D7047E - Theoretical Tasks

Kalle Kalliorinne

April 16, 2020

1 Exercise

1.1 Task

Here we have the following image I and kernal k as input:

$$I = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & -3 & -4 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \qquad k = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{bmatrix} \tag{1}$$

These will be used within the convolutions operation $I * k$, where a zero padding will be used on the boundaries of the image. Two types of width for the zero padding are chosen to receive both a so called full output and an output with the same size as I. The results are the following matrices:

$$I * k = \begin{bmatrix} 1 & 1 & 2 & 2 & 1 & 1 \\ 2 & 4 & 7 & 6 & 5 & 2 \\ 2 & 1 & 3 & 5 & 1 & 2 \\ 2 & 1 & -6 & -6 & 0 & 2 \\ 1 & 4 & 1 & 0 & 4 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}, \qquad I * k = \begin{bmatrix} 4 & 7 & 6 & 5 \\ 1 & 3 & 5 & 1 \\ 1 & -6 & -6 & 0 \\ 4 & 1 & 0 & 4 \end{bmatrix} \tag{2}$$

1.2 Task

The ReLU function passes through all values above zero and transforms values below zero to zero. Using the ReLU operator on the output from the previous Task yields this result:

$$I * k = \begin{bmatrix} 4 & 7 & 6 & 5 \\ 1 & 3 & 5 & 1 \\ 1 & -6 & -6 & 0 \\ 4 & 1 & 0 & 4 \end{bmatrix}, \qquad I_1 = \mathrm{ReLU}(I * k) = \begin{bmatrix} 4 & 7 & 6 & 5 \\ 1 & 3 & 5 & 1 \\ 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 4 \end{bmatrix} \tag{3}$$

1.3 Task

Applying Max Pooling on the output from the previous task, the result is as follows:

$$I_1 = \begin{bmatrix} 4 & 7 & 6 & 5 \\ 1 & 3 & 5 & 1 \\ 1 & 0 & 0 & 0 \\ 4 & 1 & 0 & 4 \end{bmatrix}, \qquad I_2 = \mathrm{MaxPool}(I_1) = \begin{bmatrix} 7 & 6 \\ 4 & 4 \end{bmatrix} \tag{4}$$

1.4 Task

Applying a Flattening operation on the output from the previous task, the result is as follows:

$$I_2 = \begin{bmatrix} 7 & 6 \\ 4 & 4 \end{bmatrix}, \qquad I_3 = \mathrm{Flatten}(I_2) = \begin{bmatrix} 7 & 6 & 4 & 4 \end{bmatrix}^T \tag{5}$$

1.5 Task

The flattened matrix I_3 can now in a very simple way be passed through a fully connected layer with weights W. The result is as followed:

$$W = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}, \quad I_3 = \begin{bmatrix} 7 & 6 & 4 & 4 \end{bmatrix}^T, \quad I_4 = W \cdot I_3 = \begin{bmatrix} 47 \\ 131 \end{bmatrix} \tag{6}$$

1.6 Task

The output from the previous task can now be transformed using a SoftMax() function.

$$I_4 = \begin{bmatrix} 47 \\ 131 \end{bmatrix}, \quad \mathrm{SoftMax}(I_4) = \begin{bmatrix} 0.264 \\ 0.736 \end{bmatrix} \tag{7}$$

The output from the SoftMax() indicates that the second class is more probable to be correct.