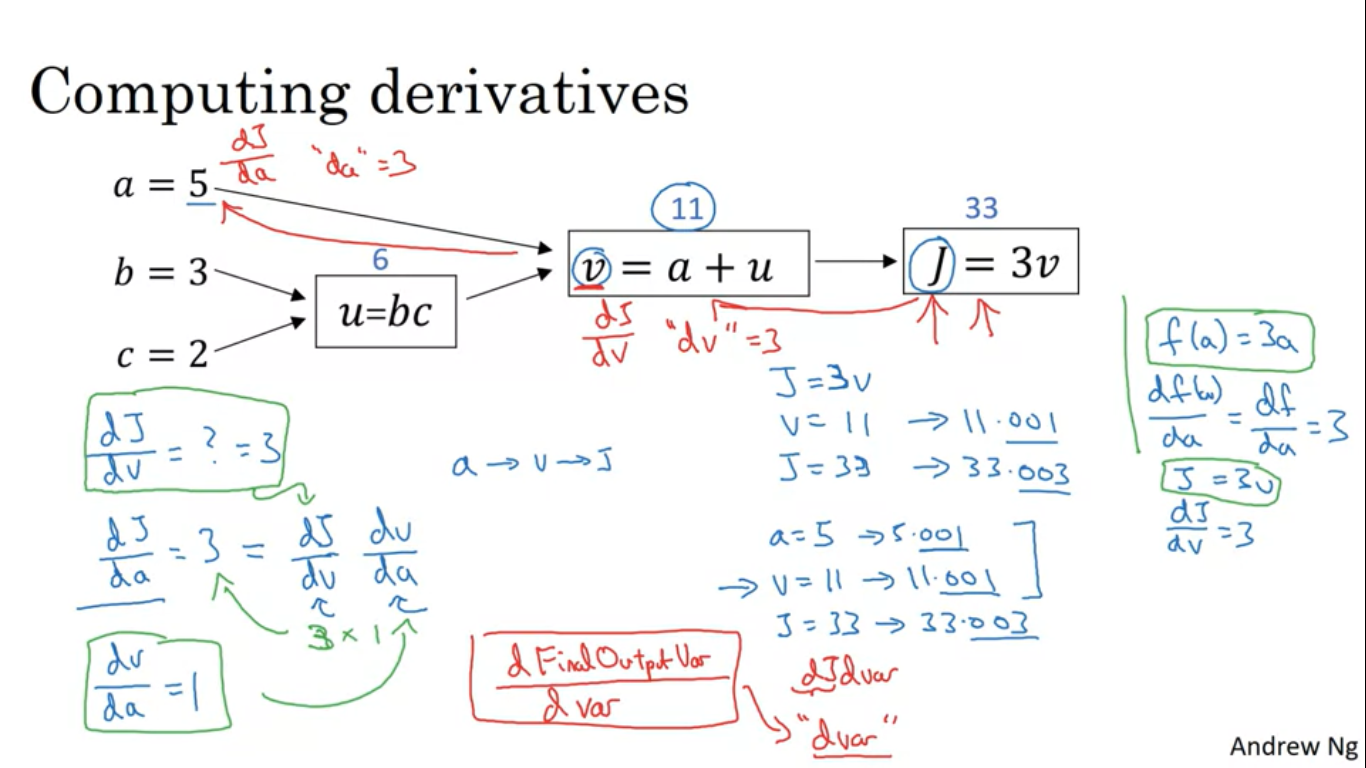
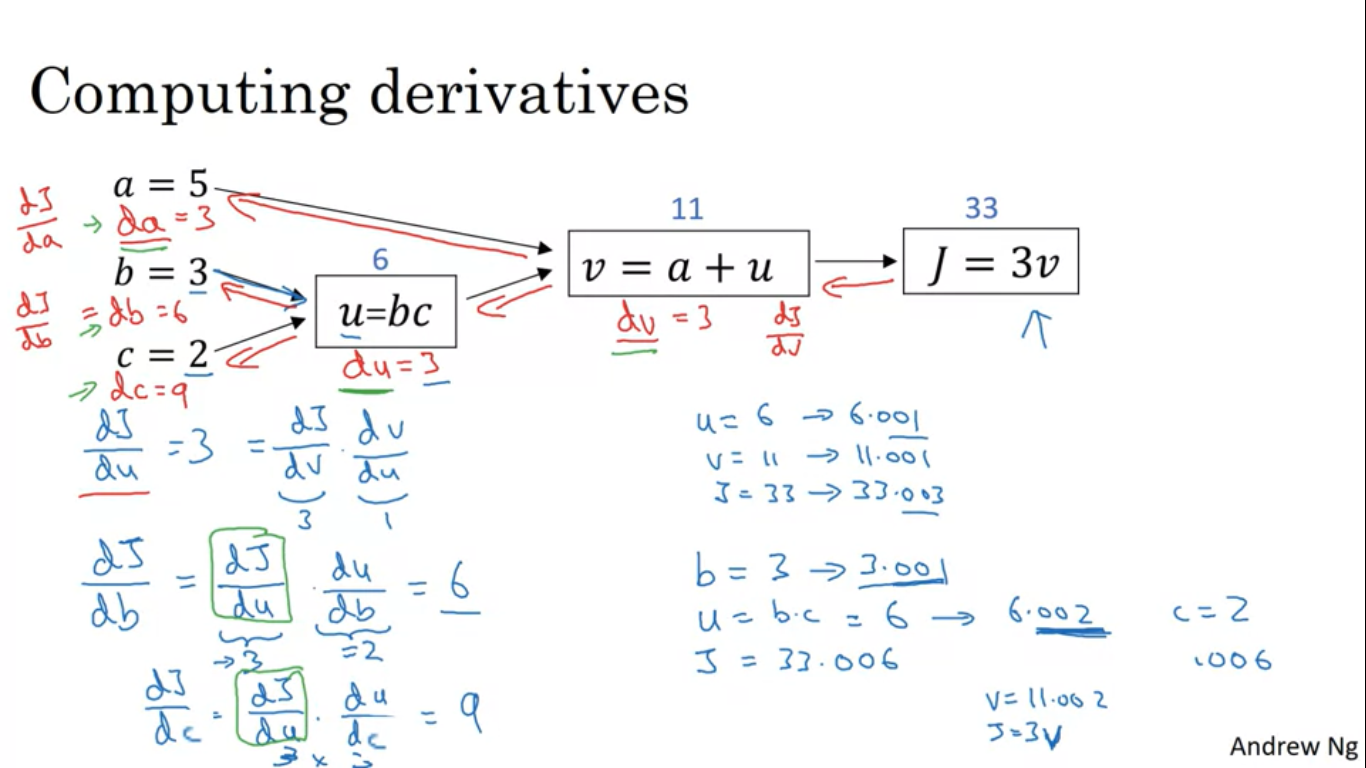
Computation of NN are organized in terms of forward pass or a forward propagation step. This help us to calibrate the output. From a backward propagation we can compute the gradient or derivatives as shown in the picture below.

#### Forward propagation:



#### Backword Propagation:

Chain rules is used to calibrate these values

### Logistic Regression Gradient Descent:

