CPEN455: Deep Learning
Homework 2

Mercury Mcindoe 85594505

February 18th 2025

1 Problem 1

1.1

Solution:

With a single linear-layer network, the decision boundary in two dimensions is always a single straight line. Because the XOR function is not linearly separable, it is impossible to place a single straight line that correctly separates the points $(0,0)$ and $(1,1)$ (label (0)) from $(0,1)$ and $(1,0)$ (label (1)).

1.2

Solution:

Let's denote the weights and biases $W_1 \in \mathbb{R}^{2\times 2}, \mathbf{b}_1 \in \mathbb{R}^2$ and $W_2 \in \mathbb{R}^{1\times 2}, b_2 \in \mathbb{R}$ respectively for linear layers.
Then, I proposed them as the following,

$$W_1 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \quad \mathbf{b}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$W_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad b_2 = 0$$

We validate the proposed parameters by verifying with each input,

Case 1: $\mathbf{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$,

$$\mathbf{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \sigma(\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix})$$
$$= \sigma(\begin{bmatrix} -1 \\ 1 \end{bmatrix}) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$
$$\rightarrow \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0 = 1$$

Case 2: $\mathbf{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$\mathbf{x} = \begin{bmatrix}0\\0\end{bmatrix} \rightarrow \sigma(\begin{bmatrix}1 & -1\\1 & 1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix})$$
$$= \sigma(\begin{bmatrix}0\\0\end{bmatrix}) = \begin{bmatrix}0\\0\end{bmatrix}$$
$$\rightarrow \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} + 0 = 0$$

Case 3: $\mathbf{x} = \begin{bmatrix}1\\0\end{bmatrix}$

$$\mathbf{x} = \begin{bmatrix}1\\0\end{bmatrix} \rightarrow \sigma(\begin{bmatrix}1 & -1\\1 & 1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix})$$
$$= \sigma(\begin{bmatrix}1\\1\end{bmatrix}) = \begin{bmatrix}1\\1\end{bmatrix}$$
$$\rightarrow \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} + 0 = 1$$

Case 4: $\mathbf{x} = \begin{bmatrix}1\\1\end{bmatrix}$

$$\mathbf{x} = \begin{bmatrix}1\\1\end{bmatrix} \rightarrow \sigma(\begin{bmatrix}1 & -1\\1 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix})$$
$$= \sigma(\begin{bmatrix}0\\2\end{bmatrix}) = \begin{bmatrix}0\\2\end{bmatrix}$$
$$\rightarrow \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}0\\2\end{bmatrix} + 0 = 0$$

If we summaraize the results,

$\mathbf{x}$	label
$\begin{bmatrix}0 & 1\end{bmatrix}^T$	1
$\begin{bmatrix}0 & 0\end{bmatrix}^T$	0
$\begin{bmatrix}1 & 0\end{bmatrix}^T$	0
$\begin{bmatrix}1 & 1\end{bmatrix}^T$	1

as required.

1.3

Solution:

Let's denote the weights and biases $W_1 \in \mathbb{R}^{2\times2}, \mathbf{b}_1 \in \mathbb{R}^2$ and $W_2 \in \mathbb{R}^{1\times2}, b_2 \in \mathbb{R}$ respectively for linear layers.
Then, I proposed them as the following,

$$W_1 = \begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix} \quad b_1 = \begin{bmatrix}0\\0\end{bmatrix}$$
$$W_2 = \begin{bmatrix}1 & 1\end{bmatrix} \quad b_2 = \begin{bmatrix}0\\0\end{bmatrix}$$

We validate the proposed parameters by verifying with each input,

Case 1: $\mathbf{x} = \begin{bmatrix}0\\1\end{bmatrix}$,

$$\begin{aligned}\mathbf{x} = \begin{bmatrix}0\\1\end{bmatrix} &\rightarrow \sigma(\begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix})\\ &= \sigma(\begin{bmatrix}-1\\1\end{bmatrix}) = \begin{bmatrix}0\\1\end{bmatrix}\\ &\rightarrow \begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} + 0 = 1\end{aligned}$$

Case 2: $\mathbf{x} = \begin{bmatrix}0\\0\end{bmatrix}$

$$\begin{aligned}\mathbf{x} = \begin{bmatrix}0\\0\end{bmatrix} &\rightarrow \sigma(\begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix})\\ &= \sigma(\begin{bmatrix}0\\0\end{bmatrix}) = \begin{bmatrix}0\\0\end{bmatrix}\\ &\rightarrow \begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} + 0 = 0\end{aligned}$$

Case 3: $\mathbf{x} = \begin{bmatrix}1\\0\end{bmatrix}$

$$\begin{aligned}\mathbf{x} = \begin{bmatrix}1\\0\end{bmatrix} &\rightarrow \sigma(\begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix})\\ &= \sigma(\begin{bmatrix}1\\-1\end{bmatrix}) = \begin{bmatrix}1\\0\end{bmatrix}\\ &\rightarrow \begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} + 0 = 1\end{aligned}$$

Case 4: $\mathbf{x} = \begin{bmatrix}1\\1\end{bmatrix}$

$$\begin{aligned}\mathbf{x} = \begin{bmatrix}1\\1\end{bmatrix} &\rightarrow \sigma(\begin{bmatrix}1 & -1\\-1 & 1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} + \begin{bmatrix}0\\0\end{bmatrix})\\ &= \sigma(\begin{bmatrix}0\\0\end{bmatrix}) = \begin{bmatrix}0\\0\end{bmatrix}\\ &\rightarrow \begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix} + 0 = 0\end{aligned}$$

If we summaraize the results,

$\mathbf{x}$	label
$\begin{bmatrix}0 & 1\end{bmatrix}^T$	1
$\begin{bmatrix}0 & 0\end{bmatrix}^T$	0
$\begin{bmatrix}1 & 0\end{bmatrix}^T$	0
$\begin{bmatrix}1 & 1\end{bmatrix}^T$	1

as required.

2 Problem 2

Solution:

Let's first identify the dimensions after each layer.

In the **first layer**, we have 128 convolutional filters with kernel size $5 \times 3 \times 3$ along with stride 2 and padding 2. Given that the input dimensions for each image within the batch is $3 \times 512 \times 512$, the resulting width and height is

$$\left\lfloor \frac{512 + 2 \times 2 - 5}{2} \right\rfloor + 1 = \left\lfloor \frac{511}{2} \right\rfloor + 1 = 256$$

then giving that we convolve among all the channels at once, the resulting depth is 128 a.k.a the number of filters. Hence we remain with the dimensions,

$$H_1 : 100 \times (128 \times 256 \times 256)$$

In the **second layer**, we now have to convolve the $100 \times (128 \times 256 \times 256)$ batch with filter $5 \times 5 \times ?$, when performing convolution, we need to have the depth of the filter match the depth of the input, thus $? = 128$.

If we compute the output dimensions regarding that the stride is 2 and padding is 2,

$$\left\lfloor \frac{256 + 2 \times 2 - 5}{2} \right\rfloor + 1 = \left\lfloor \frac{255}{2} \right\rfloor + 1 = 128$$

with resulting depth of 64, since we convolve 64 filters and the convolution happens with all channels, we end up with

$$H_2 : 100 \times (64 \times 128 \times 128)$$

In the **third layer**, we again convolve with 32 $3 \times 3 \times ?$ filters with stide 2 and padding 1. Similar to the logic from the previous layer, we need to have the depth matching for the input and the kernels, thus $? = 64$, where 64 is the depth of H_2.

We once again compute the resulting size of the input,

$$\left\lfloor \frac{128 + 1 \times 2 - 3}{2} \right\rfloor + 1 = \left\lfloor \frac{127}{2} \right\rfloor + 1 = 64$$

We also have depth of 32 since we convolve with 32 filters, there fore the resulting dimensions are,

$$H_3 : 100 \times (32 \times 64 \times 64)$$

In the **fourth layer**, we now convolve with 16 kernels each of size $1 \times 1 \times ?$ with stride 1 and padding 0. Similarly applying the logic from previous layers, we need the depth of the input match the depth of the kernels, hence $? = 32$.

We then compute the size of the output,

$$\left\lfloor \frac{64 + 2 \times 0 - 1}{1} \right\rfloor + 1 = \lfloor 63 \rfloor + 1 = 64$$

We have 16 kernels in total, hence

$$H_4 : 100 \times (16 \times 64 \times 64)$$

In the **fifth layer**, we apply 2D average pooling with a kernel size of 2×2, with stride 2 and padding 0. Then the resulting size (height, width) after convolution becomes,

$$\left\lfloor \frac{64 + 2 \times 0 - 2}{2} \right\rfloor + 1 = \left\lfloor \frac{62}{2} \right\rfloor + 1 = 32$$

Unlike the previous layers, the depth is maintained, therefore,

$$H_5 : 100 \times (16 \times 32 \times 32)$$

In the **final layer**, we flatten the matrix from the previous layer into $100 \times (32 \times 32 \times 16) \rightarrow 100 \times 16384$ and apply a linear payer to compute the logits which by stated in the question is to create 10 categories, therefore

$$Y : 100 \times 10$$

2.1

Solution:

From the previous section we computed the ? values, which are in order $128, 64, 32$.

2.2

Solution:

From the very first section, we also computed all of these,

$$\begin{aligned} H_1 &: 100 \times (128 \times 256 \times 256) \\ H_2 &: 100 \times (64 \times 128 \times 128) \\ H_3 &: 100 \times (32 \times 64 \times 64) \\ H_4 &: 100 \times (16 \times 64 \times 64) \\ H_5 &: 100 \times (16 \times 32 \times 32) \\ Y &: 100 \times 10 \end{aligned}$$

2.3

Solution:

We need compute the number of learnable parameters per layer (and total). We know that for pooling operations, there are no learnable parameters since we just perform pooling. For convolution layers, the number of learnable parameters (excluding bias) would be the number of filters multiplied by the dimension of each filter. And for a linear layer, the number of learnable parameters would be the dimensionality of the weight matrix we apply. Therefore,

Learnable Parameters (Excluding Bias)

$$
\begin{aligned}
H_1 &: 128 \times 3 \times 5 \times 5 = 9,600 \\
H_2 &: 64 \times 128 \times 5 \times 5 = 204,800 \\
H_3 &: 32 \times 64 \times 3 \times 3 = 18,432 \\
H_4 &: 16 \times 32 \times 1 \times 1 = 512 \\
H_5 &: 0 \quad (\because \text{ pooling}) \\
Y &: 10 \times 16 \times 32 \times 32 = 163,840
\end{aligned}
$$

Why would the number of learnable parameters in Y be $10 \times 16 \times 32 \times 32$? It is because that the layer that outputs Y would take an input of size $16 \times 16 \times 32 \times 100$ which is flatted into $100 \times 16,384$. In order to output a matrix with dimension 10×100, the weight matrix W should satisfy $W \in \mathbb{R}^{10 \times 16,384}$. Hence, the number of parameters in the last layer is $10 \times 16,384 = 163,840$.

The total number of parameters is then,

$$
\begin{aligned}
&9,600 + 204,800 + 18,432 + 512 + 163,840 \\
&= 397,184
\end{aligned}
$$

2.4

Solution:

Now, let's compute the number of learnable parameters including the bias parameters. For pooling layers, there is no bias as well hence still 0 learnable parameters. For convolutional layers, there is 1 bias term per filter. For the last linear layer, the number of additional learnable parameters would be simply the dimensionality of the bias. In this case, it would be the same number of output categories.

$$
\begin{aligned}
H_1 &: 128 \times (3 \times 5 \times 5 + 1) = 9,728 \\
H_2 &: 64 \times (128 \times 5 \times 5 + 1) = 204,864 \\
H_3 &: 32 \times (64 \times 3 \times +1) = 18,464 \\
H_4 &: 16 \times (32 \times 1 \times 1 + 1) = 528 \\
H_5 &: 0 \quad (\because \text{ pooling}) \\
Y &: 10 \times 16 \times 32 \times 32 + 10 = 163,850
\end{aligned}
$$

Hence, the total number of parameters is,

$$
\begin{aligned}
&9,728 + 204,864 + 18,464 + 528 + 163,850 \\
&= 397,434
\end{aligned}
$$

2.5

Solution:

Case 1, $\mu, \sigma \in \mathbb{R}$:

$$\mu = \frac{1}{B \times C \times H \times W} \sum_{b=1}^{B} \sum_{c=1}^{C} \sum_{h=1}^{H} \sum_{w=1}^{W} X[b, c, h, w]$$

$$\sigma = \left(\frac{1}{B \times C \times H \times W} \sum_{b=1}^{B} \sum_{c=1}^{C} \sum_{h=1}^{H} \sum_{w=1}^{W} (X[b, c, h, w] - \mu)^2 \right)^{1/2}$$

$$\hat{X}[b, c, h, w] = \frac{X[b, c, h, w] - \mu}{\sqrt{\sigma^2}}$$

Case 2, $\mu, \sigma \in \mathbb{R}^C$:

$$\mu[c] = \frac{1}{B \times H \times W} \sum_{b=1}^{B} \sum_{h=1}^{H} \sum_{w=1}^{W} X[b, c, h, w]$$

$$\sigma[c] = \left(\frac{1}{B \times H \times W} \sum_{b=1}^{B} \sum_{h=1}^{H} \sum_{w=1}^{W} (X[b, c, h, w] - \mu[c])^2 \right)^{1/2}$$

$$\hat{X}[b, c, h, w] = \frac{X[b, c, h, w] - \mu[c]}{\sqrt{\sigma[c]^2}}$$

Case 3, $\mu, \sigma \in \mathbb{R}^{H \times W}$:

$$\mu[h, w] = \frac{1}{B \times C} \sum_{b=1}^{B} \sum_{c=1}^{C} X[b, c, h, w]$$

$$\sigma[h, w] = \left(\frac{1}{B \times C} \sum_{b=1}^{B} \sum_{c=1}^{C} (X[b, c, h, w] - \mu[h, w])^2 \right)^{1/2}$$

$$\hat{X}[b, c, h, w] = \frac{X[b, c, h, w] - \mu[h, w]}{\sqrt{\sigma[h, w]^2}}$$

Case 4, $\mu, \epsilon \in \mathbb{R}^{C \times H \times W}$:

$$\mu[c, h, w] = \frac{1}{B} \sum_{b=1}^{B} X[b, c, h, w]$$

$$\sigma[c, h, w] = \left(\frac{1}{B} \sum_{b=1}^{B} (X[b, c, h, w] - \mu[h, w])^2 \right)^{1/2}$$

$$\hat{X}[b, c, h, w] = \frac{X[b, c, h, w] - \mu[c, h, w]}{\sqrt{\sigma[c, h, w]^2}}$$

If we were to use the same operations at different spatial locations but different operations across input channels, we would want to compute the mean and standard deviation using the C mean and C standard deviation. This is because batch normalization in convolutional layers typically

normalizes each feature map (channel) independently while maintaining consistency across spatial locations.

Computing per-channel statistics ensures that each channel has its own mean and variance, which helps preserve the spatial structure of the input while normalizing feature maps independently.

We would add the learnable parameters like the following,

$$\gamma[c]\frac{X[b,c,h,w]-\mu[c]}{\sqrt{\sigma[c]^2}}+\beta[c], \quad \beta,\gamma\in\mathbb{R}^{C\times 1}$$

3 Problem 3

3.1

Solution:

After padding the effective input size for both dimensions becomes,

$$H_{\text{eff}} = H + 2P$$
$$W_{\text{eff}} = W + 2P$$

For both dimensions, the starting index is 0, and the last possible step we can take would be at $H_{\text{eff}} - K, W_{\text{eff}} - K$. And since we are taking stride of S, the total number of steps we can take would be

$$\left\lfloor \frac{H_{\text{eff}} - K}{S} \right\rfloor + 1$$
$$\left\lfloor \frac{W_{\text{eff}} - K}{S} \right\rfloor + 1$$

where the $+1$ takes into account the initial step.

Hence, we get the expressions for H', W',

$$H' = \left\lfloor \frac{H + 2P - K}{S} \right\rfloor + 1$$
$$W' = \left\lfloor \frac{W + 2P - K}{S} \right\rfloor + 1$$

3.2

Solution:

Considering output padding of P', we can reverse what we did for the derviation in 3.1.

$$\begin{aligned} H' = \left\lfloor \frac{H + 2P - K}{S} \right\rfloor + 1 \rightarrow H' - 1 = \left\lfloor \frac{H + 2P - K}{S} \right\rfloor \\ \rightarrow S(H' - 1) + P' = H + 2P - K \\ \rightarrow H = S(H' - 1) - 2P + K + P' \\ W' = \left\lfloor \frac{W + 2P - K}{S} \right\rfloor + 1 \rightarrow W' - 1 = \left\lfloor \frac{W + 2P - K}{S} \right\rfloor \\ \rightarrow S(W' - 1) + P' = W + 2P - K \\ \rightarrow W = S(W' - 1) - 2P + K + P' \end{aligned}$$

Therefore,

$$\begin{aligned} P' &= H - S(H' - 1) + 2P - K \\ &= W - S(W' - 1) + 2P - K \end{aligned}$$

3.3

Solution:

Part 1

From the given information,

$$Y = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix} \quad W = \begin{bmatrix} -1 & 2 \\ 3 & -4 \end{bmatrix}$$

We can see that $H' = 2, W' = 2$ and $K = 2$. We are also given from the question that $S = 1, P = 0, P' = 0$. Thus, we can derive the original size from the derivation in 3.2,

$$\begin{aligned} H &= 1 \cdot (2 - 1) - 2 \cdot 0 + 2 + 0 = 3 \\ W &= 1 \cdot (2 - 1) - 2 \cdot 0 + 2 + 0 = 3 \end{aligned}$$

Then we start with,

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The first calculation involves $Y[1, 1]$,

$$\begin{aligned} Y[1,1] \times W[1,1] &= -4 \\ Y[1,1] \times W[1,2] &= 8 \\ Y[1,1] \times W[2,1] &= 12 \\ Y[1,1] \times W[2,2] &= -16 \end{aligned}$$

$$\begin{bmatrix} -4 & 8 & 0 \\ 12 & -16 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Moving along with $Y[1,2]$,

$$Y[1,2] \times W[1,1] = -3$$
$$Y[1,2] \times W[1,2] = 6$$
$$Y[1,2] \times W[2,1] = 9$$
$$Y[1,2] \times W[2,2] = -12$$

$$\begin{bmatrix} -4 & 5 & 6 \\ 12 & -7 & -12 \\ 0 & 0 & 0 \end{bmatrix}$$

Now with $Y[2,1]$,

$$Y[2,1] \times W[1,1] = -2$$
$$Y[2,1] \times W[1,2] = 4$$
$$Y[2,1] \times W[2,1] = 6$$
$$Y[2,1] \times W[2,2] = -8$$

$$\begin{bmatrix} -4 & 5 & 6 \\ 10 & -3 & -12 \\ 6 & -8 & 0 \end{bmatrix}$$

Finally, with $Y[2,2]$,

$$Y[2,2] \times W[1,1] = -1$$
$$Y[2,2] \times W[1,2] = 2$$
$$Y[2,2] \times W[2,1] = 3$$
$$Y[2,2] \times W[2,2] = -4$$

$$\begin{bmatrix} -4 & 5 & 6 \\ 10 & -4 & -10 \\ 6 & -5 & -4 \end{bmatrix}$$

$$\therefore \begin{bmatrix} -4 & 5 & 6 \\ 10 & -4 & -10 \\ 6 & -5 & -4 \end{bmatrix}$$

Part 2

We have that,

$$Y = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix} \quad W = \begin{bmatrix} -1 & 2 & -3 \\ 1 & -2 & 3 \\ 3 & 2 & -1 \end{bmatrix}$$

We can see that $H' = 3, W' = 3$ and $K = 3$. We are also given from the question that $S = 2, P = 1, P' = 1$. Thus, we can derive the original size from the derivation in 3.2,

$$H = 2 \cdot (3-1) - 2 \cdot 1 + 3 + 1 = 6$$
$$W = 2 \cdot (3-1) - 2 \cdot 1 + 3 + 1 = 6$$

For simplicity let's compute how we did in part 1 by multiplying W with each entry of Y,

$$\begin{bmatrix} -9 & 18 & -27 \\ 9 & -18 & 27 \\ 27 & 18 & -9 \end{bmatrix}, \begin{bmatrix} -8 & 16 & -24 \\ 8 & -16 & 24 \\ 24 & 16 & -8 \end{bmatrix}$$
$$\begin{bmatrix} -7 & 14 & -21 \\ 7 & -14 & 21 \\ 21 & 14 & -7 \end{bmatrix}, \begin{bmatrix} -6 & 12 & -18 \\ 6 & -12 & 18 \\ 18 & 12 & -6 \end{bmatrix}$$
$$\begin{bmatrix} -5 & 10 & -15 \\ 5 & -10 & 15 \\ 15 & 10 & -5 \end{bmatrix}, \begin{bmatrix} -4 & 8 & -12 \\ 4 & -8 & 12 \\ 12 & 8 & -4 \end{bmatrix}$$
$$\begin{bmatrix} -3 & 6 & -9 \\ 3 & -6 & 9 \\ 9 & 6 & -3 \end{bmatrix}, \begin{bmatrix} -2 & 4 & -6 \\ 2 & -4 & 6 \\ 6 & 4 & -2 \end{bmatrix}$$
$$\begin{bmatrix} -1 & 2 & -3 \\ 1 & -2 & 3 \\ 3 & 2 & -1 \end{bmatrix}$$

Now, given this information, let's construct a matrix applying stride 2,

$$\begin{bmatrix} -9 & 18 & -27-8 & 16 & -24-7 & 14 & -21 \\ 9 & -18 & 27+8 & -16 & 24+7 & -14 & 21 \\ 27-6 & 18+12 & -9+24-18-5 & 16+10 & -8+21-15-4 & 14+8 & -7-12 \\ 6 & -12 & 18+5 & -10 & 15+4 & -8 & 12 \\ 18-3 & 12+6 & -6+15-9-2 & 10+4 & -5+12-6-1 & 8+2 & -4-3 \\ 3 & -6 & 9+2 & -4 & 6+1 & -2 & 3 \\ 9 & 6 & -3+6 & 4 & -2+3 & 2 & -1 \end{bmatrix}$$

If we apply output padding $P' = 1$,

$$\begin{bmatrix} -9 & 18 & -35 & 16 & -31 & 14 & -21 & 0 \\ 9 & -18 & 35 & -16 & 31 & -14 & 21 & 0 \\ 21 & 30 & -8 & 26 & -6 & 22 & -19 & 0 \\ 6 & -12 & 23 & -10 & 19 & -8 & 12 & 0 \\ 15 & 18 & -2 & 14 & 0 & 10 & -7 & 0 \\ 3 & -6 & 11 & -4 & 7 & -2 & 3 & 0 \\ 9 & 6 & 3 & 4 & 1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Now let's remove the original padding $P = 1$ to match the size,

$$\begin{bmatrix} -18 & 35 & -16 & 31 & -14 & 21 \\ 30 & -8 & 26 & -6 & 22 & -19 \\ -12 & 23 & -10 & 19 & -8 & 12 \\ 18 & -2 & 14 & 0 & 10 & -7 \\ -6 & 11 & -4 & 7 & -2 & 3 \\ 6 & 3 & 4 & 1 & 2 & -1 \end{bmatrix}$$

4 Problem 4

4.1

Solution:

$$X = \begin{bmatrix} 3 & 1 & 2 & 6 & 5 \\ 6 & 8 & 1 & 7 & 9 \\ 2 & 7 & 4 & 2 & 3 \\ 8 & 3 & 5 & 4 & 1 \\ 1 & 5 & 2 & 7 & 6 \end{bmatrix} \quad W = \begin{bmatrix} -1 & 2 & -3 \\ 1 & -2 & 3 \\ 3 & 2 & -1 \end{bmatrix}$$

If we perform dilated convolution, with $D = 2, S = 2, P = 2$, then we are effectively doing the convolution with,

$$X' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 1 & 2 & 6 & 5 & 0 & 0 \\ 0 & 0 & 6 & 8 & 1 & 7 & 9 & 0 & 0 \\ 0 & 0 & 2 & 7 & 4 & 2 & 3 & 0 & 0 \\ 0 & 0 & 8 & 3 & 5 & 4 & 1 & 0 & 0 \\ 0 & 0 & 1 & 5 & 2 & 7 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, W' = \begin{bmatrix} -1 & 0 & 2 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & -1 \end{bmatrix}$$

Our expected output size it then,

$$H' = \left\lfloor \frac{5 + 2 \times 2 - 5}{2} \right\rfloor + 1 = 3$$
$$W' = \left\lfloor \frac{5 + 2 \times 2 - 5}{2} \right\rfloor + 1 = 3$$

Let's compute the convolution, for simplicity let '*' be the element-wise product and summation operator for matrices.

Step 1:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 1 & 2 \\ 0 & 0 & 6 & 8 & 1 \\ 0 & 0 & 2 & 7 & 4 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 2 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & -1 \end{bmatrix} = -6 + 6 + 4 - 4 = 0$$

Step 2:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 1 & 2 & 6 & 5 \\ 6 & 8 & 1 & 7 & 9 \\ 2 & 7 & 4 & 2 & 3 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 2 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & -1 \end{bmatrix} = 3 - 4 + 15 + 6 + 8 - 3 = 25$$

Step 3:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 6 & 5 & 0 & 0 \\ 1 & 7 & 9 & 0 & 0 \\ 4 & 2 & 3 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 2 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & -1 \end{bmatrix} = 2 - 10 + 12 + 6 = 10$$

Step 4:

$$\begin{bmatrix} 0 & 0 & 3 & 1 & 2 \\ 0 & 0 & 6 & 8 & 1 \\ 0 & 0 & 2 & 7 & 4 \\ 0 & 0 & 8 & 3 & 5 \\ 0 & 0 & 1 & 5 & 2 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 2 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & -1 \end{bmatrix} = 6 - 6 - 4 + 12 + 2 - 2 = 8$$

Step 5:

$$\begin{bmatrix} 3 & 1 & 2 & 6 & 5 \\ 6 & 8 & 1 & 7 & 9 \\ 2 & 7 & 4 & 2 & 3 \\ 8 & 3 & 5 & 4 & 1 \\ 1 & 5 & 2 & 7 & 6 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 2 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & -1 \end{bmatrix} = -3 + 4 - 15 + 2 - 8 + 9 + 3 + 4 - 6 = -10$$

Step 6:

$$\begin{bmatrix} 2 & 6 & 5 & 0 & 0 \\ 1 & 7 & 9 & 0 & 0 \\ 4 & 2 & 3 & 0 & 0 \\ 5 & 4 & 1 & 0 & 0 \\ 2 & 7 & 6 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 2 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & -1 \end{bmatrix} = -2 + 10 + 4 - 6 + 6 + 12 = 24$$

Step 7:

$$\begin{bmatrix} 0 & 0 & 2 & 7 & 4 \\ 0 & 0 & 8 & 3 & 5 \\ 0 & 0 & 1 & 5 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 2 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & -1 \end{bmatrix} = 4 - 12 - 2 + 6 = -4$$

Step 8:

$$\begin{bmatrix} 2 & 7 & 4 & 2 & 3 \\ 8 & 3 & 5 & 4 & 1 \\ 1 & 5 & 2 & 7 & 6 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 2 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & -1 \end{bmatrix} = -2 + 8 - 9 + 1 - 4 + 18 = 12$$

Step 9:

$$\begin{bmatrix} 4 & 2 & 3 & 0 & 0 \\ 5 & 4 & 1 & 0 & 0 \\ 2 & 7 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} -1 & 0 & 2 & 0 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 0 & -1 \end{bmatrix} = -4 + 6 + 2 - 12 = -8$$

Therefore, our result is,

$$\therefore \begin{bmatrix} 0 & 25 & 10 \\ 8 & -10 & 24 \\ -4 & 12 & -8 \end{bmatrix}$$

4.2

Solution:

With dilation D, let's consider what our effective kernel is. If we apply dilation of D, then each cell of our kernel is spaced by $D-1$, then since there are $K-1$ cell intersections (where two cells meet), our kernel size increases by $(K-1)(D-1)$ on one side. Therefore, our effective kernel size is,

$$K_{\text{eff}} = K + (K-1)(D-1)$$

Then now this is the same as convoluting the input with a kernel with size $K_{\text{eff}} \times K_{\text{eff}}$. Using our derivation from 3.1, we can get the expression of H', W' as the following,

$$\begin{aligned} H' &= \left\lfloor \frac{H + 2P - K_{\text{eff}}}{S} \right\rfloor + 1 \\ &= \left\lfloor \frac{H + 2P - (K + (K-1) \times (D-1))}{S} \right\rfloor + 1 \\ &= \left\lfloor \frac{H + 2P - (K-1) \times D - 1}{S} \right\rfloor + 1 \\ W' &= \left\lfloor \frac{W + 2P - K_{\text{eff}}}{S} \right\rfloor + 1 \\ &= \left\lfloor \frac{W + 2P - (K + (K-1) \times (D-1))}{S} \right\rfloor + 1 \\ &= \left\lfloor \frac{W + 2P - (K-1) \times D - 1}{S} \right\rfloor + 1 \end{aligned}$$

4.3

Solution:

We use a similar approach that was used in 3.2, to get expressions of H, W after transpose

convolution,

$$\begin{aligned}
H' = \left\lfloor \frac{H + 2P - (K-1) \times D - 1}{S} \right\rfloor + 1 \rightarrow H' - 1 &= \left\lfloor \frac{H + 2P - (K-1) \times D - 1}{S} \right\rfloor \\
&\rightarrow S(H'-1) + P' = H + 2P - (K-1) \times D - 1 \\
&\rightarrow H = S(H'-1) - 2P + (K-1) \times D + 1 + P' \\
W' = \left\lfloor \frac{W + 2P - (K-1) \times D - 1}{S} \right\rfloor + 1 \rightarrow W' - 1 &= \left\lfloor \frac{W + 2P - (K-1) \times D - 1}{S} \right\rfloor \\
&\rightarrow S(W'-1) + P' = W + 2P - (K-1) \times D - 1 \\
&\rightarrow W = S(W'-1) - 2P + (K-1) \times D + 1 + P'
\end{aligned}$$

Therefore,

$$\begin{aligned}
P' &= H - S(H'-1) + 2P - (K-1) \times D - 1 \\
&= W - S(W'-1) + 2P - (K-1) \times D - 1
\end{aligned}$$

5 Problem 5

5.1

Solution:

For convolution, padding nor the stride influences the number of learnable parameters. Also, since bias is ignored, the only thing that determines the number of learnable parameters is the kernel itself. The total number of learnable parameters would be the dimensionality of the kernel multiplied by the number of kernels. Also, since the input has a channel of C, and that there N kernels of $K \times K$, each kernel will be of dimension $C \times K \times K$ and we have N of them. Therefore, the number of learnable parameters is,

$$N \times K \times K \times C$$

5.2

Solution:

Once again, padding and stride do not influence the number of learnable parameters. In this case, when we apply group convolution with group size M, we will have a single group consisted of $\frac{N}{M}$ kernels, also each group would convolve with depth $\frac{C}{M}$ hence the total number of parameters becomes,

$$\underbrace{M}_{\text{number of groups}} \times \underbrace{\left(\frac{N}{M} \times K \times K \times \frac{C}{M}\right)}_{\text{learnable parameters per group}} = \frac{N \times K \times K \times C}{M}$$

Compared to standard (vanilla) convolution, the number of parameters in grouped convolution is reduced by a factor of M, the number of groups. This reduction significantly decreases the computational cost, making the model more efficient. Additionally, grouped convolution allows each group to learn distinct filters, enabling more diverse feature extraction.

6 Problem 6

6.1

Solution:

As stated in the previous problem, the total number of learnable parameters in vanilla convolution is,

$$N \times K \times K \times C$$

In terms of the number of operations, let's first look at one filter. One filter would do a total of $K \times K \times C$ multiplications for one location of the input. Since the input has $H \times W$ in height and width, for one input there is a total of $H \times W \times K \times K \times C$ multiplications. Since there are N filters and B inputs in a batch, the total number of multiplications are

$$B \times N \times H \times W \times K \times K \times C$$

When we perform the additions of convolving around one location of the input, since there are $K \times K \times C$ multiplications, we have $K \times C \times C - 1$ that follow. Now when we expand to the whole batch, the number of additions become,

$$B \times N \times H \times W \times (K \times K \times C - 1)$$

In summary,

$$\text{learnable parameters: } N \times K \times K \times C$$
$$\text{number of operations: } B \times N \times H \times W \times (2 \times K \times K \times C - 1)$$

Now, let's look at depthwise separable convolution. In the first step, where we perform a depth-wise $K \times K$ convolution with 1 filter per channel, our total number of learnable parameters is

$$C \times K \times K$$

We then have a total $H \times W$ locations we convolve with and we have B inputs in the batch along with C filters, the number of multiplications is

$$B \times C \times H \times W \times K \times K$$

Since for each filter at one location, we have $K \times K$ multiplications. Then we have $K \times K - 1$ additions, the total number of additions is then,

$$B \times C \times H \times W \times (K \times K - 1)$$

To summarize,

$$\text{learnable parameters: } C \times K \times K$$
$$\text{number of operations: } B \times C \times H \times W \times (2 \times K \times K - 1)$$

The second step involves point-wise convolution, where each filter will be of size $1 \times 1 \times C$ and we have N of them. This is because, the output from depth-wise convolution is $B \times H \times W \times C$ and we also want to make the output $B \times N \times H \times W$ to match vanilla convolution. The number of learnable parameters is then,

$$C \times N$$

In terms of operations, for one location there are C multiplications involved, followed with $C-1$ additions. Then, we get the total number of multiplications being

$$B \times H \times W \times N \times C$$

and the number of addition being

$$B \times H \times W \times N \times (C-1)$$

In summary,

learnable parameters: $C \times N$

number of operations: $B \times H \times W \times N \times (2 \times C - 1)$

Now let's compare the number of learnable parameters along with the number of operations for each type of convolution,

	vanilla	depth-wise separable
learnable parameters	NCK^2	$CK^2 + CN$
number of operations	$2BNHWCK^2 - BNHW$	$2BHWCK^2 - BHCW + 2BNHWC - BNHW$

In terms of learnable parameters, we can see that depth-wise separable convolution has much less parameters. Since, $NCK^2 > CK^2 + CN = NCK^2 \cdot (\frac{1}{N} + \frac{1}{K^2})$.

Let's also compare the number of operations,

$$\begin{aligned} 2BNHWCK^2 - BNHW &> 2BHWCK^2 - BHCW + 2BNHWC - BNHW \\ &= 2BNHWCK^2 \cdot (\frac{1}{N} - \frac{1}{2NK^2} + \frac{1}{K^2}) - BNHW \end{aligned}$$

It also shows that depth-wise separable convolution has a lower number of operations as well.