Simon Fraser University
Department of Computer Science

Deep Learning - Assignment 2

Name: Niloufar Saeidi
ID: 301590708

Exercise 3.1

The upper-left hand side of the figure has a vertical line, so the kernel we pick should return a relatively high value(=8) when facing a vertical line.

(a) The kernel should look like something like this. If we convolve this filter to a blank image(with equal-valued pixels) we will get an output zero because positive and negative values will cancel each other out. but if we apply it to an image with a vertical line in the middle, we will get the highest value possible.

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

If we apply this kernel to the upper-left hand side of the figure we get:

$$(-1\cdot 0+2\cdot 2+-1\cdot 0)+(-1\cdot 0+2\cdot 2+-1\cdot 0)+(-1\cdot 0+2\cdot 2+-1\cdot 0) = 0+4+0+0+4+0+0+4+0 = 12$$

To make it return 8 instead of 12, we can adjust the kernel by dividing each value by 1.5:

$$\begin{bmatrix} -1/1.5 & 2/1.5 & -1/1.5 \\ -1/1.5 & 2/1.5 & -1/1.5 \\ -1/1.5 & 2/1.5 & -1/1.5 \end{bmatrix}$$

(b) If our kernel has a vertical line on the left or the right it will work too.

$$\begin{bmatrix} 2 & -1 & -1 \\ 2 & -1 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

or

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & -1 & 2 \\ -1 & -1 & 2 \end{bmatrix}$$

These would detect vertical lines that are on the right or left side of the patch. If we adjust the values of these two matrices by dividing all their elements by 1.5, their response will be 8 too.

Exercise 3.2

Same padding is designed to ensure that the output of the convolution has the same spatial dimensions as the input. We do this by padding the edges of the input with zeros (imaginary pixels) so that the convolution filter can slide over the beginning and end edges of the input without reducing the size of the output.

The number of times a convolution filter is applied horizontally across an image depends on the stride and the width of the image, not the size of the filter. With Same padding, the stride (s_h) determines the step size for moving the filter from one position to the next.

When the filter reaches the boundary of the image, Same padding ensures that there are enough zeros added to the edges so that the filter can still fit over the image. This means the filter can have its center over the last actual pixel of the image, with the remaining part of the filter extending over the padded zeros.

The goal of Same padding is to keep the output dimensions equal to the input dimensions divided by the stride, rounded up to the nearest integer (which is what the ceiling function $\lceil \cdot \rceil$ ensures). Because of this, no matter the size of the filter, the padding will be adjusted to maintain the output size.

So with Same padding, the edges of the input are padded with enough zeros to allow for the filter to be applied as many times as needed based on the stride and input size, regardless of the actual filter size.

3.4

(a) The filters' values are stored in a 4-dimensional tensor. The dimensions correspond to filter height, filter width, input channels (depth of the input image), and number of filters.

Because the input is a 32×32 color image, we have 3 input channels (for RGB). With eight 5×5 filters, the shape of the variable to store the filters' values would be:

$$[5, 5, 3, 8]$$

(b) The output shape of a convolution operation using Valid padding and a stride $= 2$ will be:

- Output height: $\left\lfloor \frac{(i_h - f_h + 1)}{s_h} \right\rfloor$
- Output width: $\left\lfloor \frac{(i_v - f_v + 1)}{s_v} \right\rfloor$

Using Valid padding means no padding is added, so the filter only goes over the image where it fits completely within the image boundaries.

So the output dimensions will be:

- For the height: $\left\lfloor \frac{(32-5+1)}{2} \right\rfloor = \left\lfloor \frac{28}{2} \right\rfloor = 14$
- For the width: $\left\lfloor \frac{(32-5+1)}{2} \right\rfloor = \left\lfloor \frac{28}{2} \right\rfloor = 14$

The number of output channels is equal to the number of filters applied, which is 8. Therefore, the shape of the output of `tf.nn.conv2d` would be:

$$[14, 14, 8]$$

This represents the height, width, and depth (number of filters/output channels) of the output tensor after the convolution operation.

A.4

In same padding with stride = 1, we want to ensure that the output size is the same as the input size. The relationship between the input size (I), the filter size (F), the padding size (P), and the output size (O) in a convolution operation is:

$$O = \frac{I - F + 2P}{S} + 1$$

For same padding with (S) = 1 and the requirement that $O = I$, we can rearrange the formula to solve for P:

$$I = \frac{I - F + 2P}{1} + 1$$

Simplifying:

$$I = I - F + 2P + 1$$

$$F - 1 = 2P$$

$$P = \frac{F - 1}{2}$$

For P to be an integer, F should be odd.