**A complete neural network for image recognition**

- [Instructor] Let's finish the code to define our neural network. All that's left to do is tell Keras to compile it. Open up 08\_complete\_neural\_net.py. The final step of defining a neural network is to compile the neural network. When we compile it, we're telling Keras we actually want to create the neural network in memory. We're also telling Keras how we'll be training it and measuring its accuracy. Let's go down to line 39. To compile the neural network, we'll call model.compile. This function takes several parameters.

First, we need to tell Keras how to check how right or wrong the guesses from our neural network are. This is called the loss function. So we'll pass in a loss parameter, loss equals. You can write your own loss function, but most of the time there are a few standard functions that you'll choose between. If you're trying to classify an imageinto different categories, you can use a function called categorical crossentropy. If you're only checking if an image belongs to one category, you'll use binary crossentropy.

Since we have 10 different possible categories for our objects with our data set, we'll use categorical crossentropy. So we'll pass that in as a string. Categorical crossentropy. Next, we need to tell Keras which optimization algorithm we'll use to train the neural network. As researchers come up with more efficient trainingalgorithms, Keras adds support for them. For image data like this, a good starting point is to use an optimization algorithm called Adam, or Adaptive Moment Estimation.

We'll use that one. So to do that, we'll pass in, optimizer equals adam. Finally, we need to tell Keras what metrics we want it to report during the training process. For this we can just use accuracy. So we'll pass in, metrics equals, and then inside in the array, accuracy, as a string. We pass it in the array for metrics since you can choose to report more than one, but for our purposes, accuracy is all we need.

All right, just to make sure we've typed in everything correctly, let's run the code. Right click and choose Run. And everything looks good. Notice that compiling the neural network doesn't actually start the training process. In the next section, we'll run the training process and use the training neural net to make predictions.