where $\| \theta \|_2$ and $\| \theta \|_1$ denote the L^1 and L^2 penalties, respectively. From personal experience, these penalties tend to be less useful for deep models than dropout and early stopping. Some practitioners still make use of weight regularization, so it's worth understanding how to apply these penalties when tuning deep networks.

Training Fully Connected Networks

Training fully connected networks requires a few tricks beyond those you have seen so far in this book. First, unlike in the previous chapters, we will train models on larger datasets. For these datasets, we will show you how to use minibatches to speed up gradient descent. Second, we will return to the topic of tuning learning rates.

Minibatching

For large datasets (which may not even fit in memory), it isn't feasible to compute gradients on the full dataset at each step. Rather, practitioners often select a small chunk of data (typically 50–500 datapoints) and compute the gradient on these datapoints. This small chunk of data is traditionally called a minibatch.

In practice, minibatching seems to help convergence since more gradient descent steps can be taken with the same amount of compute. The correct size for a minibatch is an empirical question often set with hyperparameter tuning.

Learning rates

The learning rate dictates the amount of importance to give to each gradient descent step. Setting a correct learning rate can be tricky. Many beginning deep-learners set learning rates incorrectly and are surprised to find that their models don't learn or start returning NaNs. This situation has improved significantly with the development of methods such as ADAM that simplify choice of learning rate significantly, but it's worth tweaking the learning rate if models aren't learning anything.

Implementation in TensorFlow

In this section, we will show you how to implement a fully connected network in TensorFlow. We won't need to introduce many new TensorFlow primitives in this section since we have already covered most of the required basics.

Installing DeepChem

In this section, you will use the DeepChem machine learning toolchain for your experiments (full disclosure: one of the authors was the creator of DeepChem). Detailed installation directions for DeepChem can be found online, but briefly the Anaconda installation via the `conda` tool will likely be most convenient.