(https://google.com/racialequity?authuser=0)

# Reducing Loss: Learning Rate

**ated Time:** 5 minutes

As noted, the gradient vector has both a direction and a magnitude. Gradient descent algorithms multiply the gradient by a scalar known as the **learning rate** (also sometimes called **step size**) to determine the next point. For example, if the gradient magnitude is 2.5 and the learning rate is 0.01, then the gradient descent algorithm will pick the next point 0.025 away from the previous point.

**Hyperparameters** are the knobs that programmers tweak in machine learning algorithms. Most machine learning programmers spend a fair amount of time tuning the learning rate. If you pick a learning rate that is too small, learning will take too long:

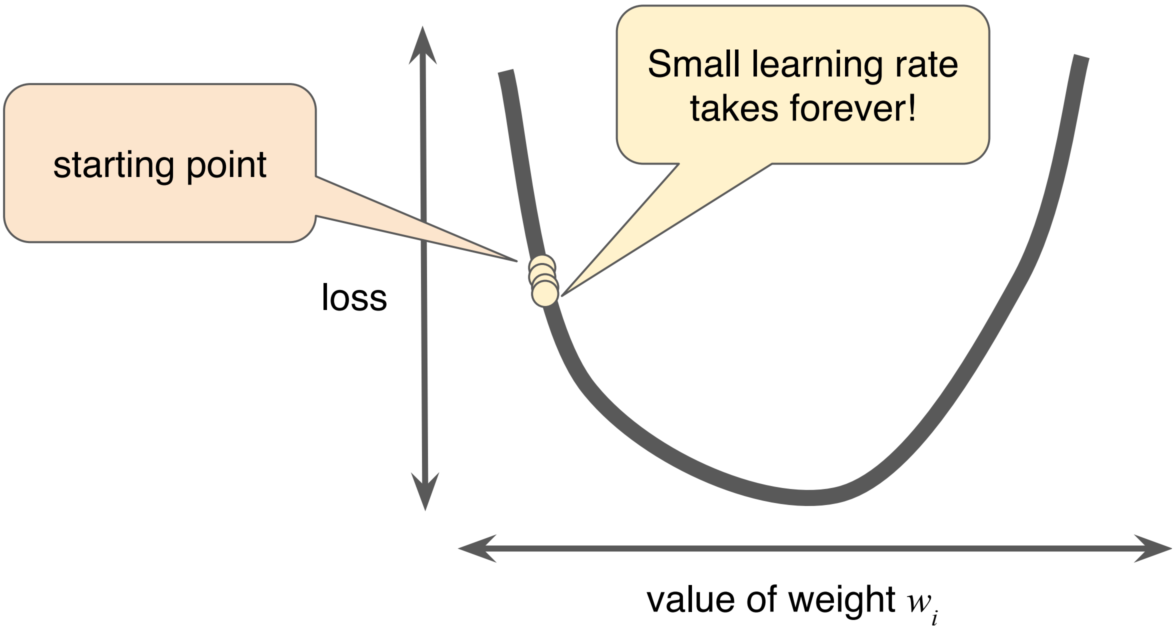


**Figure 6. Learning rate is too small.**

Conversely, if you specify a learning rate that is too large, the next point will perpetually bounce haphazardly across the bottom of the well like a quantum mechanics experiment gone horribly wrong:

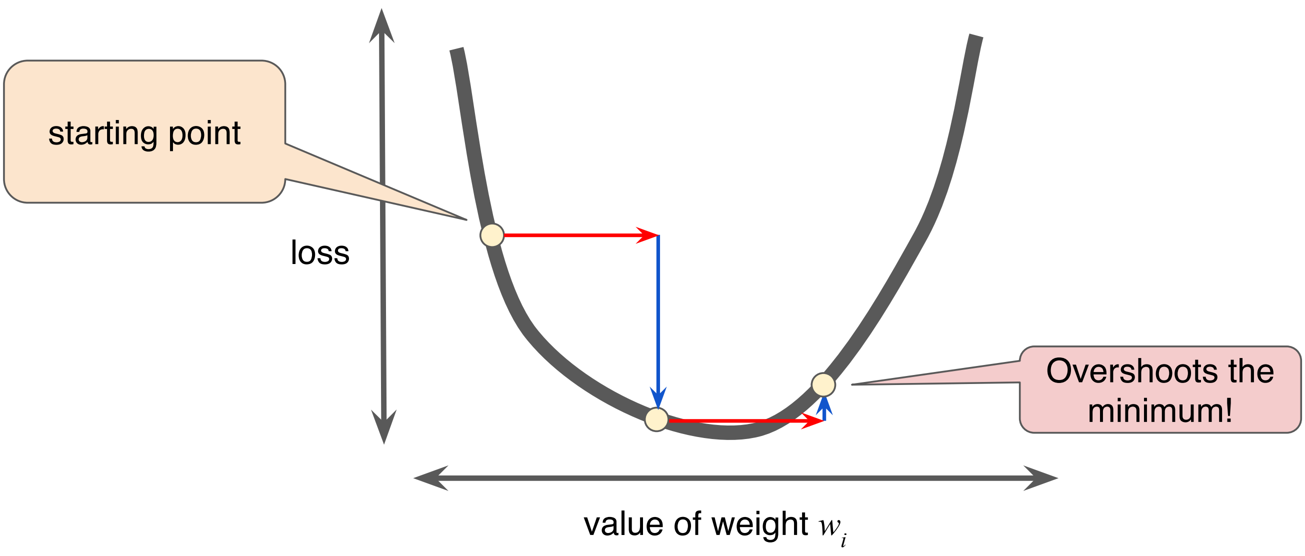


**Figure 7. Learning rate is too large.**

There's a Goldilocks (https://wikipedia.org/wiki/Goldilocks_principle) learning rate for every regression problem. The Goldilocks value is related to how flat the loss function is. If you know the gradient of the loss function is small then you can safely try a larger learning rate, which compensates for the small gradient and results in a larger step size.

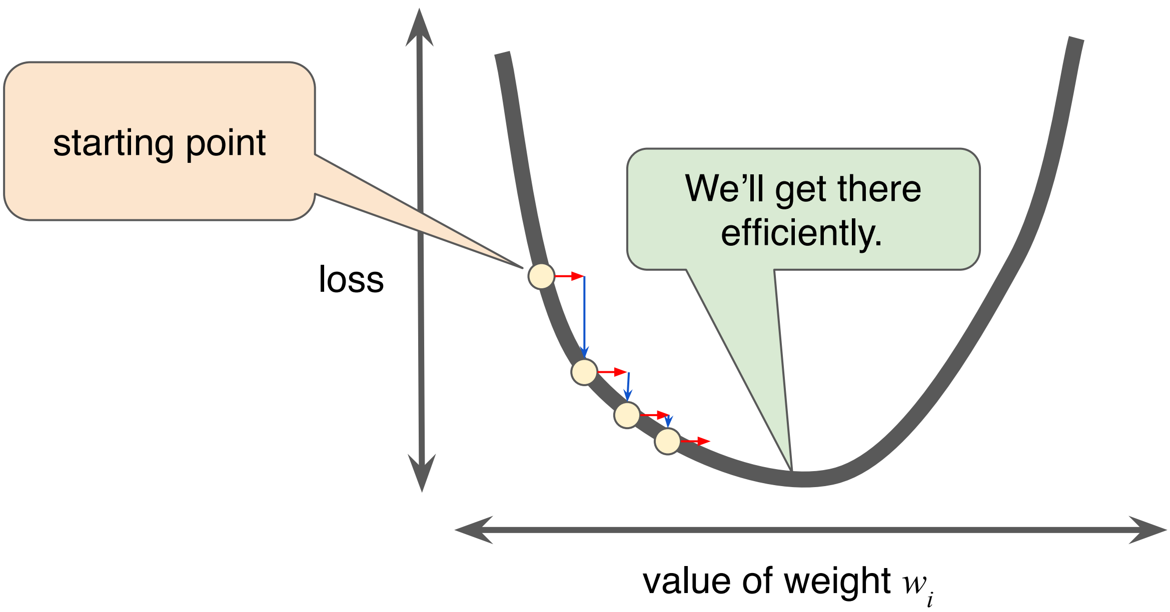


**Figure 8. Learning rate is just right.**

+ **Click the plus icon to learn more about the ideal learning rate.**

The ideal learning rate in one-dimension is $\frac{1}{f(x)''}$ (the inverse of the second derivative of f(x) at x).

The ideal learning rate for 2 or more dimensions is the inverse of the Hessian (https://wikipedia.org/wiki/Hessian_matrix) (matrix of second partial derivatives).

The story for general convex functions is more complex.

**rms**

- perparameter
 ://developers.google.com/machine-
 ıg/glossary?authuser=0#hyperparameter)
- ɛp size
 ://developers.google.com/machine-
 ıg/glossary?authuser=0#step_size)
- learning rate
 (https://developers.google.com/machine-learning/glossary?authuser=0#learning_rate)

Help Center (https://support.google.com/machinelearningeducation?authuser=0)

Previous
← Gradient Descent
(https://developers.google.com/machine-learning/crash-course/reducing-loss/gradient-descent?authuser=0)

Next
Optimizing Learning Rate →
(https://developers.google.com/machine-learning/crash-course/fitter/graph?authuser=0)



Last updated 2020-02-10 UTC.