Assignment 2: CS 663, Fall 2021

Question 1

- Consider a 1D convolution mask given as $(w_0, w_1, ..., w_6)$. Express the convolution of the mask with a 1D image f as the multiplication of a suitable matrix with the image vector f. What are the properties of this matrix? What could be a potential application of such a matrix-based construction? [10 points]

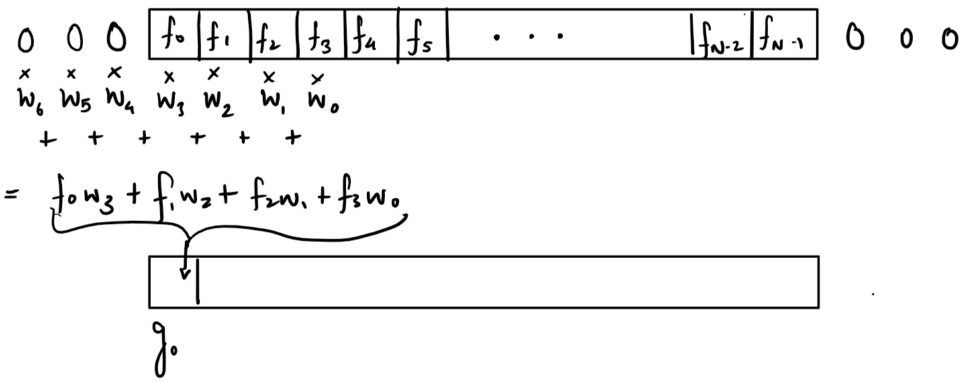
  Answer: The 1D convolution mask is given as $\{w_0, w_1, w_2, w_3, w_4, w_5, w_6\}$
  Let the image vector be represented as $\{f_0, f_1, f_2, f_3, ..., f_{N-1}\}$.
  Also, let the result of the convolution be represented as $\{g_0, g_1, g_2, ..., g_{N-1}\}$
  (Here we have assumed that the necessary padding has been done so that the output has the same dimension as the image)
  To find convolution between the mask and the image, we flip the mask by 180^0, multiply the elements of the mask with the image, followed by summing them, which represents the value of the output. We can show it diagrammatically a moving train of flipped mask as follows

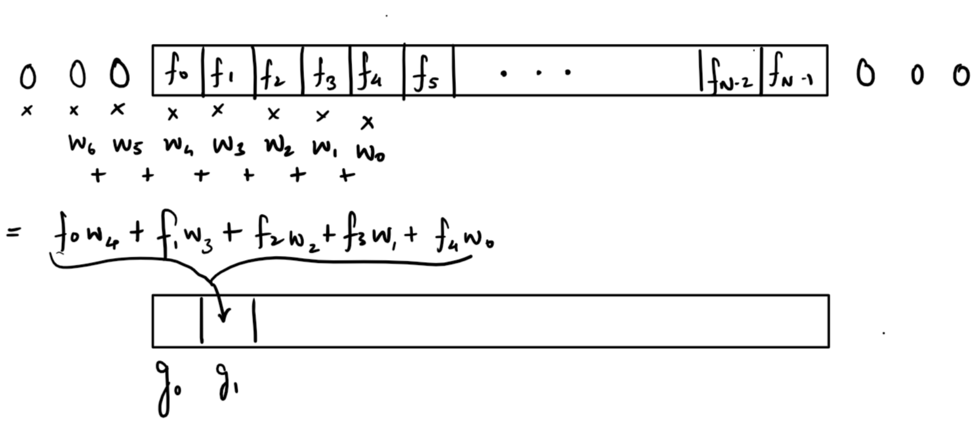
  The value that we get as the result is equivalent to the following

$$g_0 = \mathbf{f} \begin{bmatrix} w_3 \\ w_2 \\ w_1 \\ w_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

  where $\mathbf{f}$ is the $1 \times N$ row vector, $\mathbf{f} = [f_0, f_1, f_2, ..., f_{N-1}]$

  Moving the train (flipped mask) forward the 1 to calculate the value of g_1





This is equivalent to the following product

$$g_1 = \mathbf{f} \begin{bmatrix} w_4 \\ w_3 \\ w_2 \\ w_1 \\ w_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Performing the same calculation for all the g_i we can easily see that the result of the convolution can be represented by the following matrix multiplication

$$\mathbf{g} = \mathbf{f} \begin{bmatrix}
w_3 & w_4 & w_5 & w_6 & 0 & 0 & 0 & \dots & 0 & 0 \\
w_2 & w_3 & w_4 & w_5 & w_6 & 0 & 0 & \dots & 0 & 0 \\
w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & 0 & \dots & 0 & 0 \\
w_0 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & \dots & 0 & 0 \\
0 & w_0 & w_1 & w_2 & w_3 & w_4 & w_5 & \dots & 0 & 0 \\
0 & 0 & w_0 & w_1 & w_2 & w_3 & w_4 & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & w_5 & w_6 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & w_4 & w_5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & w_3 & w_4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & w_2 & w_3
\end{bmatrix}$$

From the above matrix constructed for convolution, we can easily see that it is Band Matrix (sparse matrix whose non-zero entries are confined to a diagonal band, comprising the main diagonal and zero or more diagonals on either side). It is also called as the Toeplitz Matrix
These can be used for cases where several convolutions are needed to be done on a given image such as in Convolutional Neural Network (Since all the convolution operations can be converted into one matrix by multiplication of all the convolution matrices and then be with the on a given image (vector form of it) at once). For applications to images or convolution networks, to more efficiently use the matrix multipliers in modern GPUs, the inputs are typically reshaped into columns of an activation matrix that can then be multiplied with multiple filters/kernels at once.