

viewed as assigning individual entries of a multidimensional array, when provided indices into the array as arguments.

We won't use this more mathematical definition much in this book, but it serves as a useful bridge to connect the deep learning concepts you will learn about with the centuries of mathematical research that have been undertaken on tensors by the physics and mathematics communities.



Covariance and Contravariance

Our definition here has swept many details under the rug that would need to be carefully attended to for a formal treatment. For example, we don't touch upon the notion of covariant and contravariant indices here. What we call a rank- n tensor is better described as a (p, q) -tensor where $n = p + q$ and p is the number of contravariant indices, and q the number of covariant indices. Matrices are $(1,1)$ -tensors, for example. As a subtlety, there are rank-2 tensors that are not matrices! We won't dig into these topics carefully here since they don't crop up much in machine learning, but we encourage you to understand how covariance and contravariance affect the machine learning systems you construct.

Basic Computations in TensorFlow

We've spent the last sections covering the mathematical definitions of various tensors. It's now time to cover how to create and manipulate tensors using TensorFlow. For this section, we recommend you follow along using an interactive Python session (with IPython). Many of the basic TensorFlow concepts are easiest to understand after experimenting with them directly.

Installing TensorFlow and Getting Started

Before continuing this section, you will need to install TensorFlow on your machine. The details of installation will vary depending on your particular hardware, so we refer you to [the official TensorFlow documentation](#) for more details.

Although there are frontends to TensorFlow in multiple programming languages, we will exclusively use the TensorFlow Python API in the remainder of this book. We recommend that you install [Anaconda Python](#), which packages many useful numerical libraries along with the base Python executable.

Once you've installed TensorFlow, we recommend that you invoke it interactively while you're learning the basic API (see [Example 2-1](#)). When experimenting with TensorFlow interactively, it's convenient to use `tf.InteractiveSession()`. Invoking this statement within IPython (an interactive Python shell) will make TensorFlow