Basic Idea behind Gradient Descent

Goal: Find parameter values that minimize the loss function

$$w^* = \underset{w}{\operatorname{argmin}}\, L(w)$$

$L(w)$ = Loss function

w = Parameters

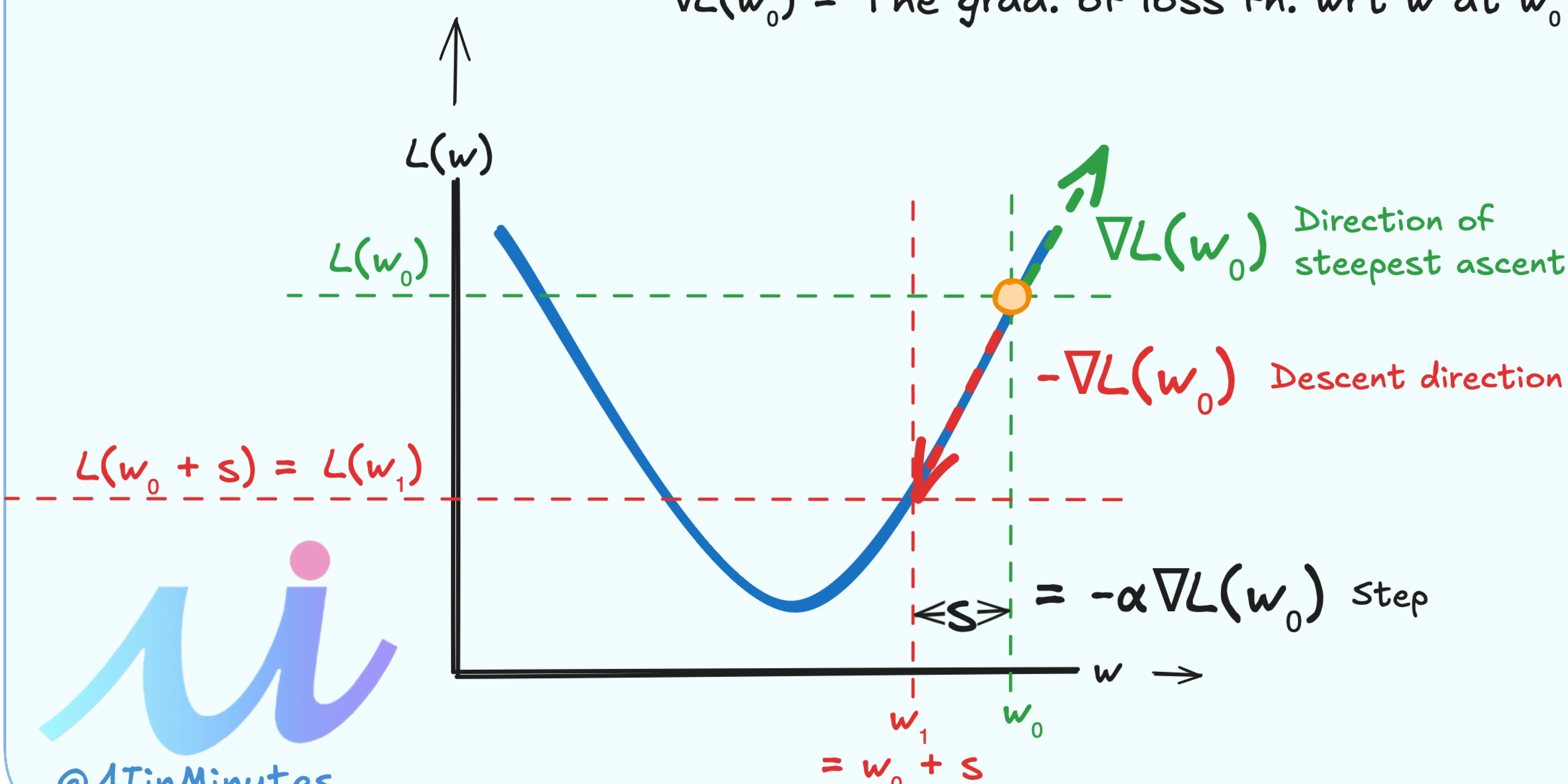
Idea: Initialize parameter values; iteratively update them in the direction opposite to the gradient

$$w_1 = w_0 + s$$

$$s = -\alpha \nabla L(w_0)$$

$$w_1 = w_0 - \alpha \nabla L(w_0)$$

w_1 = New estimate
w_0 = Initialized/previous estimate
$L(w_1)$ = Loss at new estimate
$L(w_0)$ = Loss at inti/prev. estimate
s = Step
α = Learning rate or step size (>0)
$\nabla L(w_0)$ = The grad. of loss fn. wrt w at w_0



@AIInMinutes

Can we be sure this works? Time to bring Taylor back!

Key point: The loss function $L(w)$ can be approximated using the first order Taylor series expansion at w_0, which should be valid for a sufficiently small step s, and

$$L(w_1) = L(w_0 + s) \approx L(w_0) + s^T \nabla L(w_0)$$

if s is sufficiently small
(so that the linear approximation holds),
we can prove that $L(w_1) = L(w_0 + s) \leq L(w_0)$

put $s = -\alpha \nabla L(w_0)$ in

$$L(w_1) \approx L(w_0) - \alpha \boxed{\nabla L(w_0)^T \nabla L(w_0)}$$

What is this?

positive by design
used to control
the size of s

always non-negative

If x is a vector,
$x^T x$ represents
the square of its length,
which is always non-negative.
$\nabla L(w_0)^T \nabla L(w_0)$ is the square
of the length of the grad.

$$L(w_1) \approx L(w_0) - \boxed{\alpha \nabla L(w_0)^T \nabla L(w_0)}$$

Always non-negative
as long as appx. holds

$$L(w_1) \leq L(w_0)$$

The loss function is smaller (or equal) at the new parameter estimate.

This is done iteratively.

$$w_{t+1} = w_t - \alpha \nabla L(w_t)$$

@AIinMinutes