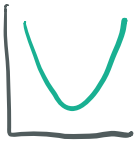
DS 100: Principles and Techniques of Data Science **Date: July 24, 2019**

Discussion #9

Name: Raguvir Kunani

Loss Functions

Squared



Absolute

Huber

1. Recall the loss functions discussed during lecture. θ represents our estimate of our parameter, and y represents a data point. Discuss the advantages and drawbacks of each of the following loss functions:

   θ, y are scalars here

   when θ, y, x are vectors, then we compute the loss of each $(\hat{y}, y)$ pair [where $\hat{y} = \text{model}(\theta, x)$] and take the average of all these losses; sometimes the $(\hat{y}, y)$ pairs are denoted as $(f_\theta(x_i), y_i)$

   (a) Squared loss: $L(\theta, y) = (y - \theta)^2$

   A: can find minimum with calculus

   D: penalizes outliers heavily

   (b) Absolute Loss: $L(\theta, y) = |y - \theta|$

   A: penalizes outliers proportionally

   D: minimum point is not differentiable

   (c) Huber Loss: $L_\alpha(\theta, y) = \begin{cases} (y-\theta)^2 & |y-\theta| < \alpha \\ \alpha(|y-\theta| - \alpha/2) & \text{otherwise} \end{cases}$

   A: penalizes outliers proportionally and has differentiable minimum

   D: complicated derivative

α

choose a small α to retain the advantages of absolute loss

Loss Minimization

2. Consider the following loss function:

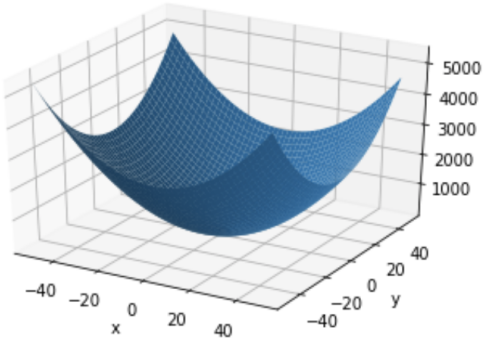
$$L(\theta, x) = \begin{cases} 4(\theta - x) & \theta \geq x \\ x - \theta & \theta < x \end{cases}$$

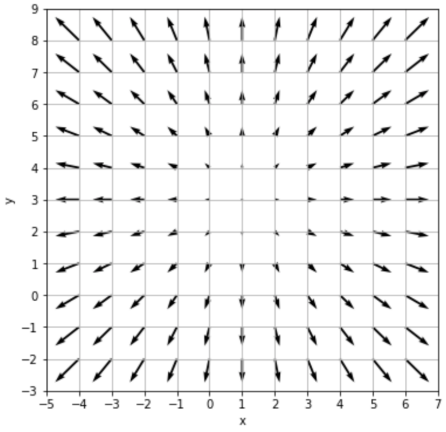
Given a sample of $x_1, ..., x_n$, find the optimal θ that minimizes the the average loss.

See end of worksheet for solutions

Gradients

3. On the left is a 3D plot of $f(x, y) = (x-1)^2 + (y-3)^2$. On the right is a plot of its **gradient field**. Note that the arrows show the relative magnitudes of the gradient vector.





(a) From the visualization, what do you think is the minimal value of this function and where does it occur?

(1,3)

The gradient field has no arrows at (1,3) which means the gradient vector at (1,3) has length 0

(b) Calculate the gradient $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^T$.

$\frac{\partial f}{\partial x} = 2(x-1)$

$\frac{\partial f}{\partial y} = 2(y-3)$

$\Rightarrow \nabla f = \begin{bmatrix} 2(x-1) \\ 2(y-3) \end{bmatrix}$

The gradient is a vector that shows the direction of steepest ascent from a given point

(c) When $\nabla f = \mathbf{0}$, what are the values of x and y?

$\nabla f = \vec{0} \Rightarrow \begin{bmatrix} 2(x-1) \\ 2(y-3) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \begin{matrix} x = 1 \\ y = 3 \end{matrix}$

4. In this question, we will explore some basic properties of the gradient.

   Note: In this class, we use the following conventions:

   - x represents a scalar
   - X represents a random variable
   - $\mathbf{x}$ represents a vector
   - $\mathbf{X}$ represents a matrix or a random vector (context will tell)

   (a) Determine the derivative of $f(x) = a_0 + a_1 x$ and gradient of $g(x_1, x_2) = a_0 + a_1 x_1 + a_2 x_2$.

$f'(x) = a_1$

$$\nabla \vec{g} = \begin{bmatrix} \frac{\partial g(x_1, x_2)}{\partial x_1} \\ \frac{\partial g(x_1, x_2)}{\partial x_2} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

   (b) Suppose $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T$, and $h(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$, where $\mathbf{a}, \mathbf{x} \in \mathbb{R}^n$. Determine ∇h.

$\vec{a}$ is a vector, $\vec{x}$ is a vector $\Rightarrow \vec{a}^T\vec{x}$ is the dot product of $\vec{a}$ and $\vec{x}$

$$\vec{a}^T\vec{x} = \sum_{i=1}^{n} a_i x_i \Rightarrow \nabla h = \begin{bmatrix} \frac{\partial h(\vec{x})}{\partial x_1} \\ \frac{\partial h(\vec{x})}{\partial x_2} \\ \vdots \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \vec{a}$$

   (c) Determine the gradient of $f(\mathbf{x}) = \mathbf{x}^T\mathbf{x}$. *(Hint: f is a scalar-valued function. How can you write $\boldsymbol{x}^T\boldsymbol{x}$ as a sum of scalars?)*

$$f(\vec{x}) = \vec{x}^T\vec{x} = \sum_{i=1}^{n} x_i x_i = \sum_{i=1}^{n} x_i^2 = x_1^2 + x_2^2 + \dots + x_n^2$$

$$\nabla f = \begin{bmatrix} \frac{\partial f(\vec{x})}{\partial x_1} \\ \frac{\partial f(\vec{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\vec{x})}{\partial x_n} \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_n \end{bmatrix} = 2\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = 2\vec{x}$$

Gradient Descent Algorithm

5. Given the following loss function and $\mathbf{x} = (x_i)_{i=1}^n$, $\mathbf{y} = (y_i)_{i=1}^n$, θ^t, explicitly write out the update equation for θ^{t+1} in terms of x_i, y_i, θ^t, and α, where α is the step size.

$$L(\theta, \mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^{n} \left(\theta^2 x_i^2 - log(y_i)\right)$$

See end of worksheet for solutions

6. (a) In your own words, describe how to use the update equation in the gradient descent algorithm.

   (b) Say that x and y are your model parameters and f as defined in question 1 is your loss function. Describe in your own words what happens "visually" as the gradient descent algorithm runs.

See end of worksheet for solutions

Convexity

Convexity allows optimization problems to be solved more efficiently and for global optimums to be realized. Mainly, it gives us a nice way to minimize loss (i.e. gradient descent). There are three ways to informally define convexity.

a. Walking in a straight line between points on the function keeps you above the function. This works for any function.

b. The tangent line at any point lies below the function (globally). The function must be differentiable.

c. The second derivative is non-negative everywhere (aka "concave up" everywhere). The function must be twice differentiable.

7. Find a counterexample for the claim that the composition of two convex functions is also convex. $h = g(f(x))$

Loss Minimization

. Consider the following loss function:

$$L(\theta, x) = \begin{cases} 4(\theta - x) & \theta \geq x \quad x \leq \theta \\ x - \theta & \theta < x \quad x > \theta \end{cases}$$

Given a sample of $x_1, ..., x_n$, find the optimal θ that minimizes the the average loss.

The average loss for a loss function $L(\theta, x)$ is

$$\text{Avg. Loss} = \frac{1}{n} \sum_{i=1}^{n} L(\theta, x_i)$$ [each x_i represents one x in the samples $x_1 \dots x_n$]

We want to minimize average loss, so take the derivative:

$$\frac{\partial \text{ Avg. Loss}}{\partial \theta} = \frac{\partial}{\partial \theta}\left[\frac{1}{n}\sum_{i=1}^{n} L(\theta, x_i)\right] = \frac{\partial}{\partial \theta}\left[\sum_{i=1}^{n} L(\theta, x_i)\right] = \sum_{i=1}^{n} \frac{\partial}{\partial \theta}\left[L(\theta, x_i)\right] \quad (1)$$

$$\frac{\partial}{\partial \theta}\left[L(\theta, x_i)\right] = \begin{cases} 4 & x_i \leq \theta \\ -1 & x_i > \theta \end{cases} \quad (2)$$

← The derivative of a piecewise function is the derivative of each of its pieces

Plugging (2) into (1) and setting $\frac{\partial \text{ Avg. Loss}}{\partial \theta} = 0$:

$$\frac{\partial \text{ Avg. Loss}}{\partial \theta} = \sum_{i=1}^{n} \begin{cases} 4 & x_i \leq \theta \\ -1 & x_i > \theta \end{cases} = \sum_{x_i > \theta} -1 + \sum_{x_i \leq \theta} 4 = 0$$

↑ sum over all x_i that are greater than θ

$$\left(\sum_{x_i>\theta} -1\right) + \left(\sum_{x_i\le\theta} 4\right) = 0$$

Pull the constants out of the sum

$$-1\left(\sum_{x_i>\theta} 1\right) + 4\left(\sum_{x_i\le\theta} 1\right) = 0$$

Using $\sum_{i=1}^{n} 1 = \underbrace{1+1+\dots+1}_{n} = n$

$$-1\left(\#\, x_i>\theta\right) + 4\left(\#\, x_i\le\theta\right) = 0 \qquad (3)$$

↑ number of x_i that are greater than 0

We also know that $\#\, x_i>\theta + \#\, x_i\le\theta = n$, which means that $\#\, x_i\le\theta = n - \#\, x_i>\theta$ (4)

this is saying that the total # of points is the sum of # of points greater than θ and # of points less than or equal to θ

Plugging (4) into (3):

$$-1\left(\#\, x_i>\theta\right) + 4\left(n - \#\, x_i>\theta\right) = 0$$

$$4\left(n - \#\, x_i>\theta\right) = 1\left(\#\, x_i>\theta\right)$$

$$4n - 4\#\, x_i>\theta = \#\, x_i>\theta$$

$$4n = 5\,\#\, x_i>\theta$$

$$\#\, x_i>\theta = \frac{4}{5}n$$

This means we should choose θ so that 80% of $x_1 \dots x_n$ are greater than θ.

5. Given the following loss function and $\mathbf{x} = (x_i)_{i=1}^n$, $\mathbf{y} = (y_i)_{i=1}^n$, θ^t, explicitly write out the update equation for θ^{t+1} in terms of x_i, y_i, θ^t, and α, where α is the step size.

$$L(\theta, \mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^{n} \left(\theta^2 x_i^2 - log(y_i)\right)$$

The general form of the update equation is

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \alpha \frac{\partial L}{\partial \theta}$$

$$\frac{\partial L(\theta, \vec{x}, \vec{y})}{\partial \theta} = \frac{\partial}{\partial \theta}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\theta^2 x_i^2 - \log(y_i)\right)\right]$$

$$= \frac{1}{n}\frac{\partial}{\partial \theta}\left[\sum_{i=1}^{n}\left(\theta^2 x_i^2 - \log(y_i)\right)\right]$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial \theta}\left[\left(\theta^2 x_i^2 - \log(y_i)\right)\right]$$

$$= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\partial}{\partial \theta}\left[\theta^2 x_i^2\right] - \frac{\partial}{\partial \theta}\left[\log(y_i)\right]\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n} 2\theta x_i^2$$

$$= \boxed{\frac{2\theta}{n}\sum_{i=1}^{n} x_i^2}$$

6. (a) In your own words, describe how to use the update equation in the gradient descent algorithm.

   (b) Say that x and y are your model parameters and f as defined in question 1 is your loss function. Describe in your own words what happens "visually" as the gradient descent algorithm runs.

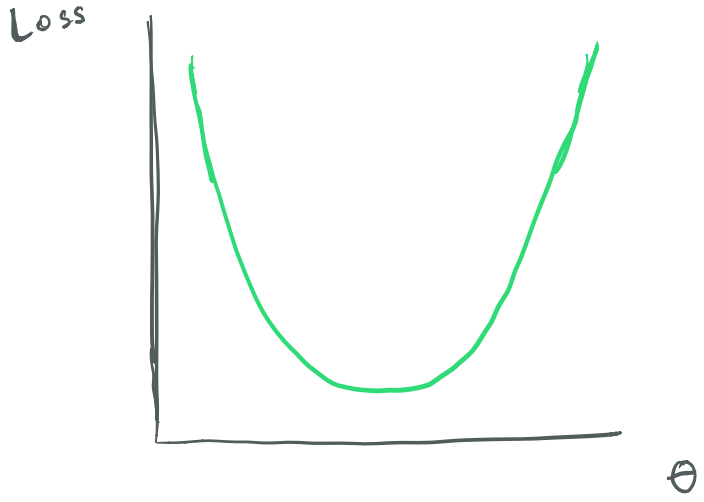
Recall the general update equation for gradient descent:

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \alpha \frac{\partial L}{\partial \theta}\Big|_{\theta=\theta^{(t)}}$$

$\frac{\partial L}{\partial \theta}$ is a function of θ, so evaluate it at $\theta = \theta^{(t)}$

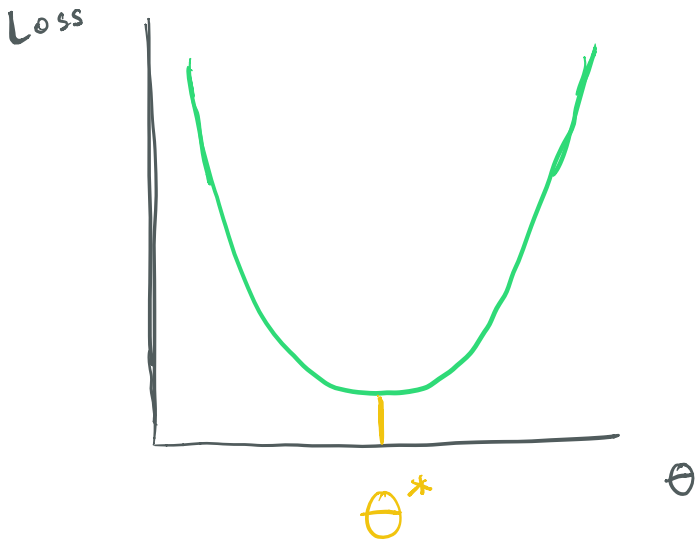
Let's look at an example of gradient descent in 2-D to see what gradient descent is actually doing.

Suppose we use the average squared loss, so our loss function looks something like:

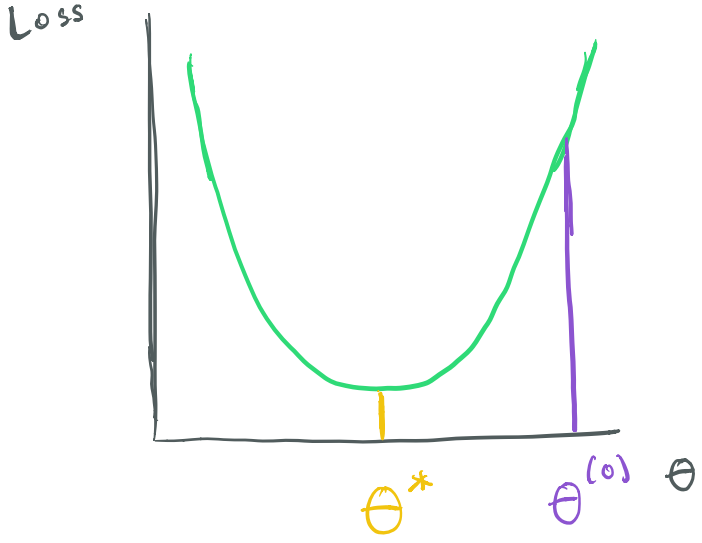


Remember, the goal of gradient descent is to find the θ that minimizes the loss function. I'll call that optimal value θ^*.

On the graph, θ^* looks like:

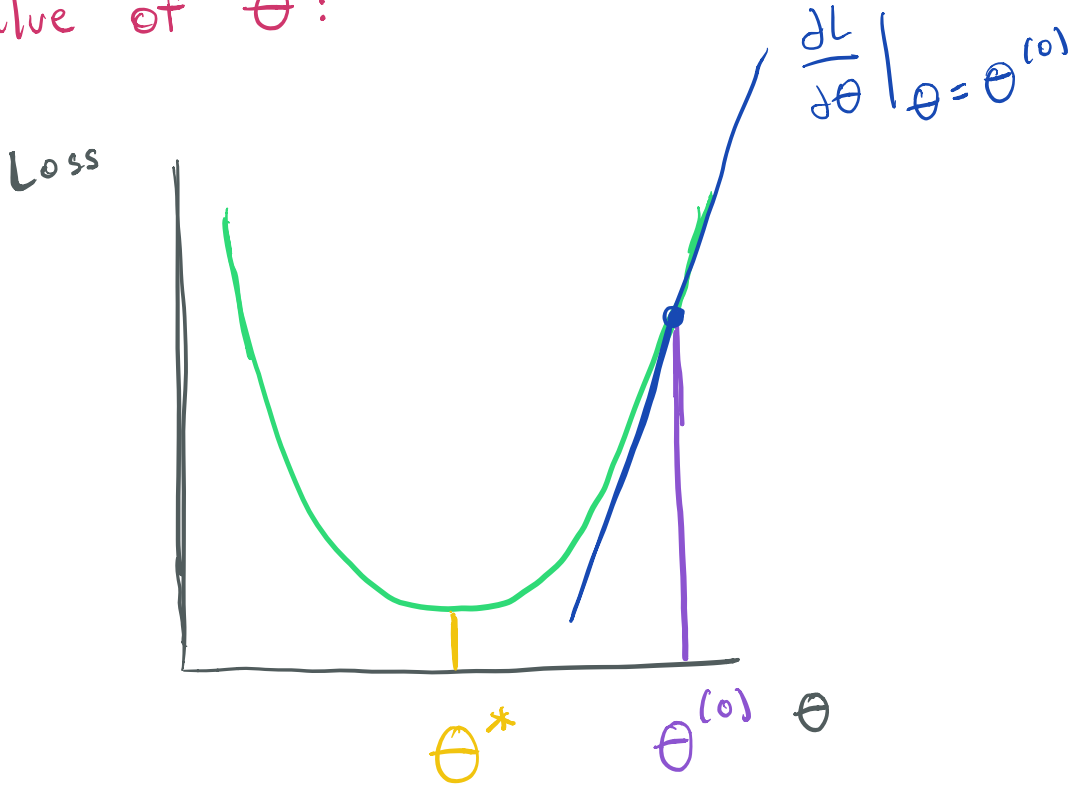


The way gradient descent works is that starts with a random guess for θ^*, which is the initial value $\theta^{(0)}$. On the graph, $\theta^{(0)}$ could be anywhere since it is a random guess.



Obviously, $\theta^{(0)}$ is not θ^*, as we can see from the graph. But how does gradient descent know that $\theta^{(0)} \neq \theta^*$? That's where $\frac{\partial L}{\partial \theta}$ comes in.

In our 2D example, $\frac{\partial L}{\partial \theta}$ represents the slope of the tangent line to the loss function at our current value of θ:



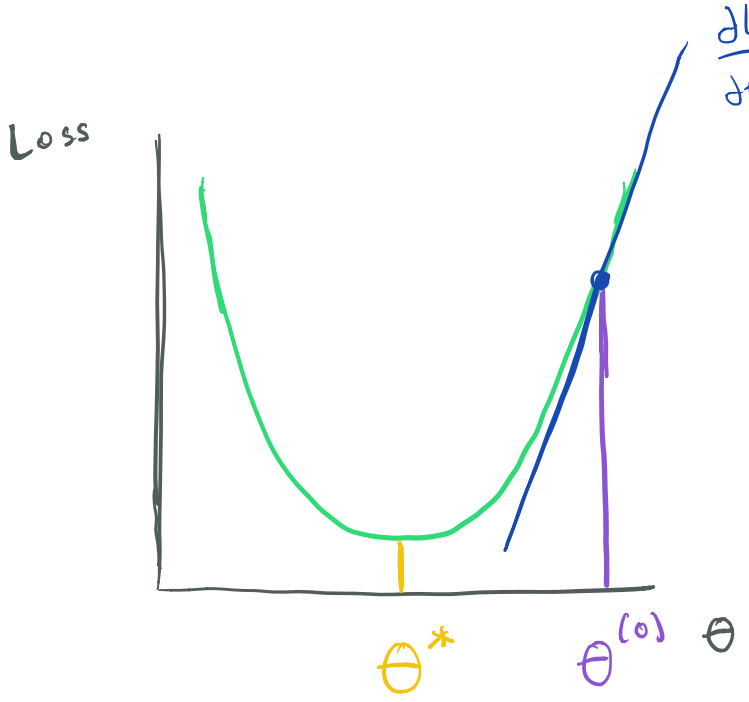
Since $\frac{\partial L}{\partial \theta}$ at $\theta = \theta^{(0)}$ is large and positive, gradient descent knows that $\theta^{(0)} \neq \theta^*$. This is because $\frac{\partial L}{\partial \theta} = 0$ at $\theta = \theta^*$, since θ^* is the minimum for L.

So gradient descent knows that $\theta^{(0)} \neq \theta^*$, so it needs to guess another value for θ, which will be $\theta^{(1)}$. But what should $\theta^{(1)}$ be?

Let's use the information that $\frac{\partial L}{\partial \theta}$ is large and positive at $\theta = \theta^{(0)}$. Since $\frac{\partial L}{\partial \theta}$ is positive, we know that L is increasing at $\theta = \theta^{(0)}$. This means that $\theta^* < \theta^{(0)}$, because functions increase after attaining their minimum value (we can say this based on the definition of what minimum is). Thus, we know that we need to "push back" our estimate for θ^* by choosing $\theta^{(1)}$ to be less than $\theta^{(0)}$.

→ this is where the - sign comes from in the update equation

But how much less than $\theta^{(0)}$ should $\theta^{(1)}$ be? Let's use the other piece of information we have about $\frac{\partial L}{\partial \theta}$ at $\theta = \theta^{(0)}$, that $\frac{\partial L}{\partial \theta}$ is large (in addition to positive)



Since $\frac{\partial L}{\partial \theta}$ at $\theta = \theta^{(0)}$ is large, we can say that $\theta^{(0)}$ is pretty far from θ^*. The reason for this relies on the interpretation of $\frac{\partial L}{\partial \theta}$ as the slope of the tangent line. If the slope of the tangent line is large and positive, then the function is increasing quickly. You can see this from $\frac{\partial L}{\partial \theta}\Big|_{\theta=\theta^{(0)}}$ on the graph. Since we are trying to find the value of θ that minimizes the function, we want to move away from values of θ where the function is increasing quickly.

Specifically, we know that when the slope of the tangent line is large, we need to change our θ by a lot. In general, we know that the larger our $\frac{\partial L}{\partial \theta}$, the more we need to change θ. The way the gradient descent update expresses this is by changing θ by an amount proportional to $\frac{\partial L}{\partial \theta}$:

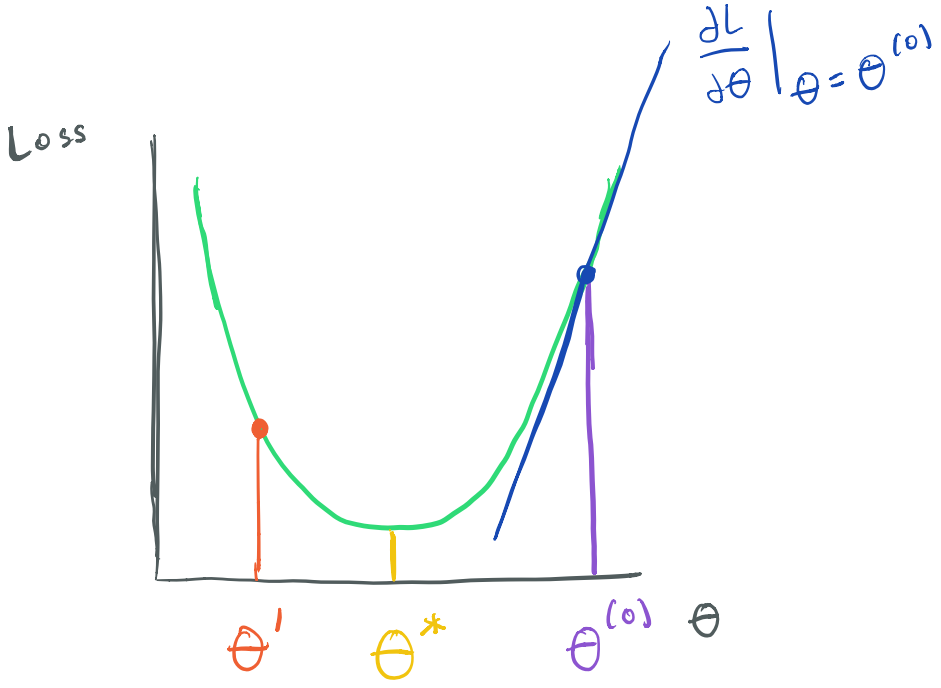
$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \alpha \left.\frac{\partial L}{\partial \theta}\right|_{\theta=\theta^{(t)}}$$

"push back" θ by an amount proportional to $\left.\frac{\partial L}{\partial \theta}\right|_{\theta=\theta^{(t)}}$

Let's stop and do a quick summary of what we've said so far. The reason the update rule has a minus sign is because when $\frac{\partial L}{\partial \theta}$ is positive, we've overshot θ^* and have to "push back" our estimate to be less than the previous estimate.

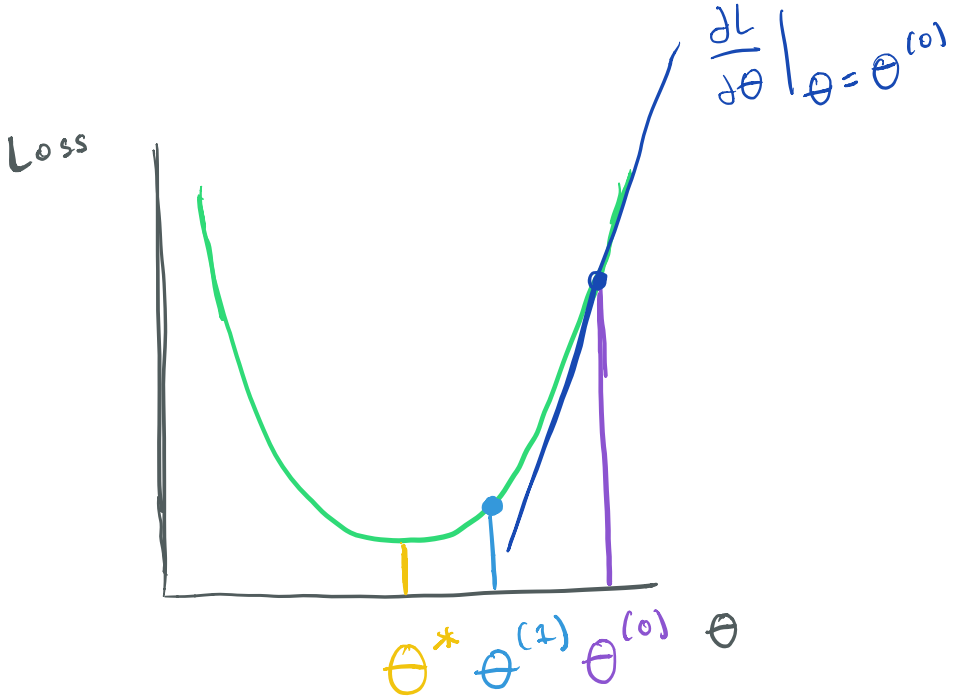
Additionally, we push back the estimate for θ^* by an amount proportional to $\frac{\partial L}{\partial \theta}$ because when $\frac{\partial L}{\partial \theta}$ is large, we know we are far from θ^*.

One last thing to take care of: what is the purpose/role of α?



When we push back $\theta^{(0)}$ in the graph above, depending on how large $\frac{\partial L}{\partial \theta}\Big|_{\theta=\theta^{(0)}}$ is, we might push θ back to a value like θ', which overshoots θ^* on the other side.

To avoid this problem of overshooting θ^*, we multiply $\frac{\partial L}{\partial \theta}$ by a fraction α so we don't push our estimate by the full magnitude of $\frac{\partial L}{\partial \theta}$. Keep in mind $0 < \alpha < 1$ (think about why). With this fraction α, we push $\theta^{(0)}$ to a value like $\theta^{(1)}$.



All of what we've said so far happens each time θ is updated. But how do we know how many times to update θ?

We continue updating θ until $\theta^{(t+1)} = \theta^{(t)}$.

This is because, looking at the update equation, the only way for $\theta^{(t+1)} = \theta^{(t)}$ is if $\frac{\partial L}{\partial \theta}\Big|_{\theta=\theta^{(t)}} = 0$. But if $\frac{\partial L}{\partial \theta}\Big|_{\theta=\theta^{(t)}} = 0$, then $\theta^{(t)}$ is the minimum! Thus, $\theta^{(t)} = \theta^*$ and we've found the optimal θ^*.