

of course, the additional epochs usually provide enough benefit due to decreased training error to offset the harm they cause by increasing the gap between training error and test error.

With some datasets growing rapidly in size, faster than computing power, it is becoming more common for machine learning applications to use each training example only once or even to make an incomplete pass through the training set. When using an extremely large training set, overfitting is not an issue, so underfitting and computational efficiency become the predominant concerns. See also [Bottou and Bousquet \(2008\)](#) for a discussion of the effect of computational bottlenecks on generalization error, as the number of training examples grows.

## 8.2 Challenges in Neural Network Optimization

Optimization in general is an extremely difficult task. Traditionally, machine learning has avoided the difficulty of general optimization by carefully designing the objective function and constraints to ensure that the optimization problem is convex. When training neural networks, we must confront the general non-convex case. Even convex optimization is not without its complications. In this section, we summarize several of the most prominent challenges involved in optimization for training deep models.

### 8.2.1 Ill-Conditioning

Some challenges arise even when optimizing convex functions. Of these, the most prominent is ill-conditioning of the Hessian matrix  $\mathbf{H}$ . This is a very general problem in most numerical optimization, convex or otherwise, and is described in more detail in section [4.3.1](#).

The ill-conditioning problem is generally believed to be present in neural network training problems. Ill-conditioning can manifest by causing SGD to get “stuck” in the sense that even very small steps increase the cost function.

Recall from equation [4.9](#) that a second-order Taylor series expansion of the cost function predicts that a gradient descent step of  $-\epsilon \mathbf{g}$  will add

$$\frac{1}{2} \epsilon^2 \mathbf{g}^\top \mathbf{H} \mathbf{g} - \epsilon \mathbf{g}^\top \mathbf{g} \quad (8.10)$$

to the cost. Ill-conditioning of the gradient becomes a problem when  $\frac{1}{2} \epsilon^2 \mathbf{g}^\top \mathbf{H} \mathbf{g}$  exceeds  $\epsilon \mathbf{g}^\top \mathbf{g}$ . To determine whether ill-conditioning is detrimental to a neural network training task, one can monitor the squared gradient norm  $\mathbf{g}^\top \mathbf{g}$  and