

1. Going deeper

One of the major strengths of convolutional neural networks comes from building networks with multiple layers of convolutional filters. This is why using artificial neural networks is sometimes also called "deep learning".

2. Network with one convolutional layer

This is the diagram that describes the network that we saw before. It has one convolutional layer, followed by a flattening and readout with a fully connected layer with 3 units.

3. Network with one convolutional layer: implementation

And here is the code that implements this network.

4. Building a deeper network

In this diagram, we have added one more convolutional layer. This layer also has 10 feature maps. Instead of operating directly on the image, the convolutions in this layer operate on each of the feature maps in the first convolutional layer. After the second layer we again flatten the output of the convolutions and then pass on the flattened output to a 3-unit fully connected layer that then provides the output of the network. Let's see how we might implement this.

5. Building a deep network

Here is the code for this network. First, we create the input layer. This is the same convolutional layer that we had in our previous network. Then, we add one more convolutional layer to the network. Because this layer receives its inputs from the first convolutional layer, it doesn't require the `input_shape` keyword argument to be provided. Finally, we flatten the output of the second layer and pass it to a Dense layer with softmax activation, to decide on the output.

6. Why do we want deep networks?

Why do we want to add additional layers to a network? This is again motivated by our own visual system, which has multiple layers of processing in it. For example, this is the architecture of a network developed by Google researchers in 2014. It has 22 layers of convolutions, and some other kinds of layers, like pooling layers, that we will discuss in a subsequent lesson. For the time being, one way to understand why we would want a network this deep is by looking at the kinds of things that the kernels and feature maps in the different layers tend to respond to.

7. Features in early layers

For example, these are the kinds of things that layers in the early part of the network tend to respond to. Oriented lines, or simple textures.

8. Features in intermediate layers

Intermediate layers of the network tend to respond to more complex features, that include simple objects, such as eyes.

9. Features in late layers

By the time the information travels up to higher layers of the network, the feature maps tend to extract specific types of objects. This allows the fully connected layers at the top of the network to extract useful information for object classification based on the responses of these layers. In other words, having multiple layers of convolutions in the network allows the network to gradually build up representations of objects in the images from simple features to more complex features and up to sensitivity to distinct categories of objects.

10. How deep?

How deep should your network be? Consider the following: despite the advantages mentioned above, depth does come at a computational cost. In addition, to train a very deep network you might need a larger amount of training data.