Playing Tic-Tac-Toe

Tic-tac-toe is a simple two-player game. Players place Xs and Os on a 3 × 3 game board until one player succeeds in placing three of her pieces in a row. The first player to do so wins. If neither player succeeds in obtaining three in a row before the board is filled up, the game ends in a draw. Figure 8-6 illustrates a tic-tac-toe game board.

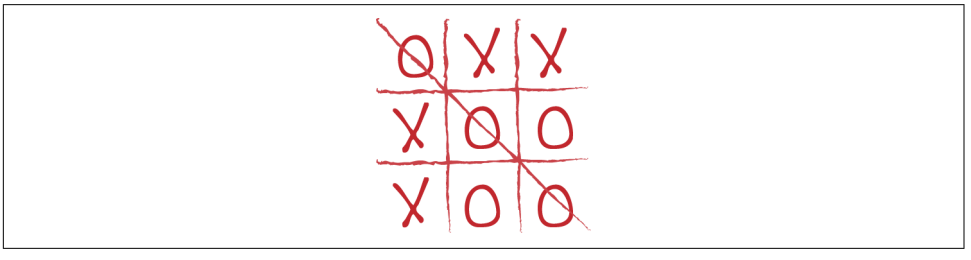

Figure 8-6. A tic-tac-toe game board.

Tic-tac-toe is a nice testbed for reinforcement learning techniques. The game is simple enough that exorbitant amounts of computational power aren't required to train effective agents. At the same time, despite tic-tac-toe's simplicity, learning an effective agent requires considerable sophistication. The TensorFlow code for this section is arguably the most sophisticated example found in this book. We will walk you through the design of a TensorFlow tic-tac-toe asynchronous reinforcement learning agent in the remainder of this section.

Object Orientation

The code we've introduced thus far in this book has primarily consisted of scripts augmented by smaller helper functions. In this chapter, however, we will swap to an object-oriented programming style. This style of programming might be new to you, especially if you hail from the scientific world rather than from the tech world. Briefly, an object-oriented program defines *objects* that model aspects of the world. For example, you might want to define `Environment` or `Agent` or `Reward` objects that directly correspond to these mathematical concepts. A *class* is a template for objects that can be used to *instantiate* (or create) many new objects. For example, you will shortly see an `Environment` class definition we will use to define many particular `Environment` objects.

Object orientation is particularly powerful for building complex systems, so we will use it to simplify the design of our reinforcement learning system. In practice, your real-world deep learning (or reinforcement learning) systems will likely need to be object oriented as well, so we encourage taking some time to master object-oriented design. There are many superb books that cover the fundamentals of object-oriented design, and we recommend that you check them out as necessary.