Deep learning - Assignment 3

Akshay R -A20442409

CONTAINS THEORITICAL QUESTIONS

Theoretical questions;

1.

GIVEN IMAGE :

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad G = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

CONVOLUTION FILTER : $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

CONVOLUTION ON DIFFERENT CHANNELS :

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad R = \begin{bmatrix} 9 & 9 \\ 9 & 9 \end{bmatrix} \quad G = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix} \quad G = \begin{bmatrix} 18 & 18 \\ 18 & 18 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 18 & 18 \\ 27 & 27 \end{bmatrix}$$

OUTPUT : (R+G+B)

$$\Rightarrow \begin{bmatrix} 9+18+18 & 9+18+18 \\ 9+18+27 & 9+18+27 \end{bmatrix} = \begin{bmatrix} 45 & 45 \\ 54 & 54 \end{bmatrix}$$

2. WITH ZERO PADDING

$$R = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow R = \begin{bmatrix} 4 & 6 & 6 & 4 \\ 6 & 9 & 9 & 6 \\ 6 & 9 & 9 & 6 \\ 4 & 6 & 6 & 4 \end{bmatrix}$$

INPUT

$(\text{OUTPUT})_R$

INPUT

G :

0	0	0	0	0	0
0	2	2	2	2	0
0	2	2	2	2	0
0	2	2	2	2	0
0	2	2	2	2	0
0	0	0	0	0	0

(OUTPUT)$_G$

G :

8	12	12	8
12	18	18	12
12	18	18	12
8	12	12	8

INPUT

B :

0	0	0	0	0	0
0	1	1	1	1	0
0	2	2	2	2	0
0	3	3	3	3	0
0	4	4	4	4	0
0	0	0	0	0	0

(OUTPUT)$_B$

B :

6	9	9	6
12	18	18	12
18	27	27	18
14	21	21	14

FINAL_OUTPUT : (R+G+B):

18	27	27	18
30	45	45	30
36	54	54	36
26	39	39	26

3.

INPUT

R :

0	0	0	0	0	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	1	1	0
0	0	0	0	0	0

FILTER

1	0	1	0	1
0	0	0	0	0
1	0	1	0	1
0	0	0	0	0
1	0	1	0	1

RED_OUTPUT

4	4
4	4

INPUT

G :

0	0	0	0	0	0
0	2	2	2	2	0
0	2	2	2	2	0
0	2	2	2	2	0
0	2	2	2	2	0
0	0	0	0	0	0

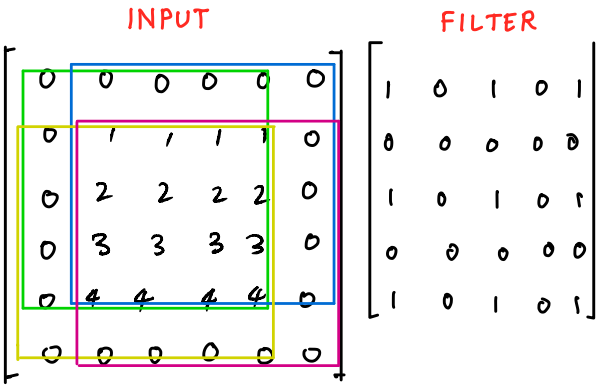
FILTER

1	0	1	0	1
0	0	0	0	0
1	0	1	0	1
0	0	0	0	0
1	0	1	0	1

GREEN_OUTPUT

8	8
8	8

B.

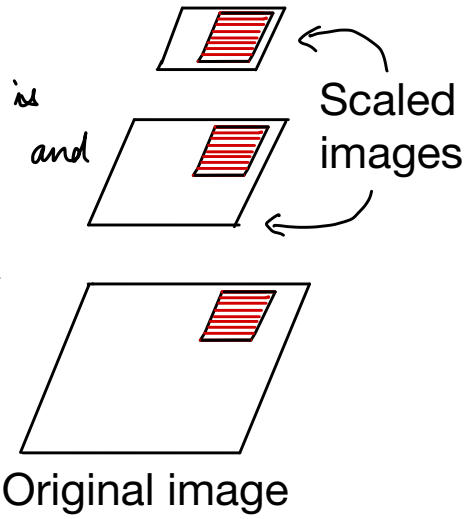


4. Template matching is performed through convolution by selecting the filter as the template itself. If at all the feature we are trying to capture matches with any part of the image then the values at those regions have higher value compared to the areas where the template doesn't match. This is how convolution is interpreated as template matching.

5. Multiple scale analysis can be achieved through the same window by pooling.

As seen in the figure alongside it is seen that by retaining the original filter and scaling the images the same template (filter) can be detected over different image scale.

This is how multiple scale analysis is performed through the same window.



6- Performing multiple convolutions results in spacial resolution decreare as the edges of the images gets clipped. This is compensated by increasing the number of filters resulting in increasing the channels.
More number of channels results in better feature extraction because more filters captures more information.

7. Given tensor 128 (W) x 128 x 32 filter size 3 (K) x 3 x 32
Number of filters = 16. Padding: 0 (P) Stride: 1 (S)

we have the formula ⇒

$$\text{Output size} = \frac{w - k + 2p}{s} + 1 = \frac{128 - 3 + 0}{1} + 1 = 126$$

Output tensor = 126 x 126 x 16

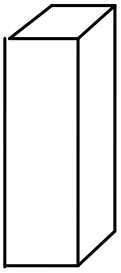
8. Same data as previous question with stride : 2

$$\text{Output size} = \frac{w - k + 2p}{s} = \frac{128 - 3 + 0}{2} + 1 = 64$$

Output tensor = 64 x 64 x 16

9. To perform channel reduction we perform convolution using 1x1 filters (desired number to achieve o/p size) over the original image. The process is illustrated in the diagram below.

Eg:



28x28x32

(*) zero padding + 16 convoluted filters of size 1x1x32 ⇒

28x28x16

10. Early and deeper convolutional layers are interpreted to complexities of the patterns that are being recognised. More convolutional layers means more features has to be extracted.
If we take the example of recognizing a human face compared to recognizing a simple shape like a circle, the number of convolutional layers needed to recognize a human face would be deeper compared to recognizing a circle.

11. Given image:

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad G = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$$

Convolution filter: $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

After pooling we get the following results

$$R = \begin{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \end{bmatrix} \quad R = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad G = \begin{bmatrix} \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} & \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \\ \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} & \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \end{bmatrix} \quad G = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

$$B = \begin{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} & \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \\ \begin{bmatrix} 3 & 3 \\ 4 & 4 \end{bmatrix} & \begin{bmatrix} 3 & 3 \\ 4 & 4 \end{bmatrix} \end{bmatrix} \quad B = \begin{bmatrix} 2 & 2 \\ 4 & 4 \end{bmatrix}$$

12. The main purpose of pooling is to scale the images after every convolutional layer to reduce the amount of features that are extracted and to prevent overfitting.

13. Data augmentation is performed to generate duplicates of existing data with variations in the scale, contrast, angle or color of the original image.
Data augmentation is performed when there is less data to work with. This improves the generalization of the model and reduces overfitting.

14. Transfer learning is a technique used where we use a pre trained model as the base to train a desired model.
How transfer learning works is, we use a model that is trained over a large dataset and use its updated weights as the weights of our model. In essence we are transfering what model 1 learnt to model 2.
This is mainly performed we we do not have enough data for a model to learn. This is when transfer learning is most useful.

15. When we use transfer learning we only use part of the pre trained model or use additional layers to meet our needs. While traing our network if the weights of the pretrained model are not frozen, our additional layers tend to change the weights of the pre trained model where the essence of transfer learning is lost.
Hence, we freeze the weights of the pre trained model.

16. After training the fully connected layers with frozen pre trained model we fine tune by unfreezing the top layers of the pre trained model. This allows updation of weights of the top layers more suited to the problem statement we seek to address.

We then train the unfrozen weights with our custom network to obtain a desired model.

17. With the use of transfer learning we tackle the problem of traing our model with quality features. But these models that are pretrained tend to use many parameters. For example VGG16 itself requires 2359808 parameters The main purpose of using inception blocks is to reduce the number of parameters used for such deep models.

For example, Googlenet uses 22 layers in comparison with 16 layers of VGG16 but uses only 6.7 million parameters which is a considerate improvement over VGG16's architecture.

18. Advantages of residual blocks are:
- → It helps with vanishing gradients as the input is added to the output of the residual block.
- → Zero weights in the block produces identity instead of destroying the signal. The network can learn to zero blocks to eleminate non needed layers.
- → Quicker training as gradients are passed directly through skip connections.

19. To visualize intermediate activations first a new model is created from the existing one and use the model class instead of the sequential class where the model allows multiple outputs.

Visualizing intermediate activations allows us to know what this filter is doing to the image.

20. Filters can be interpreted as templates that are being matched and hence visualizing weights allows us to know the template of the filter.
Using gradient ascent find the input that will maximize the response of the filter. Visualize the input the minimizes the loss or maximizes the response.

21. To compute the heat map we first an image to to the network and compute gradients at selected output node with respect to each channel of the target layer where activations need to be computed.
Compute the average gradients at each channel by computing the weighted sum of gradient magnitude.
Finally superimpose activations on input image to obtain the heatmap.

While interpreting the heatmap we consider the area in the image that is most red to be the area where an important feature was extracted that lead to classification.

END