Algorithms and Application in Computer Vision - 046746

Homework #2

Alexander Shender 328626114

Vladimir Tchuiev 309206795

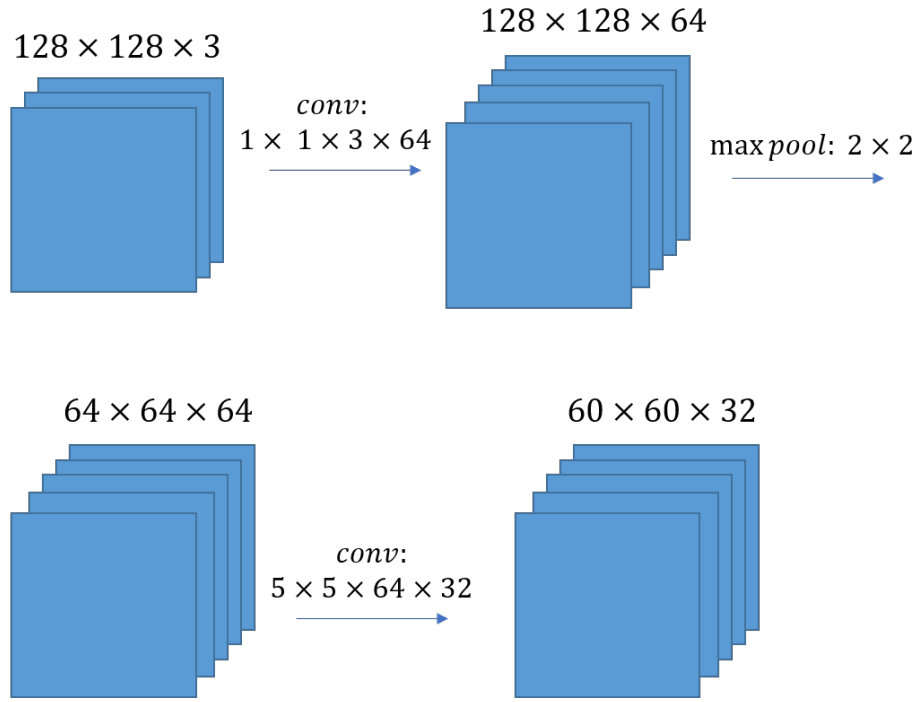
Technion - Israel Institute of Technology

I. Dry section

A. Question 1.

1. a.

The dimensions of the layers change in the following way:



2. b.

The convolution of the size 1X1X(?) performs convolution on the same pixes in different channels. The input image contains 3 channels in our case, thus the convolution of the size 1X1X3 fits perfectly to result in a block of new layers without changing size (no need for padding). One kernel results in an output layer of size 128×128, but since we have 64 kernels, the depth of the next layers block is 64, accordingly.

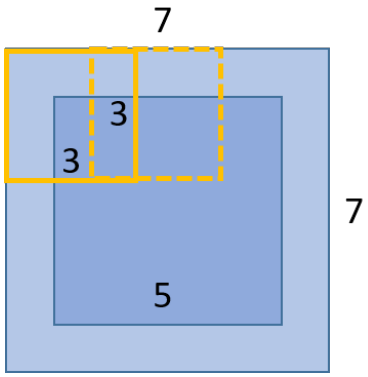
3. c.

Let's say, our normalized filter is the following:

$$\begin{bmatrix} 0.1 & 0.2 & 0.05 \\ 0.05 & 0.2 & 0.1 \\ 0.15 & 0.1 & 0.05 \end{bmatrix}$$

We choose 2 options for stride and padding:

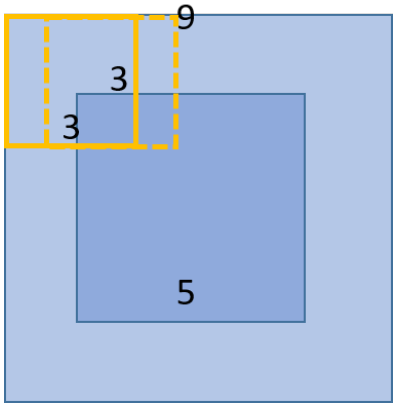
1. $stride = 2, padding = 1$ The image now has a dimensions of 9×9, and with a stride of 1 it gives an output dimensions: 3×3



Output result is the following:

$$\begin{bmatrix} 1.3 & 2.7 & 1.9 \\ 1.9 & 5.25 & 3.1 \\ 0.5 & 3.5 & 1.7 \end{bmatrix}$$

2. $stride = 1, padding = 2$ The image now has a dimensions of 7×7, but with stride of 2 it fits with the filter. Output dimensions: 7×7



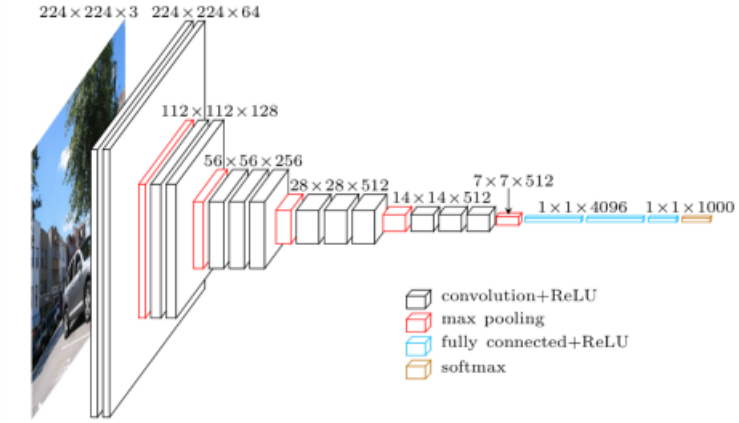
Output result is the following:

$$\begin{bmatrix} 0.2 & 0.45 & 1 & 0.8 & 1.15 & 0.45 & 0.45 \\ 0.55 & 1.3 & 2.0 & 2.7 & 2.65 & 1.9 & 0.45 \\ 0.6 & 2.1 & 3.3 & 5.15 & 4.6 & 2.6 & 1.45 \\ 0.4 & 1.9 & 3.75 & 5.25 & 5.145 & 3.1 & 1 \\ 0.2 & 1.3 & 3.5 & 4.7 & 4.0 & 3.35 & 1.45 \\ 0.05 & 0.5 & 2.05 & 3.5 & 2.55 & 1.7 & 0.5 \\ 0 & 0.05 & 0.5 & 1.5 & 1.6 & 1.2 & 0.4 \end{bmatrix}$$

. The code is provided in the appendix.

B. Question 2.

The architecture selected is the VGG16 architecture. The image found in the internet which describes the structure is the following:

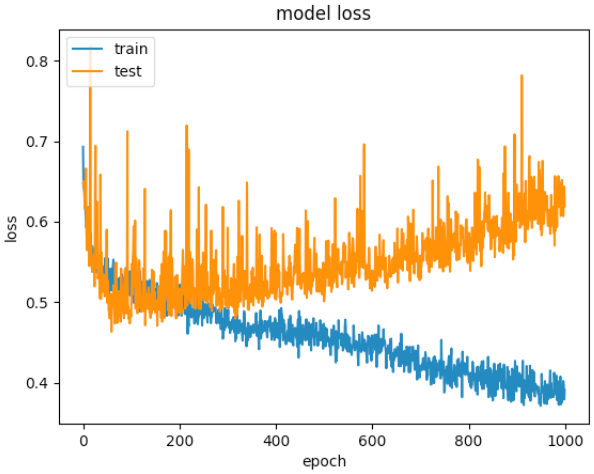


Writing the exact outputs for every layer:

Operation	Size	Padding	Stride	Output size
Conv block	[3X3X3]X64	[1 1]	1	[224X224X64]
Conv block	[3X3X64]X64	[1 1]	1	[224X224X64]
Pool 2D	[2 2]	N/A	N/A	[112X112X64]
Conv block	[3X3X64]X128	[1 1]	1	[112X112X128]
Conv block	[3X3X128]X128	[1 1]	1	[112X112X128]
Pool 2D	[2 2]	N/A	N/A	56X56X128
Conv block	[3X3X128]X256	[1 1]	1	[56X56X256]
Conv block	[3X3X256]X256	[1 1]	1	[56X56X256]
Conv block	[3X3X256]X256	[1 1]	1	[56X56X256]
Pool 2D	[2 2]	N/A	N/A	[28X28X256]
Conv block	[3X3X256]X512	[2 2]	1	[28X28X512]
Conv block	[3X3X512]X512	[2 2]	1	[28X28X512]
Conv block	[3X3X512]X512	[2 2]	1	[28X28X512]
Pool 2D	[2 2]	N/A	N/A	[14X14X512]
Conv block	[3X3X512]X512	[1 1]	1	[14X14X512]
Conv block	[3X3X512]X512	[1 1]	1	[14X14X512]
Conv block	[3X3X512]X512	[1 1]	1	[14X14X512]
Conv block	[3X3X512]X512	[1 1]	1	[14X14X512]
Pool 2D	[2 2]	N/A	N/A	[7X7X512]
Fully connected	[7·7·512]X4096	N/A	N/A	[1X1X4096]
Fully connected	[4096X4096]	N/A	N/A	[1X1X4096]
Fully connected	[4096X1000]	N/A	N/A	[1X1X1000]

C. Question 3.

Definition: Overfitting - is a situation, where network is fitted too much to the training data, and finds it difficult to generalize to create predictions for the new data.
How to spot: First of all, a researcher will notice that the accuracy of the model on the Test Dataset decreases, while the accuracy on the Training Set will still be increasing. The typical graph vizuasing error on the Test & Training set may be seen, demonstrating this exact situation:



How to avoid: There are numerous way to avoid overfitting:

1. Stop the training before the accuracy for the validation set starts increasing. If the accuracy does not satisfy, find a better dataset / improve the network / apply other changes. Training for more time will worsen the situation.

2. Use one of the following methods: regularisation, lambda factor, dropout, etc.

3. Increase the dataset size. Feed the network new examples for learning.

D. Question 4.

The learned parameters are being updated using the backpropagation algorithm. The name derives from the way that the Error value propagates backward through the network, affecting the parameters according to the contribution that those gave to the error value. In the general view, this may be seen in update function for the parameters (in this case - weights):

$$W := W - \alpha \cdot \frac{\delta E}{\delta W}$$

where:

$$\begin{aligned} W &- \text{lweight parameter} \\ \alpha &- \text{learning rate} \\ \frac{\delta E}{\delta W} &- \text{"contribution" of the parameter to the loss} \end{aligned}$$

E. Question 5.

Definition: Batch normalization - is a method which is used to normalize the layer inputs, in order to solve the problem called *internal covariate shift*.

internal covariate shift: the problem which arises in the intermediate layers during training because the distribution of the activations is constantly changing during training. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities. [1] [2]

So, actually we force the input of a specific layer to have approximately the same distribution in every training step. The batch normalization is performed in 4 steps (image taken from the original article):

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

Steps are the following:

1. Calculate the batch mean of the values x of a particular layer (that we do the normalization on) μ_β
2. Similarly, calculate the variance of those values x for this layer σ^2_β
3. Normalize the values, substracting the mean μ_β and dividing by STD (+constant) $\sqrt{\sigma^2_\beta + \epsilon}$. This will result in a new Gaussian distribution with mean of 0 and Variance of 1.
4. Scale and Shift by learnable parameters γ and β. Those parameters are being learned, and are inserted to make it possible to the distribution to be scaled and shifted if such is needed. For example, if it is of our interest to make the batch normalization an identity transform.

[1] Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift , Sergey Ioffe, Christian Szegedy

[2] Towards Data Science: Batch normalization: theory and how to use it with Tensorflow