

# **CS663 Assignment 2**

Saksham Rathi, Kavya Gupta, Shravan Srinivasa Raghavan

September 2024

# Contents

Question 2	3
------------	---

## Question 2

### Matrix Calculation

Say the 1D image  $\mathbf{f}$  looks like  $(f_0, f_1, \dots, f_{n-1})$  and its convoluted form (say  $\mathbf{F}$ ) looks like  $(F_0, F_1, \dots, F_{n-1})$ .

Given 1D convolution mask  $\mathbf{w}$  is  $(w_0, w_1, \dots, w_6)$ . Rotate it by  $180^\circ$  and move over  $\mathbf{f}$ . Let's do the first step of convolution:-

$$\begin{array}{cccccccc} & & & f_0 & f_1 & f_2 & f_3 & \dots & f_{n-1} \\ w_6 & w_5 & w_4 & w_3 & w_2 & w_1 & w_0 & & \end{array}$$

This gives  $F_0 = w_3f_0 + w_2f_1 + w_1f_2 + w_0f_3 = w_3f_0 + w_2f_1 + w_1f_2 + w_0f_3 + \sum_{i=4}^{n-1} 0 \times f_i$  (Putting other coefficients as 0). This can be equivalently written as:-

$$F_0 = \mathbf{f} \begin{bmatrix} w_3 \\ w_2 \\ w_1 \\ w_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Move  $\mathbf{w}$  one step right. The next step of convolution would look like:-

$$\begin{array}{cccccccc} & & & f_0 & f_1 & f_2 & f_3 & f_4 & \dots & f_{n-1} \\ w_6 & w_5 & w_4 & w_3 & w_2 & w_1 & w_0 & & & \end{array}$$

This gives  $F_1 = w_4f_0 + w_3f_1 + w_2f_2 + w_1f_3 + w_0f_4 = w_4f_0 + w_3f_1 + w_2f_2 + w_1f_3 + w_0f_4 + \sum_{i=5}^{n-1} 0 \times f_i$ . This can be written as:-

$$F_1 = \mathbf{f} \begin{bmatrix} w_4 \\ w_3 \\ w_2 \\ w_1 \\ w_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

As you can see above, each step of convolution can be imitated by matrix multiplying  $\mathbf{f}$  with a suitable column vector. It is easy to see that the full result of convolution would be same as the matrix multiplication with the 2D matrix as shown (next page):-

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

## Matrix Properties

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).

## Potential Applications

These methods are particularly useful when performing multiple convolutions on a given image, such as in a Convolutional Neural Network (CNN). Instead of applying each convolution operation separately, all the convolution matrices can be combined into a single matrix through multiplication. This combined matrix is then applied to the image (in its vectorized form) in one step. For applications involving images or convolutional networks, the input data is often reshaped into columns of an activation matrix. This approach allows for more efficient use of matrix multipliers in modern GPUs, enabling simultaneous multiplication with multiple filters or kernels.