Before passing an input into the FCN, the image matrix will have to be flattened. For example, a 28 x 28 x 3 image matrix will become 2352 input weights plus a bias of 1 into the fully connected network.

In the case of our convolutional network, the feature maps of either the convolutional or pooling layer are flattened before passing into the FCN to compute the final network probabilities using the softmax function.

An Example CNN Architecture

We have discussed the building blocks of a convolutional neural network system. As you've seen, a CNN system is principally composed of convolution layers, pooling layers, and the fully connected layer. However, the way these layers are arranged and in what number are down to the preferred heuristics of the particular use case that a CNN is employed in solving.

An example CNN modeling pipeline is shown here:

1. The first layer following the input layer of images must be a convolutional layer for extracting image features. A 3 x 3 image filter is commonly used depending on the size of the input image.
2. Pooling layers typical follow a set of one or more convolutional layers. Typically, a 2 x 2 filter size is used in the pooling layer.
3. The fully connected layer must be the final layer of the CNN. It is also called the dense layer. It contains the softmax activation function to give the probabilities of class membership.
4. CNN may include one or more Dropout layers to prevent the network from overfitting.

Figure 35-15 is an example of a CNN architecture.

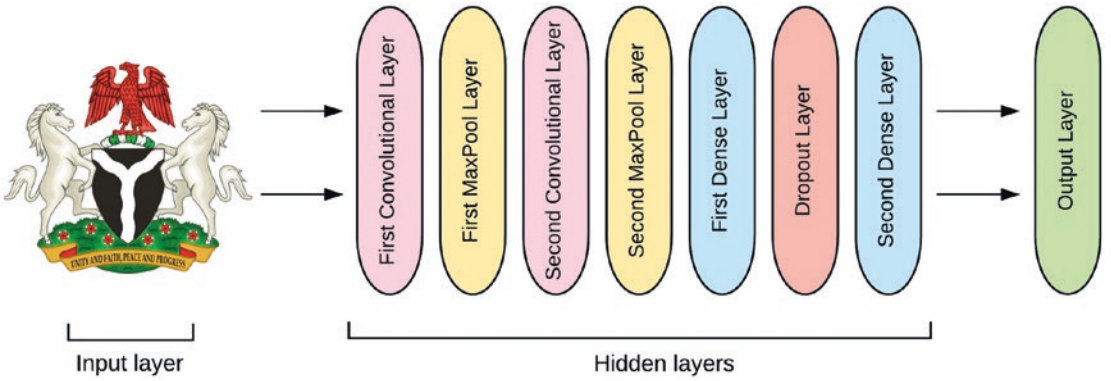

Figure 35-15. *CNN architecture*

CNN for Image Recognition with TensorFlow 2.0

In this example, we will build a convolutional neural network (CNN) to classify images from the CIFAR-10 dataset. CIFAR-10 is another standard image classification dataset to classify a colored 32 x 32 pixel image data into ten image classes, namely, airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The focus of this section is exclusively on using TensorFlow 2.0 methods to build a CNN image classifier.

The CNN model architecture implemented loosely mirrors the Krizhevsky's architecture, also known as AlexNet. The network architecture has the following layers:

- Convolution layer: kernel_size = [5 x 5]
- Convolution layer: kernel_size = [5 x 5]
- Batch normalization layer
- Convolution layer: kernel_size = [5 x 5]
- Max pooling: pool size = [2 x 2]
- Convolution layer: kernel_size = [5 x 5]
- Convolution layer: kernel_size = [5 x 5]
- Batch normalization layer
- Max pooling: pool size = [2 x 2]