Deep Learning Primitives

Most deep architectures are built by combining and recombining a limited set of architectural primitives. Such primitives, typically called neural network layers, are the foundational building blocks of deep networks. In the rest of this book, we will provide in-depth introductions to such layers. However, in this section, we will provide a brief overview of the common modules that are found in many deep networks. This section is not meant to provide a thorough introduction to these modules. Rather, we aim to provide a rapid overview of the building blocks of sophisticated deep architectures to whet your appetite. The art of deep learning consists of combining and recombining such modules and we want to show you the alphabet of the language to start you on the path to deep learning expertise.

Fully Connected Layer

A fully connected network transforms a list of inputs into a list of outputs. The transformation is called fully connected since any input value can affect any output value. These layers will have many learnable parameters, even for relatively small inputs, but they have the large advantage of assuming no structure in the inputs. This concept is illustrated in Figure 1-1.

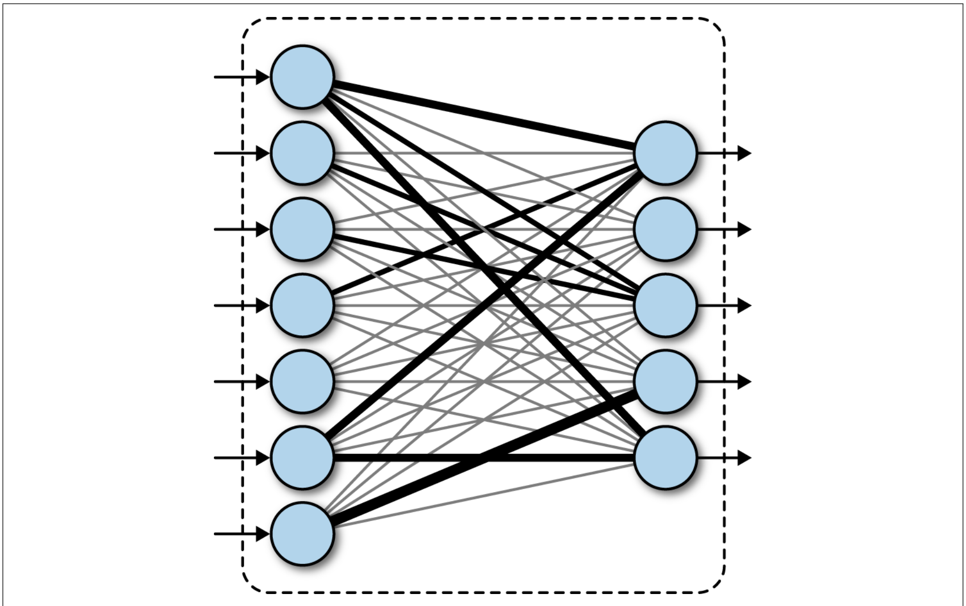

Figure 1-1. A fully connected layer. Inbound arrows represent inputs, while outbound arrows represent outputs. The thickness of interconnecting lines represents the magnitude of learned weights. The fully connected layer transforms inputs into outputs via the learned rule.

Convolutional Layer

A convolutional network assumes special spatial structure in its input. In particular, it assumes that inputs that are close to each other spatially are semantically related. This assumption makes most sense for images, since pixels close to one another are likely semantically linked. As a result, convolutional layers have found wide use in deep architectures for image processing. This concept is illustrated in Figure 1-2.

Just like fully connected layers transform lists to lists, convolutional layers transform images into images. As a result, convolutional layers can be used to perform complex image transformations, such as applying artistic filters to images in photo apps.

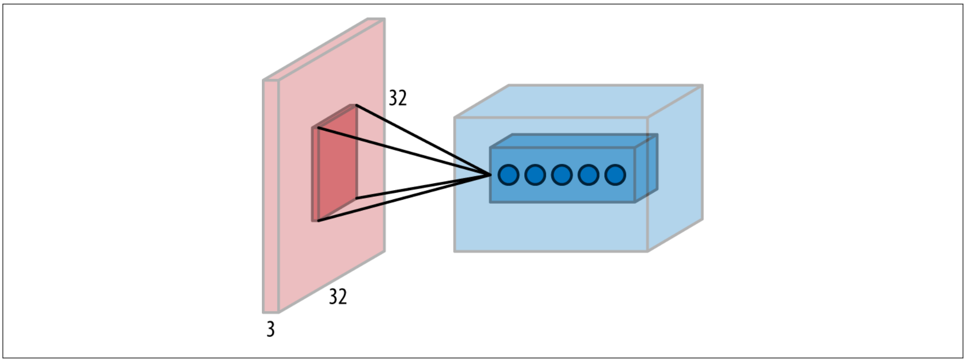

Figure 1-2. A convolutional layer. The red shape on the left represents the input data, while the blue shape on the right represents the output. In this particular case, the input is of shape (32, 32, 3). That is, the input is a 32-pixel-by-32-pixel image with three RGB color channels. The highlighted region in the red input is a "local receptive field," a group of inputs that are processed together to create the highlighted region in the blue output.

Recurrent Neural Network Layers

Recurrent neural network (RNN) layers are primitives that allow neural networks to learn from sequences of inputs. This layer assumes that the input evolves from step to step following a defined update rule that can be learned from data. This update rule presents a prediction of the next state in the sequence given all the states that have come previously. An RNN is illustrated in Figure 1-3.

An RNN layer can learn this update rule from data. As a result, RNNs are very useful for tasks such as language modeling, where engineers seek to build systems that can predict the next word users will type from history.