# COM S 573: Machine Learning

Homework #5

Abdurahman Mohammed

April 15, 2022

## Question 1

(a) The total number of trainable parameters is

$$K \times K \times C_{in} \times C_{out}$$

and we have kernel size=$(3 \times 3)$

So for the first convolution layer= $3 \times 3 \times 128 \times 256 = 294,192$

For the second convolution layer= $3 \times 3 \times 256 \times 512 = 1,179,648$

Hence, the total number of trainable parameters will be=$294,192 + 1,179,648 = 1,474,560$.

(b) The total number of multi-add operations = $K \times K \times M \times N \times W' \times D'$.
We know that W=D=64. Hence,

$$W' = ((W - K)/S) + 1 = ((63 - 3)/2) + 1 = 31$$

$$D' = ((D - K)/S) + 1 = ((63 - 3)/2) + 1 = 31$$

Now, for the first convolutional layer, the number of multi-add operations will be

$$3 \times 3 \times 128 \times 256 \times 31 \times 31 = 283,410,432.$$

And for the second convolutional layer

$$W' = ((W - K)/S) + 1 = ((31 - 3)/2) + 1 = 29$$

$$D' = ((D - K)/S) + 1 = ((31 - 3)/2) + 1 = 29$$

the total number of multi-add operations will be

$$3 \times 3 \times 256 \times 512 \times 29 \times 29 = 992,083,968$$

.

Hence the total number of multi-add operations will be

$$992,083,968 + 283,410,432 = 1,275,494,400$$

(c) First we will find the size of the output feature map as follows

$$128 \times 64 = 524,288$$

.

Then we will compute the size of the output of intermediate feature map

$$256 \times 31 \times 31 = 246,016$$

.

And the size of Output feature map will be

$$516 \times 29 \times 29 = 430,592$$

.

Then we find the change between the output and input memory requirements after the 2 convolutional layers.

$$\frac{430,592}{524,288} * 100 = 82.13\%$$

We can say that 83.13% of the input storage is required which is 17.87% less memory.

## Question 2

1.

$$\mu = \frac{1}{m}\sum_{i=1}^{m} z_i$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(z_i - \mu)^2$$

$$\hat{z_i} = \frac{(z_i - \mu)}{\sqrt{\sigma^2}}$$

For the first mini-batch we'll first calculate the $\mu$ and $\sigma^2$.
$\mu == \frac{1}{4}(12 + 14 + 14 + 12) = 13$
$\sigma^2 = \frac{1}{4}((12-13)^2 + (14-13)^2 + (14-13)^2 + (12-13)^2) = 1.$

So for the first mini-batch $\hat{z_i} = (-1, 1, 1, -1)$.

Now, we will continue with the second mini-batch. For the second mini-batch we'll calculate the $\mu$ and $\sigma^2$.
$\mu == \frac{1}{4}(0 + 10 + 10 + 0) = 5$
$\sigma^2 = \frac{1}{4}((0-5)^2 + (10-5)^2 + (10-5)^2 + (0-5)^2) = 25.$

So for the second mini-batch $\hat{z_i} = (-1, 1, 1, -1)$.

And, For the third mini batch we'll calculate the $\mu$ and $\sigma^2$.
$\mu == \frac{1}{4}(-5 + 5 + 5 - 5) = 0$
$\sigma^2 = \frac{1}{4}((-5-0)^2 + (5-0)^2 + (5-0)^2 + (-5-0)^2) = 25.$

So for the third mini-batch $\hat{z_i} = (-1, 1, 1, -1)$.

Then, we will finally get $\hat{Z_i} = \begin{pmatrix} -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \end{pmatrix}$

2. By using $\tilde{z_i} = \gamma\hat{z_i} + \beta$ For the first set of $\tilde{Z_i}$ we get the following.
   $\tilde{z_i} = (1,1,1)(-1,-1,-1) + (0,-10,10) = (-1,11,9)$.

   For the second set of $\tilde{Z_i}$ we get the following.
   $\tilde{z_i} = (1,1,1)(1,1,1) + (0,-10,10) = (1,-9,11)$.

   For the third set of $\tilde{Z_i}$ we get the following.
   $\tilde{z_i} = (1,1,1)(1,1,1) + (0,-10,10) = (-1,-11,-9)$.

   For the fourth set of $\tilde{Z_i}$ we get the following.
   $\tilde{z_i} = (1,1,1)(-1,-1,-1) + (0,-10,10) = (1,-9,11)$.

   And we finally get

$$\tilde{Z_i} = \begin{pmatrix} -1 & 1 & 1 & -1 \\ -11 & -9 & -9 & -11 \\ 9 & 11 & 11 & 9 \end{pmatrix}$$

3. The mean and variance keep changing for each mini batch that we have during training. Batch-normalization deals with the moving average and moving variance. On the other hand, When testing, it uses the mean and variance calculated during the training phase. It does not calculate the mean and variance for the test data.

4. During the testing phase, batch size only affects the mean and variance. During testing, the batch size has no direct effect on the results as we look at individual data points. But the batch can affect the values of mean and variance which were used in testing. Therefore, there might be an indirect effect of batch size on the testing results.

## Question 3

(a) With the kernels $W^{ij}$, we get the following output on convolution of the input.

$$y_{11} = w_1^{11}x_{11} + w_2^{11}x_{12} + w_3^{11}x_{13} + w_1^{21}x_{21} + w_2^{21}x_{22} + w_3^{21}x_{23}$$
$$y_{12} = w_1^{11}x_{12} + w_2^{11}x_{13} + w_3^{11}x_{14} + w_1^{21}x_{22} + w_2^{21}x_{23} + w_3^{21}x_{24}$$
$$y_{21} = w_1^{12}x_{11} + w_2^{12}x_{12} + w_3^{12}x_{13} + w_1^{22}x_{21} + w_2^{22}x_{22} + w_3^{22}x_{23}$$
$$y_{22} = w_1^{12}x_{12} + w_2^{12}x_{13} + w_3^{12}x_{14} + w_1^{22}x_{22} + w_2^{22}x_{23} + w_3^{22}x_{24}$$

And the 4 kernels used are the following.
$\left[w_1^{11}, w_2^{11}, w_3^{11}\right]$, $\left[w_1^{12}, w_2^{12}, w_3^{12}\right]$,$\left[w_1^{21}, w_2^{21}, w_3^{21}\right]$,$\left[w_1^{22}, w_2^{22}, w_3^{22}\right]$.

This gives us a total of 12 trainable parameters.

Now, let's try to show the convolutions in the form of fully connected layers.

$$\begin{bmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \end{bmatrix} = \begin{bmatrix} w_1^{11} & w_2^{11} & w_3^{11} & 0 & w_1^{21} & w_2^{21} & w_3^{21} & 0 \\ 0 & w_1^{11} & w_2^{11} & w_3^{11} & 0 & w_1^{21} & w_2^{21} & w_3^{21} \\ w_1^{12} & w_2^{12} & w_3^{12} & 0 & w_1^{22} & w_2^{22} & w_3^{22} & 0 \\ 0 & w_1^{12} & w_2^{12} & w_3^{12} & 0 & w_1^{22} & w_2^{22} & w_3^{22} \end{bmatrix} * \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \end{bmatrix}$$

This will give us

$$A = \begin{bmatrix} w_1^{11} & w_2^{11} & w_3^{11} & 0 & w_1^{21} & w_2^{21} & w_3^{21} & 0 \\ 0 & w_1^{11} & w_2^{11} & w_3^{11} & 0 & w_1^{21} & w_2^{21} & w_3^{21} \\ w_1^{12} & w_2^{12} & w_3^{12} & 0 & w_1^{22} & w_2^{22} & w_3^{22} & 0 \\ 0 & w_1^{12} & w_2^{12} & w_3^{12} & 0 & w_1^{22} & w_2^{22} & w_3^{22} \end{bmatrix}$$

(b) First we need to find the local gradients $\frac{\partial Y}{\partial x}$ in order to get$\frac{\partial L}{\partial x} =$ and then apply chain rule to get $\frac{\partial L}{\partial x} = \sum \frac{\partial L}{\partial Y} * \frac{\partial Y}{\partial x}$

$$\begin{aligned}
&\frac{\partial y_{11}}{\partial x_{11}} = w_1^{11} \ \& \ \frac{\partial y_{21}}{\partial x_{11}} = w_1^{12} \\
&\frac{\partial y_{11}}{\partial x_{12}} = w_2^{11} \ \& \ \frac{\partial y_{21}}{\partial x_{12}} = w_1^{11} \ \& \ \frac{\partial y_{21}}{\partial x_{12}} = w_2^{12} \ \& \ \frac{\partial y_{22}}{\partial x_{12}} = w_1^{12} \\
&\frac{\partial y_{11}}{\partial x_{13}} = w_3^{11} \ \& \ \frac{\partial y_{21}}{\partial x_{13}} = w_2^{11} \ \& \ \frac{\partial y_{21}}{\partial x_{13}} = w_3^{12} \ \& \ \frac{\partial y_{22}}{\partial x_{13}} = w_2^{12} \\
&\frac{\partial y_{12}}{\partial x_{14}} = w_3^{11} \ \& \ \frac{\partial y_{22}}{\partial x_{14}} = w_3^{12} \\
&\frac{\partial y_{11}}{\partial x_{21}} = w_1^{21} \ \& \ \frac{\partial y_{21}}{\partial x_{21}} = w_1^{22} \\
&\frac{\partial y_{11}}{\partial x_{22}} = w_3^{21} \ \& \ \frac{\partial y_{12}}{\partial x_{22}} = w_1^{21} \ \& \ \frac{\partial y_{21}}{\partial x_{22}} = w_2^{22} \ \& \ \frac{\partial y_{22}}{\partial x_{23}} = w_2^{22} \\
&\frac{\partial y_{11}}{\partial x_{23}} = w_3^{21} \ \& \ \frac{\partial y_{12}}{\partial x_{23}} = w_2^{21} \ \& \ \frac{\partial y_{21}}{\partial x_{23}} = w_3^{22} \ \& \ \frac{\partial y_{22}}{\partial x_{23}} = w_2^{22} \\
&\frac{\partial y_{12}}{\partial x_{24}} = w_3^{21} \ \& \ \frac{\partial y_{22}}{\partial x_{24}} = w_3^{22}
\end{aligned}$$

After applying chain rule we get

$$\begin{aligned}
&\frac{\partial L}{\partial x} = \sum \frac{\partial L}{\partial y} * \frac{\partial y}{\partial x} \\
&\frac{\partial L}{\partial x_{11}} = \frac{\partial L}{\partial y_{11}} * w_1^{11} + \frac{\partial L}{\partial y_{21}} * w_1^{12} \\
&\frac{\partial L}{\partial x_{12}} = \frac{\partial L}{\partial y_{11}} * w_2^{11} + \frac{\partial L}{\partial y_{12}} * w_1^{11} + \frac{\partial L}{\partial y_{21}} * w_2^{12} + \frac{\partial L}{\partial y_{22}} * w_1^{12} \\
&\frac{\partial L}{\partial x_{13}} = \frac{\partial L}{\partial y_{11}} * w_3^{11} + \frac{\partial L}{\partial y_{12}} * w_2^{11} + + \frac{\partial L}{\partial y_{21}} * w_3^{12} + \frac{\partial L}{\partial y_{22}} * w_2^{12} \\
&\frac{\partial L}{\partial x_{14}} = \frac{\partial L}{\partial y_{11}} * w_3^{11} + \frac{\partial L}{\partial y_{21}} * w_1^{22} \\
&\frac{\partial L}{\partial x_{21}} = \frac{\partial L}{\partial y_{11}} * w_1^{21} + \frac{\partial L}{\partial y_{12}} * w_1^{21} \\
&\frac{\partial L}{\partial x_{22}} = \frac{\partial L}{\partial y_{11}} * w_2^{21} + \frac{\partial L}{\partial y_{12}} * w_2^{21} + \frac{\partial L}{\partial y_{21}} * w_2^{22} + \frac{\partial L}{\partial y_{22}} * w_1^{22} \\
&\frac{\partial L}{\partial x_{23}} = \frac{\partial L}{\partial y_{11}} * w_3^{21} + \frac{\partial L}{\partial y_{22}} * w_2^{21} + \frac{\partial L}{\partial y_{21}} * w_3^{22} + \frac{\partial L}{\partial y_{22}} * w_3^{22} \\
&\frac{\partial L}{\partial x_{24}} = \frac{\partial L}{\partial y_{12}} * w_3^{21} + \frac{\partial L}{\partial y_{22}} * w_3^{22}
\end{aligned}$$

Therefore, we can write $\frac{\partial L}{\partial y}$ as follows.

$$\begin{bmatrix} \frac{\partial L}{\partial x_{11}} \\ \frac{\partial L}{\partial x_{12}} \\ \frac{\partial L}{\partial x_{13}} \\ \frac{\partial L}{\partial x_{14}} \\ \frac{\partial L}{\partial x_{21}} \\ \frac{\partial L}{\partial x_{22}} \\ \frac{\partial L}{\partial x_{23}} \\ \frac{\partial L}{\partial x_{24}} \end{bmatrix} = \begin{bmatrix} w_1^{11} & 0 & w_1^{12} & 0 \\ w_2^{11} & w_1^{11} & w_2^{12} & w_1^{12} \\ w_3^{11} & w_2^{11} & w_3^{12} & w_2^{12} \\ 0 & w_3^{11} & 0 & w_3^{12} \\ w_1^{21} & 0 & w_1^{22} & 0 \\ w_2^{21} & w_1^{21} & w_2^{22} & w_1^{22} \\ w_3^{21} & w_2^{21} & w_3^{22} & w_2^{22} \\ 0 & w_3^{21} & 0 & w_3^{22} \end{bmatrix} * \begin{bmatrix} \frac{\partial L}{\partial y_{11}} \\ \frac{\partial L}{\partial y_{12}} \\ \frac{\partial L}{\partial y_{21}} \\ \frac{\partial L}{\partial y_{22}} \end{bmatrix}$$

We can see that $B = A^T$.

(c) Yes, in order to get $\frac{\partial L}{\partial x}$, we will do the following $\frac{\partial L}{\partial x} = B * \frac{\partial L}{\partial Y}$. And this corresponds to $\frac{\partial L}{\partial Y}$. If we write the convolution as a fully connected layers, the kernels would look like

$$B = \begin{bmatrix} w_1^{11} & 0 & w_1^{12} & 0 \\ w_2^{11} & w_1^{11} & w_2^{12} & w_1^{12} \\ w_3^{11} & w_2^{11} & w_3^{12} & w_2^{12} \\ 0 & w_3^{11} & 0 & w_3^{12} \\ w_1^{21} & 0 & w_1^{22} & 0 \\ w_2^{21} & w_1^{21} & w_2^{22} & w_1^{22} \\ w_3^{21} & w_2^{21} & w_3^{22} & w_2^{22} \\ 0 & w_3^{21} & 0 & w_3^{22} \end{bmatrix}$$

To show it as a kernel matrix, we can say that the kernel is $W_0^{ij} = \left[w_3^{ij}, w_2^{ij}, w_1^{ij}\right]$ for i,j=1,2 and we take the $\frac{\partial L}{\partial Y}$ matrices as the input. This shows that it is equivalent to a 180° rotation of $W^{ij}$ given in the question.

## Question 4

(a)
```
Epoch 0 training loss: 2.077 training accuracy: 22.906%
Epoch 1 training loss: 1.602 training accuracy: 41.276%
Epoch 2 training loss: 1.419 training accuracy: 48.772%
Epoch 3 training loss: 1.304 training accuracy: 53.294%
Epoch 4 training loss: 1.217 training accuracy: 56.860%
Epoch 5 training loss: 1.147 training accuracy: 59.416%
Epoch 6 training loss: 1.093 training accuracy: 61.576%
Epoch 7 training loss: 1.040 training accuracy: 63.584%
Epoch 8 training loss: 1.000 training accuracy: 64.964%
Epoch 9 training loss: 0.955 training accuracy: 66.506%
Epoch 10 training loss: 0.919 training accuracy: 68.048%
Epoch 11 training loss: 0.882 training accuracy: 68.978%
Epoch 12 training loss: 0.847 training accuracy: 70.258%
Epoch 13 training loss: 0.817 training accuracy: 71.342%
Epoch 14 training loss: 0.786 training accuracy: 72.354%
Epoch 15 training loss: 0.758 training accuracy: 73.248%
Epoch 16 training loss: 0.732 training accuracy: 74.442%
Epoch 17 training loss: 0.704 training accuracy: 75.184%
Epoch 18 training loss: 0.678 training accuracy: 76.064%
Epoch 19 training loss: 0.656 training accuracy: 76.722%
Accuracy of the network on the 10000 test set: 65.09%
```

After building the default LeNet and running, we have found a training accuracy of 80.21% and a test accuracy of 65.41%. This shows us there is a slight overfitting in our model.

(b)
```
Epoch 0 training loss: 1.650 training accuracy: 39.832%
Epoch 1 training loss: 1.294 training accuracy: 53.550%
Epoch 2 training loss: 1.159 training accuracy: 58.688%
Epoch 3 training loss: 1.067 training accuracy: 62.102%
Epoch 4 training loss: 1.003 training accuracy: 64.690%
Epoch 5 training loss: 0.947 training accuracy: 66.688%
Epoch 6 training loss: 0.901 training accuracy: 68.266%
Epoch 7 training loss: 0.862 training accuracy: 69.770%
Epoch 8 training loss: 0.826 training accuracy: 70.914%
Epoch 9 training loss: 0.792 training accuracy: 72.018%
Epoch 10 training loss: 0.763 training accuracy: 72.954%
Epoch 11 training loss: 0.734 training accuracy: 74.370%
Epoch 12 training loss: 0.706 training accuracy: 75.182%
Epoch 13 training loss: 0.686 training accuracy: 75.784%
Epoch 14 training loss: 0.659 training accuracy: 76.898%
Epoch 15 training loss: 0.636 training accuracy: 77.422%
Epoch 16 training loss: 0.618 training accuracy: 78.302%
Epoch 17 training loss: 0.595 training accuracy: 78.874%
Epoch 18 training loss: 0.580 training accuracy: 79.404%
Epoch 19 training loss: 0.562 training accuracy: 80.206%
Accuracy of the network on the 10000 test set: 65.41%
```

Introducing batch normalization as a form of regularization showed a very slight increase in our testing accuracy. The training accuracy is 80.21% and the test accuracy is 65.41%.

(c)
```
Epoch 0 training loss: 1.754 training accuracy: 35.310%
Epoch 1 training loss: 1.451 training accuracy: 47.886%
Epoch 2 training loss: 1.338 training accuracy: 52.414%
Epoch 3 training loss: 1.261 training accuracy: 55.598%
Epoch 4 training loss: 1.204 training accuracy: 57.572%
Epoch 5 training loss: 1.162 training accuracy: 59.322%
Epoch 6 training loss: 1.124 training accuracy: 60.878%
Epoch 7 training loss: 1.097 training accuracy: 61.510%
Epoch 8 training loss: 1.069 training accuracy: 62.864%
Epoch 9 training loss: 1.042 training accuracy: 63.730%
Epoch 10 training loss: 1.018 training accuracy: 64.530%
Epoch 11 training loss: 1.001 training accuracy: 65.086%
Epoch 12 training loss: 0.986 training accuracy: 65.580%
Epoch 13 training loss: 0.965 training accuracy: 66.270%
Epoch 14 training loss: 0.950 training accuracy: 66.862%
Epoch 15 training loss: 0.939 training accuracy: 67.178%
Epoch 16 training loss: 0.927 training accuracy: 67.624%
Epoch 17 training loss: 0.915 training accuracy: 68.318%
Epoch 18 training loss: 0.905 training accuracy: 68.334%
Epoch 19 training loss: 0.893 training accuracy: 68.990%
Accuracy of the network on the 10000 test set: 67.67%
```

Now, we used dropouts with rate 0.3, we can see that the the training accuracy has become 68.99% and the testing accuracy being 67.67%. Even though the training accuracy has dropped, we can see that we have much less overfitting in our model.