Model Answer for May 16th Exercise

Question 1

a) *Standard Convolution in 2-D:*

This is a very simple convolution with no padding, no stride, and just the application of the 4x4 convolution kernel once to the whole input 4x4 input matrix:

$$y = w_0x_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_6x_6 + w_7x_7 + w_8x_8 + w_9x_9 \\ + w_{10}x_{10} + w_{11}x_{11} + w_{12}x_{12} + w_{13}x_{13} + w_{14}x_{14} + w_{15}x_{15}$$

This can be written as a simple matrix multiplication, after reshaping - or unrolling - the convolution kernel and the input vector into one-dimensional arrays:

$$y = (w_0, w_1, \ldots, w_{15})\begin{pmatrix} x_0 \\ x_1 \\ \cdot \\ \cdot \\ x_{15} \end{pmatrix} \qquad \text{(a1)}$$

b) *Transposed Convolution in 2-D :*

Now the input is a 1x1 vector y and the output is a 4x4 array. $\begin{matrix} x_0 & \cdots & x_3 \\ \vdots & \ddots & \vdots \\ x_{12} & \cdots & x_{15} \end{matrix}$

If we reshape the x array into a one-dimensional vector and we transpose equation (a1) above, we see that the expression of the Transposed Convolution will be:

$$\begin{pmatrix} x_0 \\ x_1 \\ \cdot \\ \cdot \\ x_{15} \end{pmatrix} = \begin{pmatrix} w_0 \\ w_1 \\ \cdot \\ \cdot \\ w_{15} \end{pmatrix} y = \begin{pmatrix} yw_0 \\ yw_1 \\ \cdot \\ \cdot \\ yw_{15} \end{pmatrix}$$

To get the 4x4 matrix we just reshape the result into the array:

$$\begin{matrix} yw_0 & \cdots & yw_3 \\ \vdots & \ddots & \vdots \\ yw_{12} & \cdots & yw_{15} \end{matrix} \qquad \text{(b1)}$$

Which is the result of the transposed convolution: we have transformed the real number (or 1x1 matrix) y into a 4x4 matrix. It appears that the transposed convolution consists here simply of multiplying each coefficient of the convolution kernel by the input value y.

c) If the input now has two channels y_0 and y_1 and we want the 4x4 output to have only one channel, this implies that we must be dealing with only one convolution kernel (otherwise we would have more than just one output channel). This kernel has to apply to the two input channels, so its third dimension must necessary be two. Let us call $\begin{pmatrix} w_0^0 \\ w_1^0 \\ \cdot \\ \cdot \\ w_{15}^0 \end{pmatrix}$ the first slice and $\begin{pmatrix} w_0^1 \\ w_1^1 \\ \cdot \\ \cdot \\ w_{15}^1 \end{pmatrix}$ the second slice of the convolution kernel.

Hence the output will be:

$$\begin{pmatrix} y_0 w_0^0 & \cdots & y_0 w_3^0 \\ \vdots & \ddots & \vdots \\ y_0 w_{12}^0 & \cdots & y_0 w_{15}^0 \end{pmatrix} + \begin{pmatrix} y_1 w_0^1 & \cdots & y_1 w_3^1 \\ \vdots & \ddots & \vdots \\ y_1 w_{12}^1 & \cdots & y_1 w_{15}^1 \end{pmatrix} =$$

$$\begin{pmatrix} y_0 w_0^0 + y_1 w_0^1 & \cdots & y_0 w_3^0 + y_1 w_3^1 \\ \vdots & \ddots & \vdots \\ y_0 w_{12}^0 + y_1 w_{12}^1 & \cdots & y_0 w_{15}^0 + y_1 w_{15}^1 \end{pmatrix} \qquad \text{(c1)}$$

So we simply multiply each of the two inputs by its corresponding slice of the convolution kernel and then add the two results.

d) If we now want two channels for the 4x4 output, we have this time to use two convolution kernels instead of one. If we keep the first notation as above for the first convolution kernel and assume that $\begin{pmatrix} w_0^{0\prime} \\ w_1^{0\prime} \\ \cdot \\ \cdot \\ w_{15}^{0}{}' \end{pmatrix}$ is the first slice and $\begin{pmatrix} w_0^{1\prime} \\ w_1^{1\prime} \\ \cdot \\ \cdot \\ w_{15}^{1}{}' \end{pmatrix}$ is the second slice of the second convolution kernel, the first output channel will be the same as given by equation (c1) and the second channel will simply be:.

$$\begin{pmatrix} y_0 w_0^{0\prime} + y_1 w_0^{1\prime} & \cdots & y_0 w_3^{0\prime} + y_1 w_3^{1\prime} \\ \vdots & \ddots & \vdots \\ y_0 w_{12}^{0}{}' + y_1 w_{12}^{1}{}' & \cdots & y_0 w_{15}^{0}{}' + y_1 w_{15}^{1}{}' \end{pmatrix}$$

Question 2

Start from a simple example of convolution, where we take an image $\begin{matrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{matrix}$ of size 4x4, with no padding, and apply to it a convolution kernel $\begin{matrix} w_0 & w_1 & w_2 \\ w_3 & w_4 & w_5 \\ w_6 & w_7 & w_8 \end{matrix}$ of dimension 3x3.

The classical formula says that, if $n \times n$ is the size of the input image, $f \times f$ is the size of the convolution filter, p is the padding and s is the stride, then the size of the output is: $\left(\frac{n+2p-f}{s} + 1\right) \times \left(\frac{n+2p-f}{s} + 1\right)$. Here we have. $n = 4, p = 0, f = 3 \; and \; s = 1$. Hence the output image is $\left(\frac{4+0-3}{1} + 1\right) \times \left(\frac{4+0-3}{1} + 1\right)$.

This will provide us with a 2x2 output array. The output will be as follows:

$$\begin{matrix} w_0x_0 + w_1x_1 + w_2x_2 + w_3x_4 + w_4x_5 + w_5x_6 + w_6x_8 + w_7x_9 + w_8x_{10} & w_0x_1 + w_1x_2 + w_2x_3 + w_3x_5 + w_4x_6 + w_5x_7 + w_6x_9 + w_7x_{10} + w_8x_{11} \\ w_0x_4 + w_1x_5 + w_2x_6 + w_3x_8 + w_4x_9 + w_5x_{10} + w_6x_{12} + w_7x_{13} + w_8x_{14} & w_0x_5 + w_1x_6 + w_2x_7 + w_3x_9 + w_4x_{10} + w_5x_{11} + w_6x_{13} + w_7x_{14} + w_8x_{15} \end{matrix}$$

If we unroll the input and output, convolution transforms a vector $x = \begin{pmatrix} x_0 \\ x_1 \\ \cdot \\ \cdot \\ x_{15} \end{pmatrix}$ of 16 coordinates into a vector $y = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$ of 4 coordinates. As discussed in the course, the convolution operation can be written as a matrix multiplication where the matrix is 4x16 and this matrix is a function of the kernel coefficients only:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} w_0 & w_1 & w_2 & 0 & w_3 & w_4 & w_5 & 0 & w_6 & w_7 & w_8 & 0 & 0 & 0 & 0 & 0 \\ 0 & w_0 & w_1 & w_2 & 0 & w_3 & w_4 & w_5 & 0 & w_6 & w_7 & w_8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_0 & w_1 & w_2 & 0 & w_3 & w_4 & w_5 & 0 & w_6 & w_7 & w_8 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_0 & w_1 & w_2 & 0 & w_3 & w_4 & w_5 & 0 & w_6 & w_7 & w_8 \end{pmatrix} \times \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{pmatrix}$$

The transposed convolution is then obtained by taking the transposed matrix of the 4x16 matrix above and multiplying it by a vector of size 4.

$$\begin{pmatrix} w_0 & 0 & 0 & 0 \\ w_1 & w_0 & 0 & 0 \\ w_2 & w_1 & 0 & 0 \\ 0 & w_2 & 0 & 0 \\ w_3 & 0 & w_0 & 0 \\ w_4 & w_3 & w_1 & w_0 \\ w_5 & w_4 & w_2 & w_1 \\ 0 & w_5 & 0 & w_2 \\ w_6 & 0 & w_3 & 0 \\ w_7 & w_6 & w_4 & w_3 \\ w_8 & w_7 & w_5 & w_4 \\ 0 & w_8 & 0 & w_5 \\ 0 & 0 & w_6 & 0 \\ 0 & 0 & w_7 & w_6 \\ 0 & 0 & w_8 & w_7 \\ 0 & 0 & 0 & w_8 \end{pmatrix} \times \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

The reshaped result of this multiplication is as follows

$$\begin{matrix} w_0y_0 & w_1y_0 + w_0y_1 & w_2y_0 + w_1y_1 & w_2y_1 \\ w_3y_0 + w_0y_2 & w_5y_0 + w_3y_1 + w_1y_2 + w_0y_3 & w_5y_0 + w_4y_1 + w_2y_2 + w_1y_3 & w_5y_1 + w_2y_3 \\ w_6y_0 + w_3y_2 & w_7y_0 + w_6y_1 + w_4y_2 + w_3y_3 & w_8y_0 + w_7y_1 + w_5y_2 + w_4y_3 & w_8y_1 + w_5y_3 \\ w_6y_2 & w_7y_2 + w_6y_3 & w_8y_2 + w_7y_3 & w_8y_3 \end{matrix} \quad (2.1)$$

Let us now consider the other way to obtain the same result, and this is using a convolution (not a transposed convolution), but after applying a padding of 2 to the 2x2 input image:

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & y_0 & y_1 & 0 & 0 \\ 0 & 0 & y_2 & y_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Applying the convolution kernel. $\begin{matrix} w_0 & w_1 & w_2 \\ w_3 & w_4 & w_5 \\ w_6 & w_7 & w_8 \end{matrix}$ to it with a stride of one, we obtain a similar

transposed convolution as before. The output is the following 4x4 array:

$$\begin{matrix} w_8y_0 & w_7y_0 + w_8y_1 & w_0y_0 + w_7y_1 & w_6y_1 \\ w_5y_0 + w_8y_2 & w_4y_0 + w_5y_1 + w_7y_2 + w_8y_3 & w_3y_0 + w_4y_1 + w_6y_2 + w_7y_3 & w_4y_1 + w_7y_3 \\ w_2y_0 + w_5y_2 & w_1y_0 + w_2y_1 + w_4y_2 + w_5y_3 & w_0y_0 + w_1y_1 + w_3y_2 + w_4y_3 & w_0y_1 + w_3y_3 \\ w_2y_2 & w_1y_2 + w_2y_3 & w_0y_2 + w_1y_3 & w_0y_3 \end{matrix} \quad (2.2)$$

If we compare tables (2.1) and 2.2 we see that they are identical if we replace w_0 by w_8, w_1 by w_7, w_2 by w_6, w_3 by w_5, leaving w_4 unchanged. Thus the two operations are equivalent as long as we

use. $\begin{matrix} w_0 & w_1 & w_2 \\ w_3 & w_4 & w_5 \\ w_6 & w_7 & w_8 \end{matrix}$ for the first one and $\begin{matrix} w_8 & w_7 & w_6 \\ w_5 & w_4 & w_3 \\ w_2 & w_1 & w_0 \end{matrix}$ for the other one.