Limitations of TensorFlow

One of the major current weaknesses of TensorFlow is that constructing a new deep learning architecture is relatively slow (on the order of multiple seconds to initialize an architecture). As a result, it's not convenient in TensorFlow to construct some sophisticated deep architectures that change their structure dynamically. One such architecture is the TreeLSTM, which uses syntactic parse trees of English sentences to perform tasks that require understanding of natural language. Since each sentence has a different parse tree, each sentence requires a slightly different architecture. Figure 1-15 illustrates the TreeLSTM architecture.

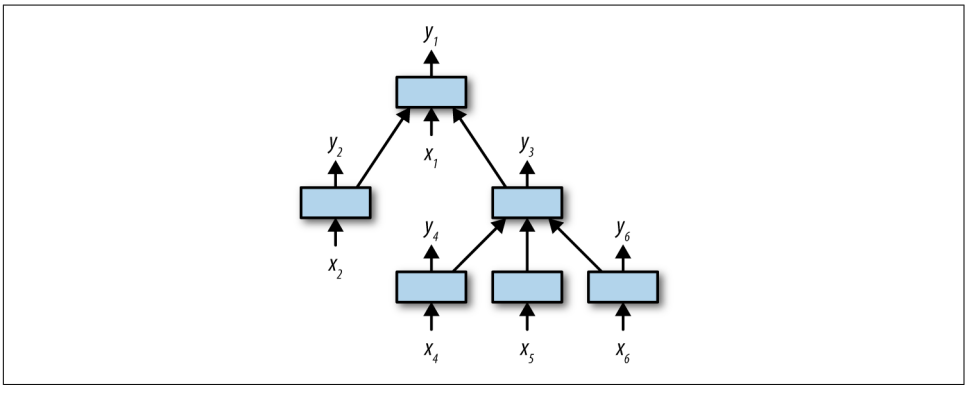

Figure 1-15. A conceptual depiction of a TreeLSTM architecture. The shape of the tree is different for each input datapoint, so a different computational graph must be constructed for each example.

While such models can be implemented in TensorFlow, doing so requires significant ingenuity due to the limitations of the current TensorFlow API. New frameworks such as Chainer, DyNet, and PyTorch promise to remove these barriers by making the construction of new architectures lightweight enough so that models like the TreeLSTM can be constructed easily. Luckily, TensorFlow developers are already working on extensions to the base TensorFlow API (such as TensorFlow Eager) that will enable easier construction of dynamic architectures.

One takeaway is that progress in deep learning frameworks is rapid, and today's novel system can be tomorrow's old news. However, the fundamental principles of the underlying tensor calculus date back centuries, and will stand readers in good stead regardless of future changes in programming models. This book will emphasize using TensorFlow as a vehicle for developing an intuitive knowledge of the underlying tensor calculus.

Review

In this chapter, we've explained why deep learning is a subject of critical importance for the modern software engineer and taken a whirlwind tour of a number of deep architectures. In the next chapter, we will start exploring TensorFlow, Google's framework for constructing and training deep architectures. In the chapters after that, we will dive deep into a number of practical examples of deep architectures.

Machine learning (and deep learning in particular), like much of computer science, is a very empirical discipline. It's only really possible to understand deep learning through significant practical experience. For that reason, we've included a number of in-depth case studies throughout the remainder of this book. We encourage you to delve into these examples and to get your hands dirty experimenting with your own ideas using TensorFlow. It's never enough to understand algorithms only theoretically!