

1. Tweaking your convolutions

Let's talk about the details of convolutions, and how you might tweak them to do things slightly differently.

2. Convolution

This is an animation of a convolution that you have seen before. One thing that you might have noticed before is that the blue input image is larger than the green output image. This is because the convolution kernel has the size of three-by-three pixels. In this case, it converts a three-by-three window into one pixel in the output image. One way to deal with this issue is to zero-pad the input image.

3. Convolution with zero padding

Here's what that looks like. The dashed boxes around the central image are zeros that are added to the image. As you can see, when the input image is zero padded, the output feature map has the same size as the input. This can be useful if you want to build a network that has many layers. Otherwise, you might lose a pixel off the edge of the image in each subsequent layer.

4. Zero padding in Keras

To implement zero padding in keras, we will use the Conv2D object's padding keyword argument. If we provide the value "valid", no zero padding is added. This is also the default behavior, so if you don't specify padding, this is what you will get.

5. Zero padding in Keras

On the other hand, if we provide the value "same", zero padding will be applied to the input to this layer, so that the output of the convolution has the same size as the input into the convolution.

6. Strides

Another factor that affects the size of the output of a convolution is the size of the step that we take with the kernel between input pixels. This is called the size of the stride. For example, in this animation the kernel is strided by two pixels in each step. This means again that the output size is smaller than the input size.

7. Strides in Keras

Strides are also implemented as a keyword argument to the Conv2D layers. The default is for the stride to be set to 1. This means that the kernel slides along the image and is multiplied and summed with each pixel location.

8. Strides in Keras

If the stride is set to more than 1, the kernel jumps in steps of that number of pixels. This also means that the output will be smaller.

9. Example

For example, if the input image is five by five and there is both zero padding and strides set to 2, the output will have a size of three by three.

10. Calculating the size of the output

Generally, we can calculate the size of the output using a simple formula: $(I - K + 2P) / (S + 1)$ Where I is the size of the input, K is the size of the kernel, P is the size of the zero padding, and S is the stride.

11. Calculating the size of the output

For example, if the input is 28 pixels, the kernel is 3 by 3, the padding is of 1 and the stride is 1, this number will be: $(28 - 3 + 2) / 1 + 1 = 28$. If instead the stride is 3 the output size would be 10 by 10.

12. Dilated convolutions

Finally, you can also tweak the spacing between the pixels affected by the kernel. This is called a dilated convolution. In this case, the convolution kernel has only 9 parameters, but it has the same field of view as a kernel that would have the size 5 by 5. This is useful in cases where you need to aggregate information across multiple scales.

13. Dilation in Keras

This too is controlled through a keyword argument, "dilation_rate", that sets the distance between subsequent pixels.