2.4.4   TensorFlow

TensorFlow is an open-source system developed initially at Google to support machine-learning applications. Like Spark, TensorFlow provides a programming interface in which one writes a sequence of steps. Programs are typically acyclic, although like Spark it is possible to iterate blocks of code.

One major difference between Spark and TensorFlow is the type of data that is passed between steps of the program. In place of the RDD, TensorFlow uses *tensors*; a tensor is simply a multidimensional matrix.

Example 2.11 : A constant, e.g. 3.14159, is regarded as a 0-dimensional tensor. A vector is a 1-dimensional tensor. For instance, the vector $(1, 2, 3)$ can be written in tensorFlow as $[1., 2., 3.]$. A matrix is a 2-dimensional tensor. For example, the matrix

$$\begin{matrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{matrix}$$

is expressed as $[[1., 2., 3., 4.], \ [5., 6., 7., 8.], \ [9., 10., 11., 12.]]$.

Higher-dimensional arrays are possible as well. For instance, a 2-by-2-by-2 cube of 0's is represented as $[[[0., 0.], \ [0., 0.]], \ [[0., 0.], \ [0., 0.]]]$.   □

Although tensors are in fact a restricted form of RDD, the power of TensorFlow comes from its selection of built-in operations. Linear algebra operations are available as functions. For example, if you want matrix C to be the product of matrices A and B, you can write

```
C = tensorflow.matmul(A,B)
```

Even more powerful are the common approaches to machine learning that are built in as operations. a single statement in the TensorFlow language can cause a model that is a tensor to be constructed from training data, which is also represented as a tensor, using a method like gradient descent. (We discuss gradient descent in Sections 9.4.5 and 12.3.4).