Gradient Descent Formalized

> ☆ **Key Points**
>
> Use a Taylor expansion to show that stepping in the direction of steepest descent always reduces the loss function.
>
> Setting the learning rate parameter α too high prevents the algorithm from converging.
>
> A popular option is to decrease the step-size α with each step

In this section, our goal is to minimize a convex continuous and differentiable loss function $\ell(\mathbf{w})$. One way to achieve this is using gradient descent (aka "steepest descent"). By Taylor's expansion, we can approximate $\ell(\mathbf{w}+\mathbf{s})$, which is the function value after we take a small update from our previous location $\mathbf{w}$ along the direction of $\mathbf{s}$ (but note this only works if the magnitude of $\mathbf{s}$, i.e., the step size, is small):

$$\ell(\mathbf{w}+\mathbf{s}) \approx \ell(\mathbf{w}) + g(\mathbf{w})^\top \mathbf{s}$$

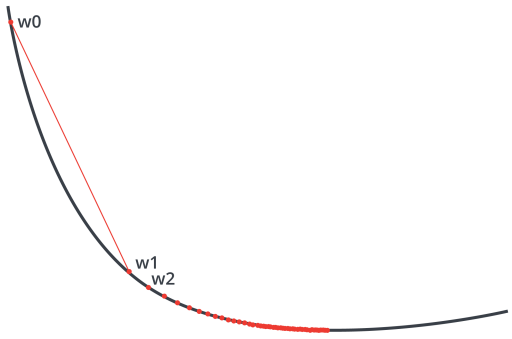
Where $g(\mathbf{w}) = \nabla\ell(\mathbf{w})$ is the gradient of the function $\ell(\mathbf{w})$.

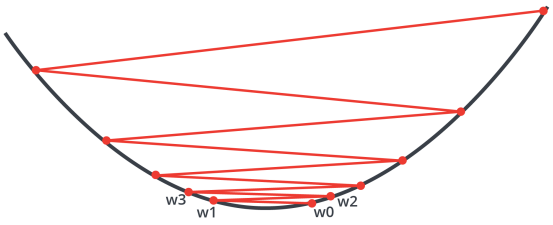
Essentially, we assume that $\ell(\mathbf{w})$ behaves linearly around $\mathbf{w}$ and we want to take a step $\mathbf{s}$ so that we can that we obtain $\ell(\mathbf{w}+\mathbf{s})$ smaller than $\ell(\mathbf{w})$ -- thus minimizing ℓ with each iteration. In gradient descent, we set $\mathbf{s} = -\alpha g(\mathbf{w})$, for some small $\alpha > 0$. It is straightforward to prove that taking a step along the direction $\mathbf{s}$ reduces the loss value, namely, $\ell(\mathbf{w}+\mathbf{s}) < \ell(\mathbf{w})$.

$$\underbrace{\ell(\mathbf{w}+(-\alpha g(\mathbf{w})))}_{\text{after one update}} \approx \ell(\mathbf{w}) + g(\mathbf{w})^\top(-\alpha g(\mathbf{w})) = \ell(\mathbf{w}) - \underbrace{\overbrace{\alpha g(\mathbf{w})^\top g(\mathbf{w})}^{>0}}_{>0} < \underbrace{\ell(\mathbf{w})}_{\text{before}}$$

Note: Setting the learning rate α is not always an exact science. This proof only holds if the learning rate is sufficiently small such that the gradient descent will converge. (See the first figure below.) If it is too large, Taylor's approximation no longer holds and the algorithm can easily *diverge* out of control. (See the second figure below.) A safe (but sometimes slow) choice is to set $\alpha = \frac{t_0}{t}$, which guarantees that it will eventually become small enough to converge (for any initial value $t_0 > 0$).

Gradient Descent Convergence **Gradient Descent Divergence**


w0
w1
w2


w3
w1
w0
w2