cs577 Assignment 4

Jorge Gonzalez Lopez
A20474413
Department of Computer Science
Illinois Institute of Technology
April 21, 2021

Part 1 (theoretical questions)

1. Image RGB: 4x4x3 $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$, $\begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix}$, $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$

Filter: 3x3x3 $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times 3$

Value 1 = Value 2 = $(1 \cdot 1) \cdot 9 + (2 \cdot 1) \cdot 9 + (1 \cdot 1) \cdot 3 + (2 \cdot 1) \cdot 3 + (3 \cdot 1) \cdot 3 =$

$= 45$

Value 3 = Value 4 = $(1 \cdot 1) \cdot 9 + (2 \cdot 1) \cdot 9 + (2 \cdot 1) \cdot 3 + (3 \cdot 1) \cdot 3 + (4 \cdot 1) \cdot 3 =$

$= 54$

Output = $\begin{bmatrix} 45 & 45 \\ 54 & 54 \end{bmatrix}$ $(\rightarrow 2 \times 2 \times 1)$

2. padding ⇒ zero padding ⇒ output size = $\frac{4 - 3 + 2 \cdot p + 1}{1} = 4 \Rightarrow \boxed{p = 1}$ ⇒ Image $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ ...

Value 1 : $(1 \cdot 1) \cdot 4 + (2 \cdot 1) \cdot 4 + (1 \cdot 1) \cdot 2 + (2 \cdot 1) \cdot 2 = 18$ = Value 4

Value 2 : $(1 \cdot 1) \cdot 6 + (2 \cdot 1) \cdot 6 + (1 \cdot 1) \cdot 3 + (2 \cdot 1) \cdot 6 = 27$ = Value 3

Value 5 : $(1 \cdot 1) \cdot 6 + (2 \cdot 1) \cdot 6 + (1 \cdot 1) \cdot 2 + (2 \cdot 1) \cdot 2 + (3 \cdot 1) \cdot 2 = 30$ = Value 8

Value 6 = Value 7 = Ej 1 and Value 10 = Value 11 = Ej 1.

Value 9 = $(1 \cdot 1) \cdot 6 + (2 \cdot 1) \cdot 6 + (2 \cdot 1) \cdot 2 + (3 \cdot 1) \cdot 2 + (4 \cdot 1) \cdot 2 = 36$ = Value 12

Value 13 = $(1 \cdot 1) \cdot 4 + (2 \cdot 1) \cdot 4 + (1 \cdot 3) \cdot 2 + (1 \cdot 4) \cdot 2 = 26$ = Value 16

Value 14 = $(1 \cdot 1) \cdot 6 + (2 \cdot 1) \cdot 6 + (3 \cdot 1) \cdot 3 + (4 \cdot 1) \cdot 3 = 39$ = Value 15

Output = $\begin{bmatrix} 18 & 27 & 27 & 18 \\ 30 & 45 & 45 & 30 \\ 36 & 54 & 54 & 36 \\ 26 & 39 & 39 & 26 \end{bmatrix}$

3. Dilation = 2 ⇒ $(F *_{\ell} K)(p) = \sum_{s + \ell t = p} F(s)\, K(t)$

$\begin{cases} \text{Val } 1 = \text{Val } 2 = 4 \cdot 1 + 4 \cdot 2 + (2 \cdot 2) + (2 \cdot 4) = 24 \\ \text{Val } 3 = \text{Val } 4 = 4 \cdot 1 + 4 \cdot 2 + (2 \cdot 1) + (2 \cdot 3) = 20 \end{cases}$

Output = $\begin{bmatrix} 24 & 24 \\ 20 & 20 \end{bmatrix}$

4.

The template matching interpretation of convolution refers to understanding filters as templates that are being matched. The template is moved across the enitre image to compare its similarity with the covered window on the image.

5.

Multiple scale analysis is a set of techniques used to obtain valid approximation solutions to perturbation problems. For example, it allows detecting the same objects in images with different scales. Therefore, multiple scale analysis can be achieved with a pyramid representation, in which the image is repeatedly smoothed and subsampled, hence the objects in the image successively scale down with the consecutives size reduNctions.

6.

The way to compensate for saptial resolution decrease is increasing the number of filters (channels) in every successive convolutional layer. Threfore, even though the convolutional layers reduce the length and width of the input images, the depth increases tyring to keep the number of parameters constant to reduce the information loss.

7.

Input size -> 128 x 128 x 32

Filter size -> 3 x 3 x 32 (Number of filters = 16)

Zero padding and stride = 1

Output length/width = $\frac{Input\ size - filter\ size + 2 * padding}{stride} + 1 = \ 128 - 3 + 1 = 126$

Output size -> 126 x 126 x 16

8.

Input size -> 128 x 128 x 32

Filter size -> 3 x 3 x 32 (Number of filters = 16)

Zero padding and stride = 2

Output length/width = $\frac{Input\ size - filter\ size + 2 * padding}{stride} + 1 = \ \left\lfloor \frac{128-3}{2} \right\rfloor + 1 = 63$

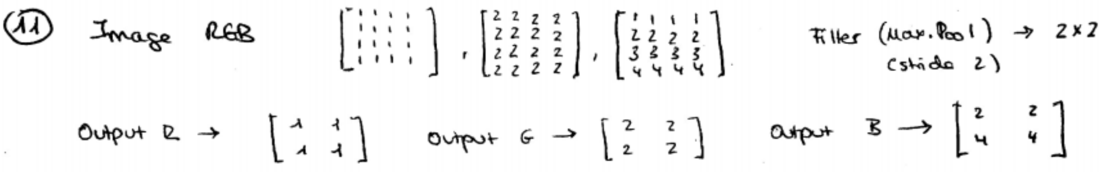
Output size -> 63 x 63 x 16

9.

With a 1x1 convolution with zero padding and a stride of 1, the length and width of the input signal remains constant. Hence, it is a pretty straight forward way of varying the number of channels (the depth) with the number of filters of an image keeping its original shape.

10.

Convolutional layers can be interpreted as “filters” that look through the image certain patterns or specific shapes. The difference between early and deeper convolutional layers is that the early layers find basic features and as the deep increases, the layers start detecting higher level features.



(11) Image RGB $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$, $\begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{bmatrix}$, $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix}$ Filter (Max.Pool) → 2×2 (stride 2)

Output R → $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ Output G → $\begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$ Output B → $\begin{bmatrix} 2 & 2 \\ 4 & 4 \end{bmatrix}$

12.

The main purpose of pooling is to reduce the spatial size of the input images to reduce the amount of parameters and computation in the network but operating on each channel independently to keep the depth unchanged.

13.

Data augmentation is a strategy that enables to significantly increase the diversity of data available for training models, without actually collecting new data. It is most useful when there are not many data available to train and test a model.

14.

The purpose of using transfer learning is to exploit the knowledge gained from a previous task to improve generalization about another. Again, it is most useful when the data available for a certain task is lacking.

15.

Freezing the coefficients of the pre-trained network means that the parameters of all those layers will not be updated during training. Therefore, even if the pre-trained network is very deep and complex, the training will be faster while keeping the increase in performance of the pre-trained network.

16.

After training the new layers added to the pre-trained network, the whole network can be unfrozen and trained again to fine-tune all the parameters (specifically the ones of the pre-trained network) to work and fit better to the new task instead of its original one.

17.

An inception block is a type of convolutional layer that aims to use multiple types of filters size and layers, instead of being restricted to a single layer with a fixed filter size. Hence, the output is a concatenation of all those layers so that the network can learn which layer and filter size is better instead of hardcoding it.

18.

A residual block is a type of network configuration in which there is a skip connection (the activation of one layer is sent directly to the output of the next layer). Therefore, it allows an increase of the depth of the networks as it avoids the problem of the vanishing gradients.

19.

Visualizing intermediate activations consists of displaying the feature maps that are output by various convolution and pooling layers in a network given a certain input. Its purpose is to be able to understand how successive layers transform their input, and for getting a first idea of the meaning of individual filters.

To do so it can be used the Keras class Model as it allows models with multiple outputs, unlike Sequential. Those outputs correspond to the activations of the successive layers of the model.

20.

The filters weights can be visualized by creating input images that maximize the activation of specific filters in a target layer (input image more correlated to the filter). Such images represent a visualization of the pattern that the filter responds to. Its purpose is to find the shapes or patterns that the filters are able to learn and identify in the images.

21.

The heatmap of class activation can be visualized with Keras using the class Grad-CAM. To generate the heatmap, it computes the outputs obtained by the network with a certain image, it calculates the gradients of the top predicted class and generate a map with their mean intensity values. To weight and get how important the different channels are for the top predicted class, the pooled gradients are used.

The purpose of this visualization is to determine the specific parts of an image that determine the result of the model predictions.