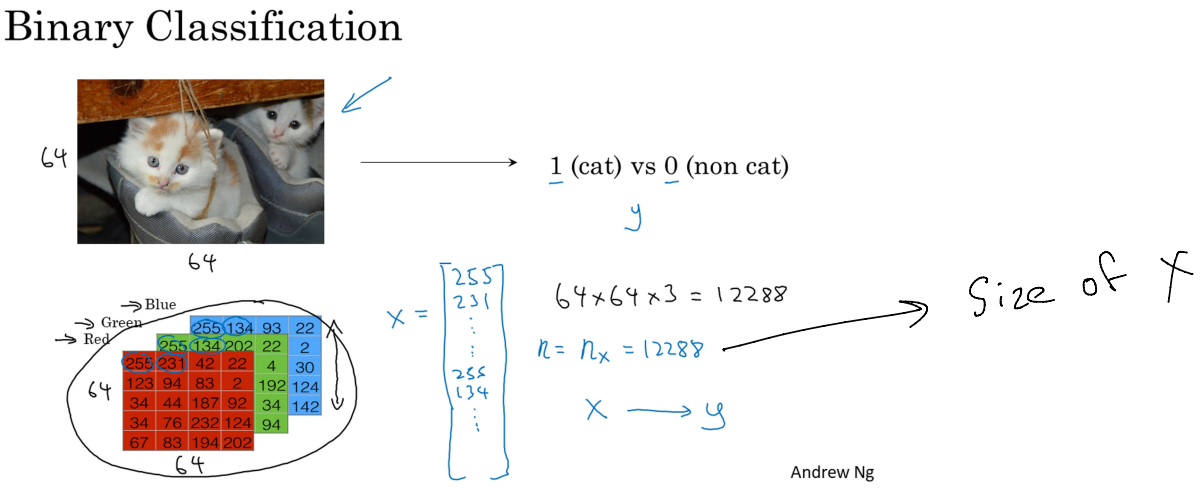
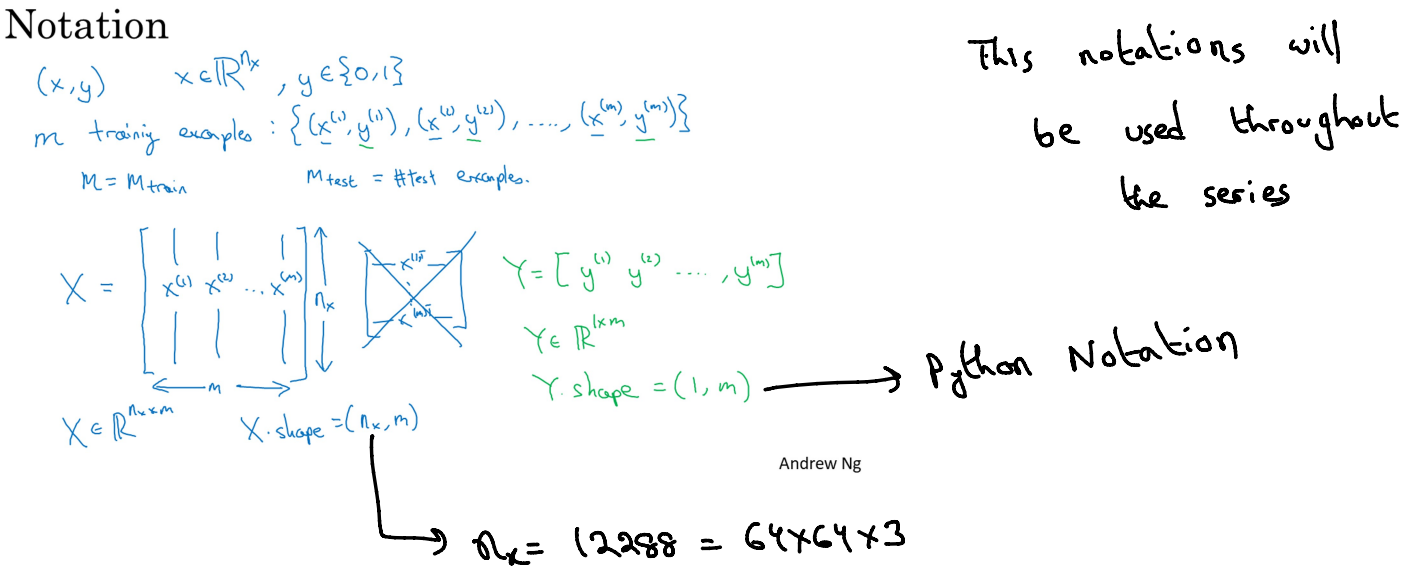
Binary Classification

Thursday, March 4, 2021 10:07 PM

Logistic regression is an algorithm for binary classification.
So let's start by setting up the problem.
Here's an example of a binary classification problem.
You might have an input of an image, like that, and
want to output a label to recognize this image as either being a cat,
in which case you output 1, or not-cat in which case you output 0,
and we're going to use y to denote the output label.
Let's look at how an image is represented in a computer.
To store an image your computer stores three separate matrices
corresponding to the red, green, and blue color channels of this image.





So, the columns of X will be each test set, where each column will have the 64*64*3 (for RGB) data(row), and m number of columns.

Logistic Regression

Thursday, March 4, 2021 11:54 PM

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.

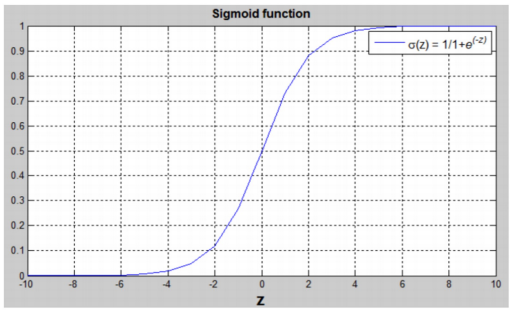
Example: Cat vs No - cat

Given an image represented by a feature vector x, the algorithm will evaluate the probability of a cat being in that image.

$$Given\ x,\ \hat{y} = P(y = 1|x), \text{where } 0 \leq \hat{y} \leq 1$$

The parameters used in Logistic regression are:

- The input features vector: $x \in \mathbb{R}^{n_x}$, where n_x is the number of features
- The training label: $y \in 0,1$
- The weights: $w \in \mathbb{R}^{n_x}$, where n_x is the number of features
- The threshold: $b \in \mathbb{R}$
- The output: $\hat{y} = \sigma(w^T x + b)$
- Sigmoid function: s = $\sigma(w^T x + b)$ = $\sigma(z)$= $\frac{1}{1+e^{-z}}$

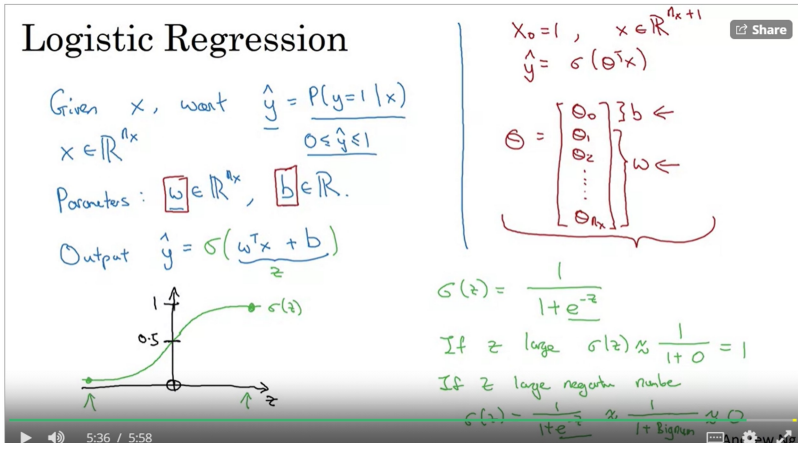


$(w^T x + b)$ is a linear function $(ax + b)$, but since we are looking for a probability constraint between [0,1], the sigmoid function is used. The function is bounded between [0,1] as shown in the graph above.

Some observations from the graph:

- If z is a large positive number, then $\sigma(z)$ = 1
- If z is small or large negative number, then $\sigma(z)$ = 0
- If $z = 0$, then $\sigma(z)$ = 0.5

Logistic Regression



Alternative Notation.

But it's easy to implement if w and b are seperate.

So, our goal is to find the w and b

Logistic Regression: Cost Function

Friday, March 5, 2021 2:04 AM

Logistic Regression: Cost Function

To train the parameters w and b, we need to define a cost function.

Recap:

$\hat{y}^{(i)} = \sigma(w^T x^{(i)} + b)$, where $\sigma(z^{(i)}) = \frac{1}{1+e^{-z^{(i)}}}$

$x^{(i)}$ the i-th training example

Given $\{(x^{(1)}, y^{(1)}), \cdots, (x^{(m)}, y^{(m)})\}$, *we want* $\hat{y}^{(i)} \approx y^{(i)}$

Loss (error) function:

The loss function measures the discrepancy between the prediction ($\hat{y}^{(i)}$) and the desired output ($y^{(i)}$). In other words, the loss function computes the error for a single training example.

$$L(\hat{y}^{(i)}, y^{(i)}) = \frac{1}{2}(\hat{y}^{(i)} - y^{(i)})^2$$

$$L(\hat{y}^{(i)}, y^{(i)}) = -(y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

- If $y^{(i)} = 1$: $L(\hat{y}^{(i)}, y^{(i)}) = -\log(\hat{y}^{(i)})$ where $\log(\hat{y}^{(i)})$ and $\hat{y}^{(i)}$ should be close to 1
- If $y^{(i)} = 0$: $L(\hat{y}^{(i)}, y^{(i)}) = -\log(1 - \hat{y}^{(i)})$ where $\log(1 - \hat{y}^{(i)})$ and $\hat{y}^{(i)}$ should be close to 0

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.

$$J(w, b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m}[(y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})]$$

So, to train the data,

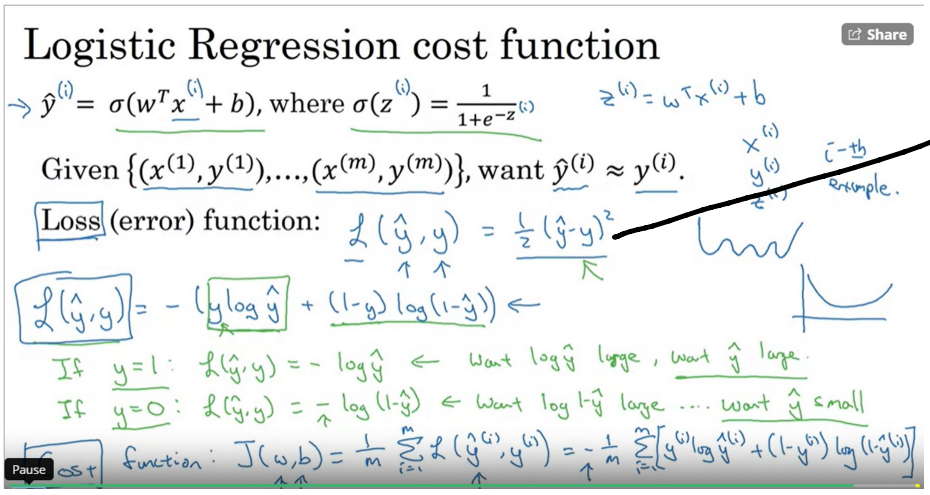
$$X = \begin{bmatrix} x^1 & x^2 & x^3 & x^4 & \cdots & x^m \\ \vdots & & & & & \end{bmatrix}$$

$$y = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}_{1 \times m}$$

$\hat{y}$ → prediction, y → output

so, for $x^{(i)}$, we want, $\hat{y}^{(i)} \approx y^{(i)}$

Logistic Regression Cost Function

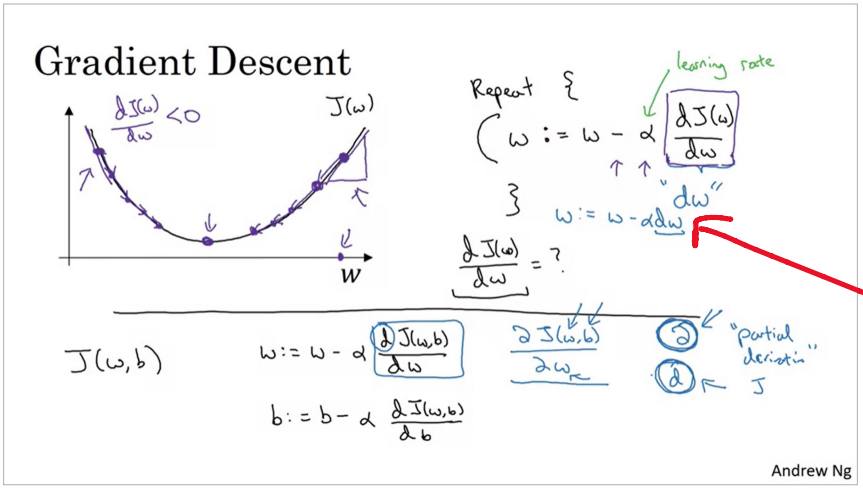


Squared error is not used because it creates many local minima

Gradient Descent

Friday, March 5, 2021 2:46 AM

Gradient Descent



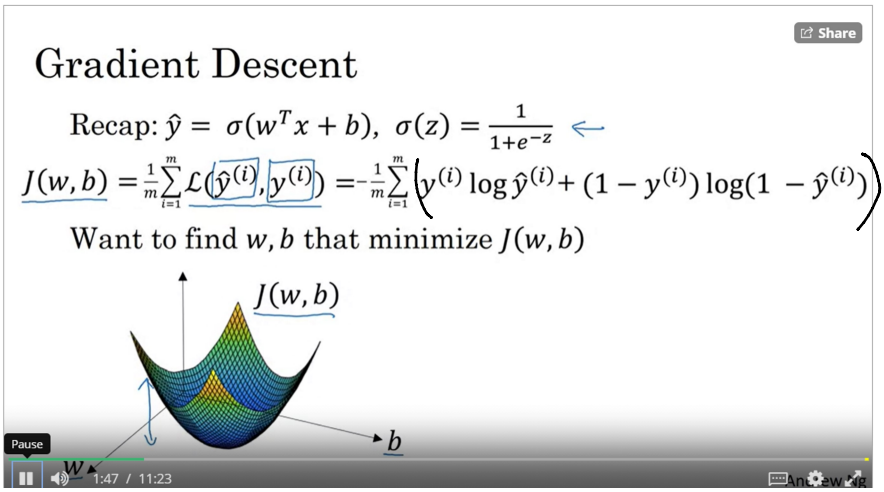
IF $\frac{dJ(w)}{dw}$ is positive, w decreases,

if $\frac{dJ(w)}{dw}$ is negative, w increases.

python notation

We use gradient descent to reach the global optimum.

Gradient Descent

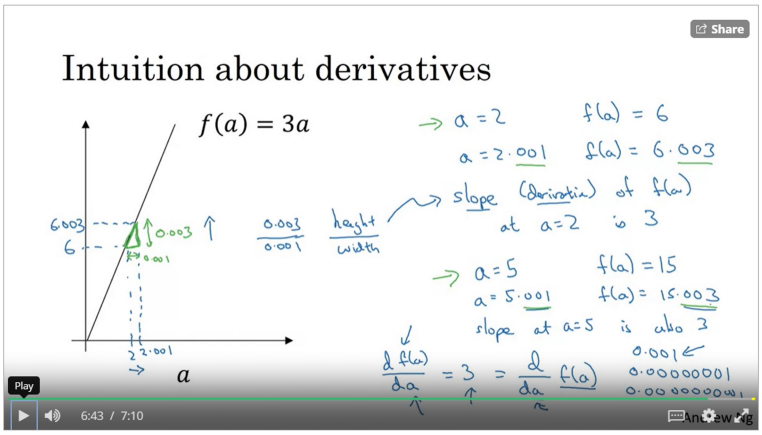


→ Summation on the whole function

Derivatives

Friday, March 5, 2021 2:59 AM

Derivatives



If we increase,
$a = 2.0001$
$\rightarrow f(a)$ becomes 6.0003

So,
the "change in $f(a)$" with respect to "change in a" is

$$= \frac{6.0003 - 6}{2.0001 - 2} = 3$$

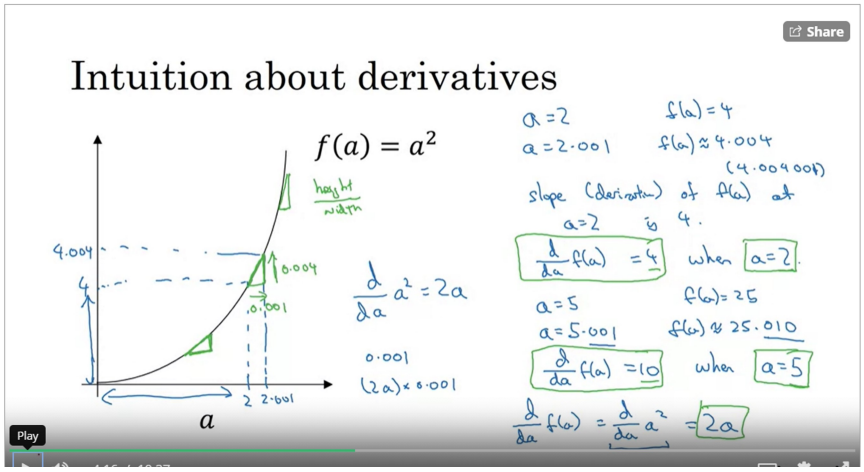
So, $\frac{df(a)}{da} = 3$

"it is true for any point on the line as long as da = same

Summary:
let's look at a few points on this function. Let say that a is equal to two. In that case, f of a, which is equal to three times a is equal to six. So, if a is equal to two, then f of a will be equal to six. Let's say we give the value of a just a little bit of a nudge. I'm going to just bump up a, a little bit, so that it is now 2.001. So, I'm going to give a like a tiny little nudge, to the right. So now, let's say 2.001, just plot this into scale, 2.001, this 0.001 difference is too small to show on this plot, just give a little nudge to that right. Now, f(a), is equal to three times that. So, it's 6.003, so we plot this over here. This is not to scale, this is 6.003. So, if you look at this little triangle here that I'm highlighting in green, what we see is that if I nudge a 0.001 to the right, then f of a goes up by 0.003. The amounts that f of a, went up is three times as big as the amount that I nudge the a to the right. So, we're going to say that, the slope or the derivative of the function f of a, at a equals to or when a is equals two to the slope is three. The term derivative basically means slope, it's just that derivative sounds like a scary and more intimidating word, whereas a slope is a friendlier way to describe the concept of derivative. So, whenever you hear derivative, just think slope of the function. More formally, the slope is defined as the height divided by the width of this little triangle that we have in green. So, this is 0.003 over 0.001, and the fact that the slope is equal to three or the derivative is equal to three, just represents the fact that when you nudge a to the right by 0.001, by tiny amount, the amount at f of a goes up is three times as big as the amount that you nudged it, that you nudged a in the horizontal direction. So, that's all that the slope of a line is. Now, let's look at this function at a different point. Let's say that a is now equal to five. In that case, f of a, three times a is equal to 15. So, let's see that again, give a, a nudge to the right. A tiny little nudge, it's now bumped up to 5.001, f of a is three times that. So, f of a is equal to 15.003. So, once again, when I bump a to the right, nudge a to the right by 0.001, f of a goes up three times as much. So the slope, again, at a = 5, is also three. So, the way we write this, that the slope of the function f is equal to three: We say, d f(a) da and this just means, the slope of the function f(a) when you nudge the variable a, a tiny little amount, this is equal to three. An alternative way to write this derivative formula is as follows. You can also write this as, d da of f(a). So, whether you put f(a) on top or whether you write it down here, it doesn't matter. But all this equation means is that, if I nudge a to the right a little bit, I expect f(a) to go up by three times as much as I nudged the value of little a. Now, for this video I explained derivatives, talking about what happens if we nudged the variable a by 0.001. If you want a formal mathematical definition of the derivatives: Derivatives are defined with an even smaller value of how much you nudge a to the right. So, it's not 0.001. It's not 0.000001. It's not 0.00000000 and so on 1. It's even smaller than that, and the formal definition of derivative says, whenever you nudge a to the right by an infinitesimal amount, basically an infinitely tiny, tiny amount. If you do that, this f(a) go up three times as much as whatever was the tiny, tiny, tiny amount that you nudged a to the right. So, that's actually the formal definition of a derivative. But for the purposes of our intuitive understanding, which I'll talk about nudging a to the right by this small amount 0.001. Even if it's 0.001 isn't exactly tiny, tiny infinitesimal. Now, one property of the derivative is that, no matter where you take the slope of this function, it is equal to three, whether a is equal to two or a is equal to five. The slope of this function is equal to three, meaning that whatever is the value of a, if you increase it by 0.001, the value of f of a goes up by three times as much. So, this function has a safe slope everywhere. One way to see that is that, wherever you draw this little triangle. The height, divided by the width, always has a ratio of three to one. So, I hope this gives you a sense of what the slope or the derivative of a function means for a straight line, where in this example the slope of the function was three everywhere.

Here the derivative depends on "a"

More Derivative Examples



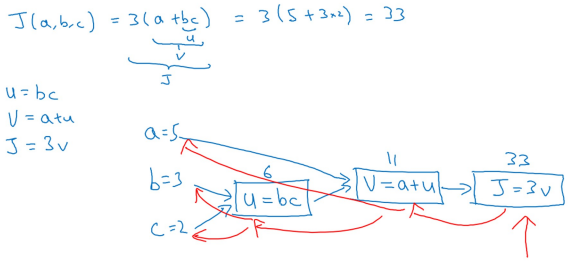
Computational Graph

Friday, March 5, 2021 3:45 AM

This helps us to learn about back propagation.

As we need dJ, partially differentiated with respect to w,
We use back propagation to update the value.

Computation Graph



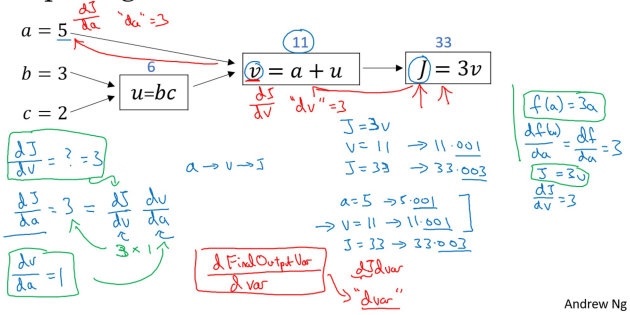
For example,

$$\frac{dJ}{da} = \frac{dJ}{dv} \cdot \frac{dv}{da}$$

we calculate these, to get $\frac{dJ}{da}$

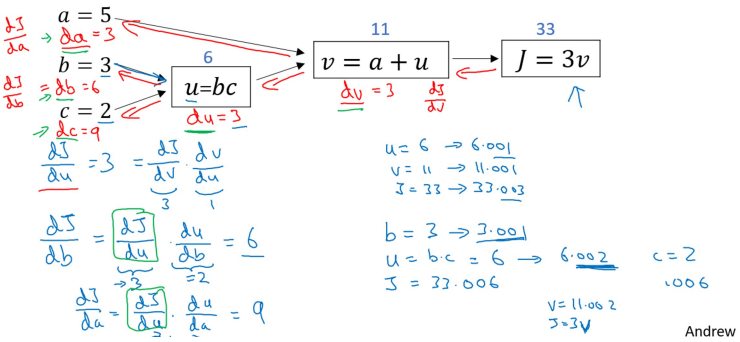
This follows the red line, so, it is called back propagation

Computing derivatives



Andrew Ng

Computing derivatives

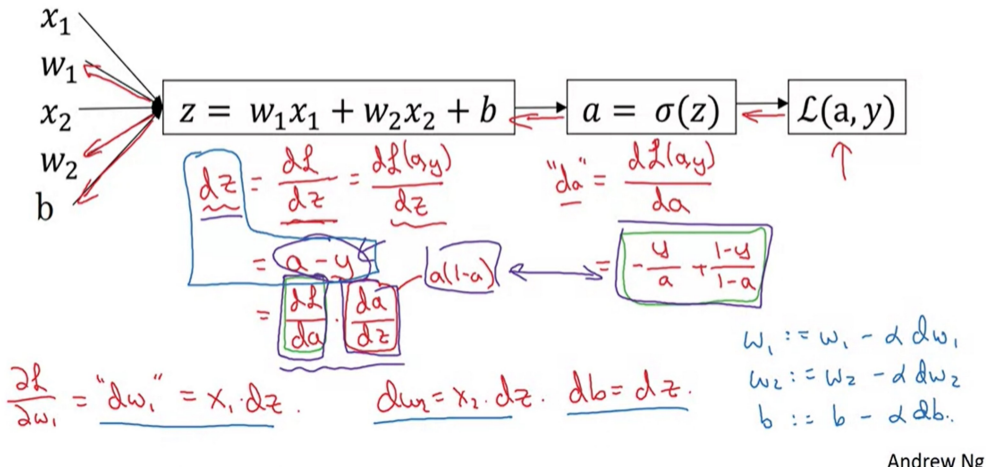


Andrew

Gradient Descent - Logistic Regression

Friday, March 5, 2021 4:09 AM

Logistic regression derivatives



Andrew Ng

Steps

1) $\frac{dL}{da} = -\frac{y}{a} + \frac{1-y}{1-a}$

2) $\frac{dL}{dz}$

$= \frac{dL}{da} \cdot \frac{da}{dz}$

step (1) calculating $a(1-a)$

$= \frac{dL}{dz} = a - y$

3) $\frac{dL}{dw_1} = x_1 \cdot dz$ ($dz = \frac{dL}{dz}$)

$dw_1 = \frac{dL}{dw_1} = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{dw_1}$

dw_1 → python

$\frac{dL}{da} \cdot \frac{da}{dz}$ → python $dz = \frac{dL}{dz}$

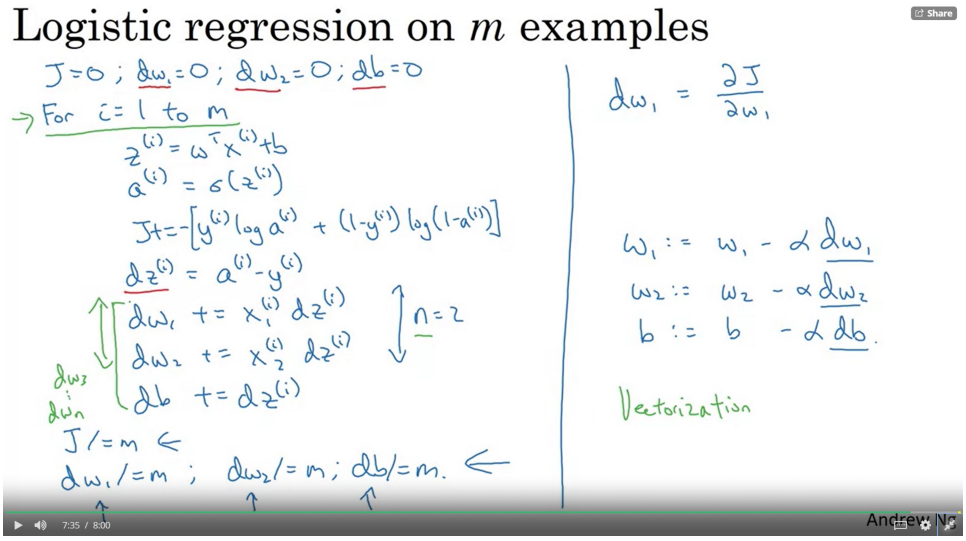
$\frac{dz}{dw_1} = x_1$ [because only done with respect to x_1.]

$= dz \cdot x_1$

then, $w_1 = w_1 - \alpha \cdot dw_1$ ← Gradient Descent

Gradient Descent on M examples

Friday, March 2021 12:26 PM



2 for loops will be needed

- i=1 to m
- for dw1,.... dwn

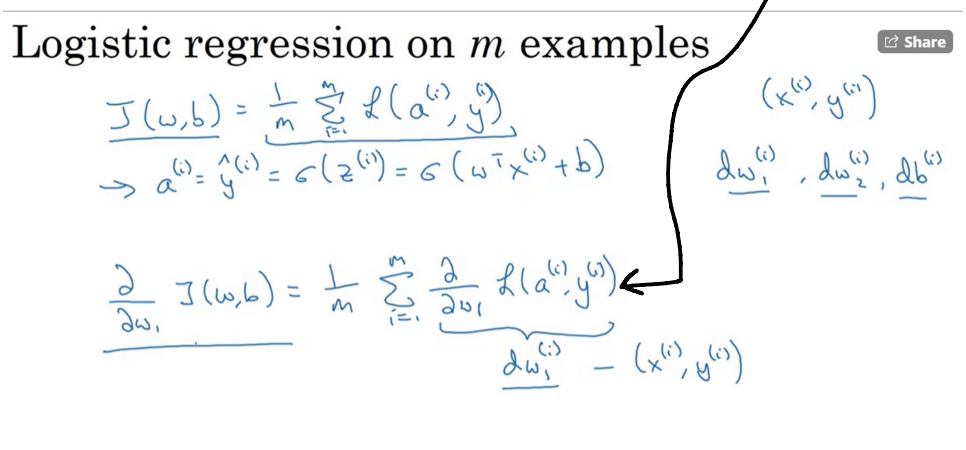
So, we need, Vectorization

1) Calculate z(i)
2) Calculate a(i)
3) Calculate Jf(i) and add that to J (which is initially 0)
4) Calculate dz(i)
5) Calculate dw1(i) and add that to dw1
6) Calculate dw2(i) and add that to dw2
7) Same for db
8) Divide everything with m
9) Update w1,w2,b

} so every single training example will be calculated i=1 to m
J will be cost func for i=1 to m.
J = J/m

} summation because

$$\frac{d}{dw_1}(J(w,b)) = \frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial w_1}\mathcal{L}\left(a^{(i)}, y^{(i)}\right)$$



} summation of all m examples