Q and A

Q: I learned from the course that L = - (ylog(y_hat) + (1-y)log(1-y_hat)), and understand that it can correctly punish wrong values. However, I wonder about one case that if y is 1 and y_hat is 0. Then the Loss is infinity. Then, how can the training continue since "+ * /" cannot apply to infinity? Will this whole training destroyed?

A:

> It is a really interesting question! From a mathematical standpoint, $\hat{y}$ is the output of the sigmoid function, so it can never be exactly 0 or 1. But we are operating in the very limited world of 64 bit floating point, as opposed to the abstract beauty of $\mathbb{R}$, so it can actually happen that $\hat{y}$ rounds to 0 or 1. If that happens and the label turns out to be the opposite, then the J cost value becomes NaN (not a number), but gradient descent actually still works. The gradients are the derivatives of J w.r.t. the various parameters, so they are still valid values because they do not directly depend on the J value. For example, here is the formula for dw :
>
> $$dw = \frac{1}{m} X \cdot (A - Y)^T$$
>
> In that formula, the $\hat{y}$ values are the elements of A . So you can see that nothing bad happens if any of the A elements happen to be 0 or 1. Thus back propagation works and gradient descent can still learn, even if the J value is nonsense.

Just to extend the thought process a little more, the one other thing that is almost a bit surprising is that the value J is not actually used at all as part of gradient descent, so you might well ask why we bother computing and printing it. The answer is that we use it as an inexpensive to compute proxy for judging whether gradient descent is converging or not. Note that Logistic Regression is a very special case in which the cost function is actually convex, but once we graduate to real Neural Networks in Week 3, that will not longer be true. But even in the case of LR, convergence is not guaranteed: if you choose too high a learning rate, you can get oscillation or even divergence, rather than convergence. For real Neural Networks, it's even more likely that getting convergence will require some tuning of the learning rate. Here in Course 1, there are just too many things to talk about, so Prof Ng saves that kind of tuning topic for Course 2 and they just give us working learning rates in all the assignments in Course 1.