COMP7703 – Homework Task 7
Joel Thomas 44793203

<u>**Note**</u> – questions **1-2, 3** and **4** have all been completed using the methodology described on pages **237**, **244** and **251** respectively of the *Dive into Deep Learning* book authored by Zhang et al.

1.

Let us define the output from the convolution operation as Y.

$\therefore H * K = Y$

We know that $\dim(Y) = (H_h - K_h + 1) \times (H_w - K_w + 1)$ where $h =$ height of the matrix/vector and $w =$ width of the matrix/vector.

$\therefore \dim(Y) = (5 - 2 + 1) \times (5 - 1 + 1) = 4 \times 5$

Using a stride of 1, performing a convolution operation on H using K, we have:

$$H * K = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = Y$$

2.

Let us define the output from the convolution operation as Y.

$\therefore X * K = Y$

We know that $\dim(Y) = (X_h - K_h + 1) \times (X_w - K_w + 1)$ where $h =$ height of the matrix/vector and $w =$ width of the matrix/vector.

$\therefore \dim(Y) = (5 - 2 + 1) \times (5 - 2 + 1) = 4 \times 4$

Using a stride of 1, performing a convolution operation on X using K, we have:

$$X * K = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 2 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 1 \\ 2 & 0 & 1 & 0 \end{bmatrix} = Y$$

3.

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Since there is now non-zero padding along with stride > 1, we use a modified formula to obtain $\dim(Y)$:

$$\dim(Y) = \left\lfloor \frac{X_h - K_h + P_h + S_h}{S_h} \right\rfloor \times \left\lfloor \frac{X_w - K_w + P_w + S_w}{S_w} \right\rfloor$$

where P_h, P_w and S_h, S_w represent the padding height and width and stride for the height and width respectively.

$$\therefore \dim(Y) = \left\lfloor \frac{5 - 2 + 1 + 2}{2} \right\rfloor \times \left\lfloor \frac{5 - 2 + 1 + 2}{2} \right\rfloor$$

$$\dim(Y) = \left\lfloor \frac{6}{2} \right\rfloor \times \left\lfloor \frac{6}{2} \right\rfloor$$

$$\dim(Y) = 3 \times 3$$

Using a stride of 2 (both height and width), performing a convolution operation on X using K, we have:

$$X * K = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{0} = Y$$

4.

Let us define the average pooling operator and output from the average pooling operation as A and Z respectively.

$$\therefore H * A = Z$$

We know that $\dim(Y) = (H_h - A_h + 1) \times (H_w - A_w + 1)$ where $h =$ height of the matrix/vector and $w =$ width of the matrix/vector noting that we have zero padding on H and stride of 1 for both height and width.

$$\therefore \dim(Y) = (5 - 3 + 1) \times (5 - 3 + 1) = 3 \times 3$$

Using a stride of 1, performing an average pooling operation on H using A, we have:

$$H \cdot A = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 3 \times 3 \\ \text{Average Pooling} \end{bmatrix} = \begin{bmatrix} \frac{5}{9} & \frac{1}{3} & \frac{5}{9} \\ \frac{5}{9} & \frac{1}{3} & \frac{5}{9} \\ \frac{5}{9} & \frac{1}{3} & \frac{5}{9} \end{bmatrix} = Z$$

5.

Firstly, notice that whilst test accuracy (%) appears in both figures on the vertical axis, figure 6.6 has the model depth on the horizontal axis whereas figure 6.7 has the model size i.e. number of parameters (100s of millions) on the horizontal axis. Model depth is increased by increasing the number of hidden layers used whereas model size is typically increased by increasing the number of neurons in each hidden layer but can also include increasing the number of hidden layers prior to training. The main reason why deeper models tend to perform better than shallower models is that shallow models containing many parameters (i.e. wide shallow models) tend to be good at memorisation but not at generalisation resulting in an overfit model. For example, such models could be trained with every possible input provided in the training data such that the network memorises the desired corresponding target but fails when it comes to predicting outputs for unseen data. Deeper models tend to be better (at generalisation) since they can learn features at varying levels of abstraction – they learn all the intermediate features between the raw input and high-level classification (e.g. edges, shapes, facial features etc.). Finally, whilst in figure 6.7 the 11 hidden layer CNN overfits at around 60 million parameters, a very wide deep model is also undesirable because this is just introducing additional parameters that the network needs to learn during training. This can result in the same problem as wide shallow models where the model is good at memorisation but fails to generalise with unseen data i.e. overfitting. Furthermore, the smallest possible model (number of hidden layers, parameters) that achieves a desired level of test accuracy should be selected as the preferred model as training additional parameters in neural network can be very computationally expensive and time inefficient.