The Pooling Layer

Pooling layers typically follow one or more convolutional layers. The goal of the pooling layer is to reduce or downsample the feature map of the convolutional layer. The pooling layer summarizes the image features learned in the previous network layers. By doing so, it also helps prevent the network from overfitting. Moreso, the reduction in the input size also bodes well for processing and memory costs when training the network.

The pooling layer can be seen as an aggregation function that consolidates learned features and extracts the essential features from previous layers. It does not conduct any multiplicative transformation on the input feature maps as seen in the convolutional layer.

The aggregation functions carried out by the pooling layer include max, sum, and average. The most frequently used aggregation function in practice is the max and is commonly called the MaxPool.

The aggregation functions of the pooling layer serve as the layers' filters. Just like the filters of the convolutional layer, they have a receptive field (although smaller in size than that of the convolutional layer) and a stride width. Howbeit, the filters which are the neurons of the pooling layer have no weight or biases. A typical size for the pooling filter is a 2 x 2 matrix as shown in Figure 35-13.

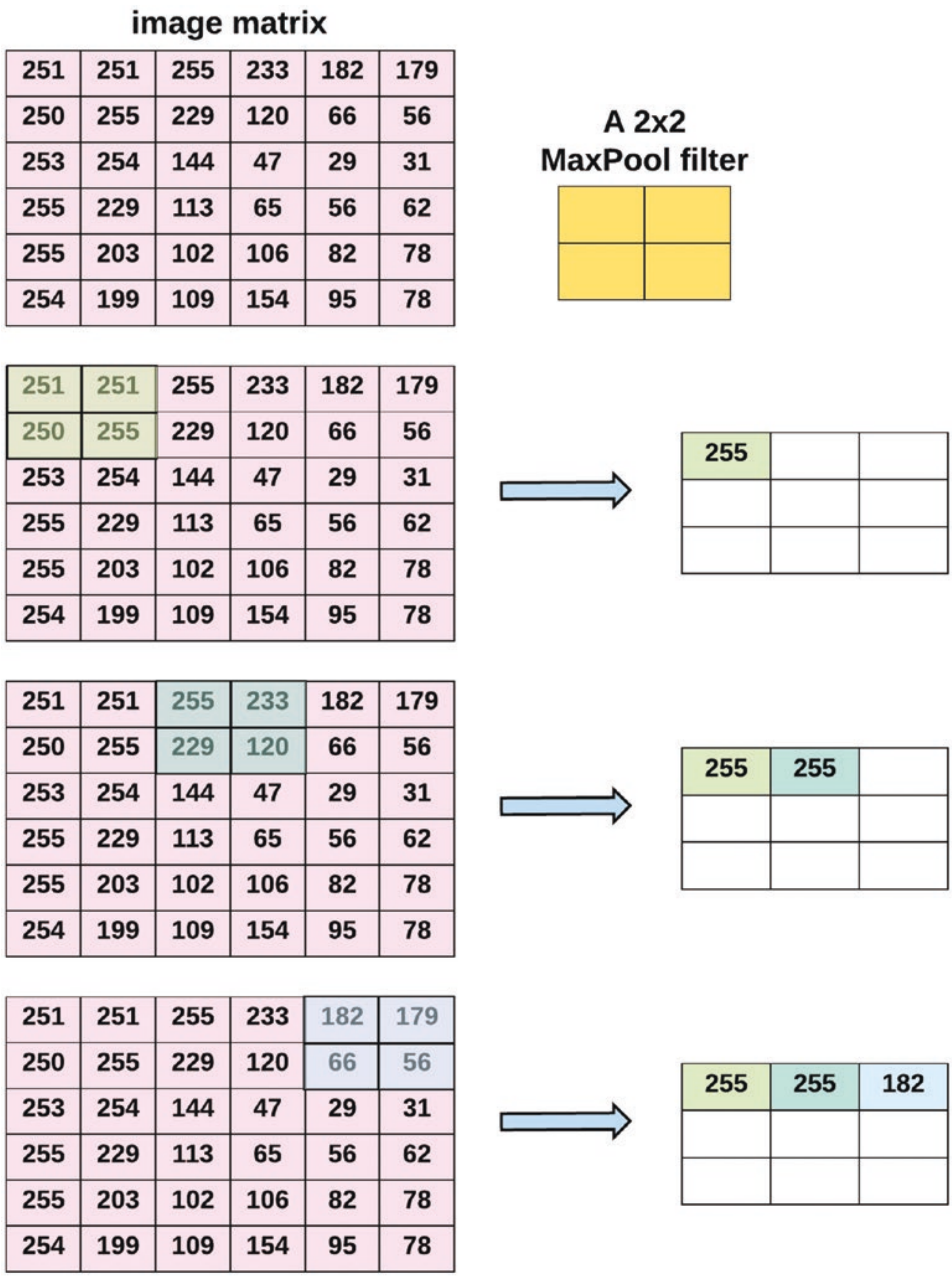

Figure 35-13. *Example of pooling with MaxPooling*

The essential advantage of the pooling layer is its ability to inject location invariance into the network. Location invariance means that features can be detected by the network no matter where they are on the image.

The pooling layer applies its aggregation function to all the channels of the input image. For example, in an R, G, B image (i.e., an image with three channels, red, green, and blue), the MaxPool will be applied independently to all the three channels. Similarly, for feature maps with a particular depth, the pooling aggregation will be applied separately to each feature map. See Figure 35-14 as an example of applying pooling to the channel depth of its inputs.

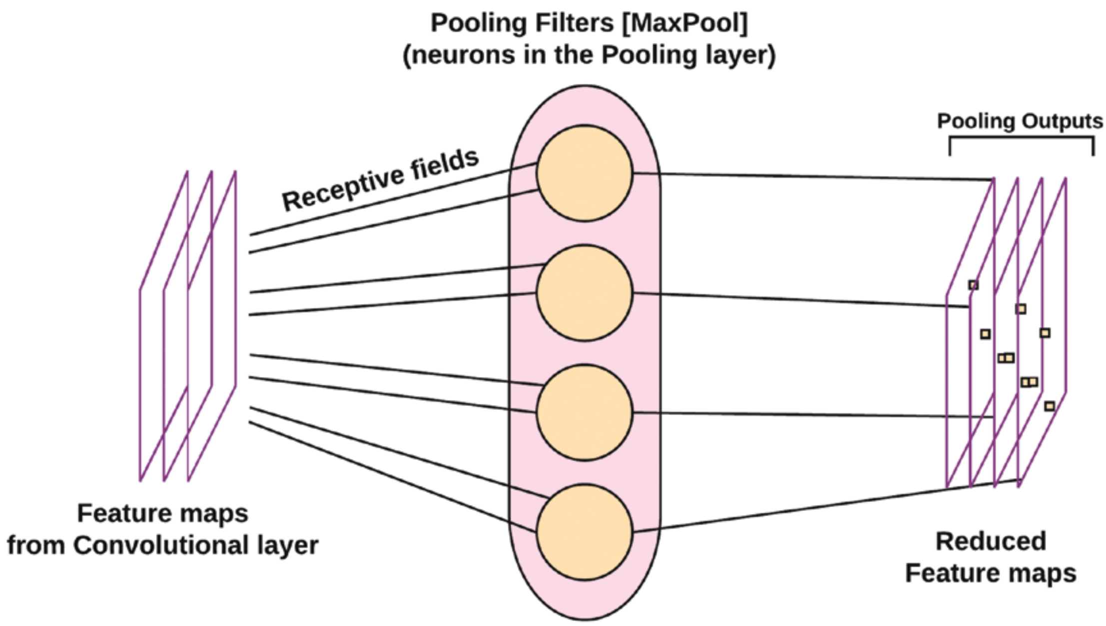

Figure 35-14. *Example of applying pooling to input with depth. Note that the filters in the pooling layer have no weights or biases*

The Fully Connected Network Layer

The fully connected network (FCN) layer is our regular feedforward neural network or multilayer perceptron. These layers typically have a non-linear activation function. In any case, the FCN is the final layer of the convolutional neural network. In this case, a softmax activation is used to output the probabilities that an input belongs to a particular class.