Challenge for the Reader

We strongly encourage you to try training tic-tac-toe models for yourself! Note that this example is more involved than other examples in the book, and will require greater computational power. We recommend a machine with at least a few CPU cores. This requirement isn't too onerous; a good laptop should suffice. Try using a tool like `htop` to check that the code is indeed multithreaded. See how good a model you can train! You should be able to beat the random baseline most of the time, but this basic implementation won't give you a model that always wins. We recommend exploring the RL literature and expanding upon the base implementation to see how well you can do.

Review

In this chapter, we introduced you to the core concepts of reinforcement learning (RL). We walked you through some recent successes of RL methods on ATARI, upside-down helicopter flight, and computer Go. We then taught you about the mathematical framework of Markov decision processes. We brought it together with a detailed case study walking you through the construction of a tic-tac-toe agent. This algorithm uses a sophisticated training method, A3C, that makes use of multiple CPU cores to speed up training. In Chapter 9, you'll learn more about training models with multiple GPUs.

CHAPTER 9

Training Large Deep Networks

Thus far, you have seen how to train small models that can be completely trained on a good laptop computer. All of these models can be run fruitfully on GPU-equipped hardware with notable speed boosts (with the notable exception of reinforcement learning models for reasons discussed in the previous chapter). However, training larger models still requires considerable sophistication. In this chapter, we will discuss various types of hardware that can be used to train deep networks, including graphics processing units (GPUs), tensor processing units (TPUs), and neuromorphic chips. We will also briefly cover the principles of distributed training for larger deep learning models. We end the chapter with an in-depth case study, adapated from one of the TensorFlow tutorials, demonstrating how to train a CIFAR-10 convolutional neural network on a server with multiple GPUs. We recommend that you attempt to try running this code yourself, but readily acknowledge that gaining access to a multi-GPU server is trickier than finding a good laptop. Luckily, access to multi-GPU servers on the cloud is becoming possible and is likely the best solution for industrial users of TensorFlow seeking to train large models.

Custom Hardware for Deep Networks

As you've seen throughout the book, deep network training requires chains of tensorial operations performed repeatedly on minibatches of data. Tensorial operations are commonly transformed into matrix multiplication operations by software, so rapid training of deep networks fundamentally depends on the ability to perform matrix multiplication operations rapidly. While CPUs are perfectly capable of implementing matrix multiplications, the generality of CPU hardware means much effort will be wasted on overhead unneeded for mathematical operations.