Gabriel BARANES

Deep learning HW4 CS577

Q°1 :

R

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

G

2	2	2	2
2	2	2	2
2	2	2	2
2	2	2	2

B

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

filter →

1	1	1
1	1	1
1	1	1

9	9
9	9

18	18
18	18

18	18
27	27

45	45
54	54

convolution of the image without zero padding.

2) with zero padding:

4	6	6	4
6	9	9	6
6	9	9	6
4	6	6	4

+

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

+

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

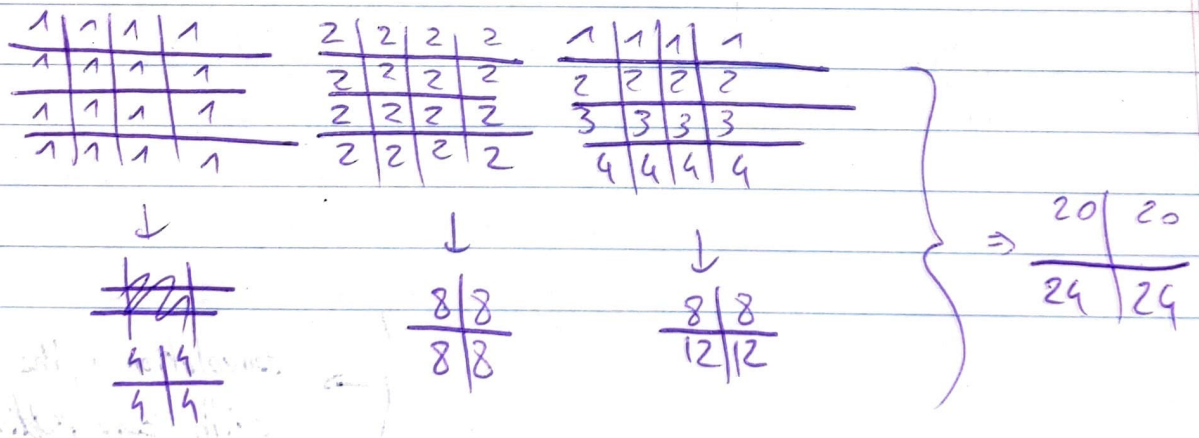
18	27	27	18
30	45	45	30
36	54	54	36
26	39	39	26

→ convolution of the image with zero padding.

5) We take the image, that's the original level of the pyramid and then we sample the image (every 4 pixels for example) and after sampling it, we get another value: 2nd level of the pyramide. Each level of the pyramid is half the size of the one just before.

4) Template matching is implemented through twodimensionnal convolution. In convolution, the value of an output pixel is computed by multiplying elements of two matrices and summing the results. One of those matrixes represents the image itself, while the other matrix is the template, which is known as a convolutional kernel.

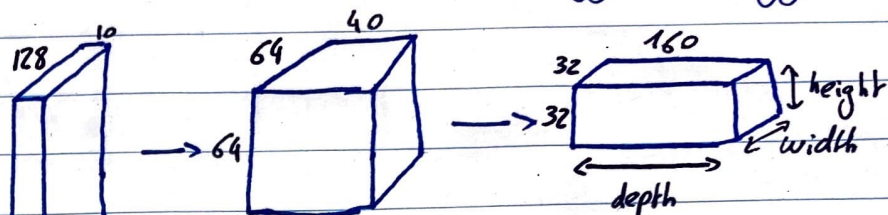
3)



Q°6:

We use multiple convolutional layers and make the image's resolution smaller and smaller to be able to detect bigger and bigger objects. (Pyramid model).

The Example:

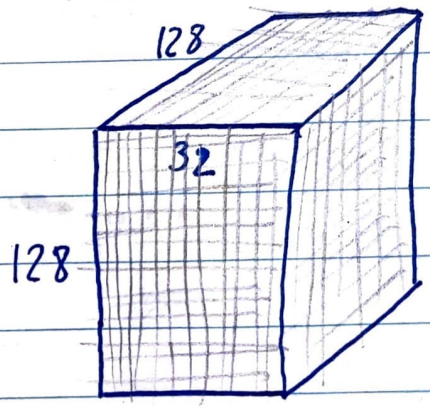


As we go deeper and deeper into the network, the resolution: height and width decreases. At the same time that we reduce the spacial resolution, we increase the depth.

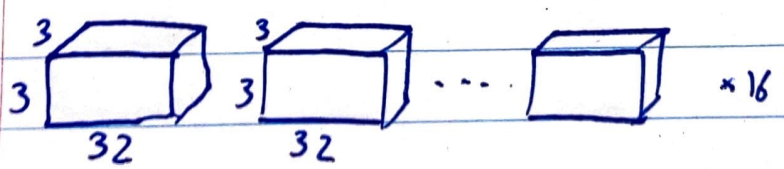
When we reduce the spacial resolution, we loose information. To compensate the low resolution spacial dimension, we get higher depth.

As spacial dimension decrease, depth increase to compensate for reduction coefficient (Keep the same number of coefficient).

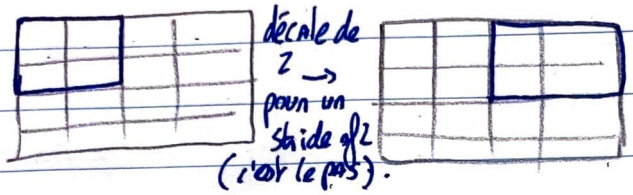
7) tensor $128 \times 128 \times 32$ and 16 convolution layers $3 \times 3 \times 32$



Size of the Resulting tenson without zero padding: $126 \times 126 \times 16$



8) with a stride of two, without zero padding, the size of the resulting tensor is 4 $63 \times 63 \times 16$.



9)

A 1x1 convolution simply maps an input pixel with all it's channels to an output pixel not looking at anything around itself. It is often used to reduce the number of depth channels, since it is often very slow to multiply volumes with extremely large depth.

Input 256 depth → 1x1 convolution: 64 depth → 4x4 convolution : 256 depth.

10) The early layers (in the network) extract simple features, and as we progress, we extract more and more complex features (= deeper layers)
A convolution layer network then have early layers and deep layers

11) max pooling with image from Q°1)

1	1	+	2	2	+	2	2	=	5	5
1	1		2	2		4	4		7	7

12) Pooling is used to reduce the spatial size of the image
Pooling reduces the hight and width but not the depth of the image

13) Data Augmentation is used to augment data for better generalization using the image data generation.
It is useful when we work with images. to reduce the overfitting

14) Transfer learning: we use a pretrained convnet trained on a large set (1 million images for example). Used for classification but also for object segmentation and other tasks.

Transfer learning is a way to speed up deep learning training. It helps solve complex problems with pre-existing knowledge.

15) The task of fine-tuning a network is to tweak the parameters of an already existing trained network so that it adapts to the new task at hand. The initial layer learns very general features and as we go higher up the network, the layers tend to learn ~~more~~ patterns more specific to the task it is being trained on. Thus, for fine-tuning, we want to keep the initial layer intact and freeze them. ~~and retrain the layers later on~~

16) After training the fully connected layer, we unfreeze some top layers in "conv-base" and retrain to allow the model to fit better the data.

Steps:
- add custom network on top of trained layers.
- Freeze trained layers
- train custom network
- unfreeze top layers in the base network
- jointly train the custom network and unfrozen layers.

17)
An inception block aims to approximate an optimal local sparse structure in a CNN. It allows for us to use multiple type of filter size instead of being restricted to a single filter size.

18)
Essentially, residual blocks allow memory or information to flow from initial to las layers.

19) To visualize intermediate activation function, we simply plot what each filter has extracted given an input.
Visualizing intermediate activation give a view into how an input is decomposed into the different filters learned by the network

20) A way of learning about what the convolution network is looking for in the images is to visualize the convolution layer filter.
By displaying the network layer filter, we can learn about the pattern to which each filter will correspond to.

21) To visualize the heatmap, we use Grad-CAM (Gradient class Activation Map). The idea behind it is to find the importance of a certain class in the model, uptake its gradient with respect to the final convolutionnal layer and then weight it against the output of this layer.
It is useful because it helps understand if the neural network is looking at appropriate parts of the image, or if the neural network is cheating.