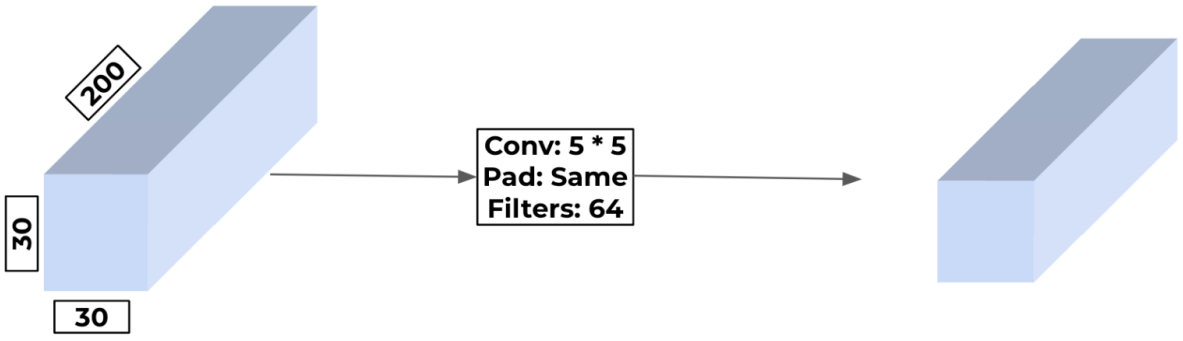
Deep Learning (Spring 2022) Homework 2
Assigned: Mar 7, 2022
Due: Mar 21, 2022

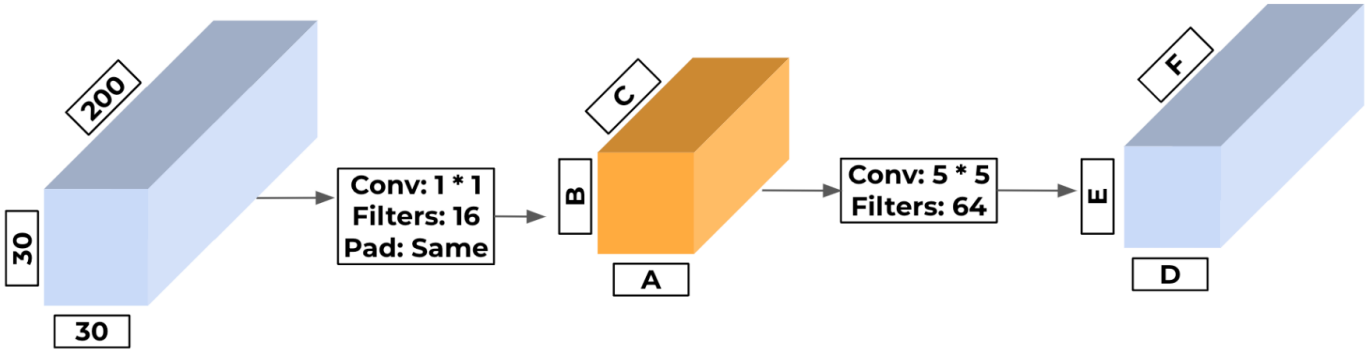
*This homework is to be done individually. Please submit your solutions to the theory questions in a single PDF document. Please submit your solutions to the coding questions in a single Python notebook (in ipynb) format. All code must use PyTorch and Python.*

1. (**IoU metric, 30 Points)** Recall the definition of **the IoU metric** for comparing bounding boxes.
    1. [15 points] Using elementary properties of sets, prove that the IoU metric between any two pairs of bounding boxes is always a non-negative real number in [0,1].
    2. [15 points] If we represent each bounding box as a function of the top-left and bottom-right coordinates (assume all coordinates are real numbers) then argue that the IoU metric is **non-differentiable** and hence cannot be directly optimized by gradient descent.

2. (**Convolutional Layer, 30 Points)** Consider the following convolutional layer in a CNN. As discussed in the class, we can use a "bottleneck" to reduce the number of multiplications needed. Thus, we implemented a bottleneck to generate an output with the same dimensions but less computational cost.
    1. [18 Points] Calculate A, B, C, E, E, F marked on the following figure.
    2. [12 Points] Calculate the computational cost of the regular conv layer and compare it with the total cost needed for the bottleneck path.




Regular Conv




Bottleneck

3. (**Object Detection, 40 Points)** In this programming exercise, we will explore the performance of three different object detection networks. We will be using Detectron2, Facebook AI's object detector library; here is the Repo. It will be helpful to go through the excellent tutorial here.

- Download the following [test image] (a picture of pedestrians in Central Park). We will run two different detectors on this image.

- First, consider the COCO Keypoint Person Detector model with a ResNet50-FPN base network, which is trained to detect human silhouettes. This can be found in the [Detectron2 Model Zoo] in the "COCO Keypoint" table. Use this model to detect as many silhouettes of people in the test image as you can. You may have to play around with the thresholds to optimize performance.

- Second, repeat the above procedure, but with the Mask R-CNN model with ResNet50-FPN backbone, available in the Model Zoo in the "COCO Instance Segmentation" table. This time, you should be able to detect both people as well as other objects in the scene. Comment on your findings.

- It appears that the balloons in the test image are not being properly detected in either model. This is because the COCO dataset used to train the above models does not contain balloons! Following the tutorial code above, start with the above pre-trained Mask R-CNN model and train a balloon detector using the (fine-tuning) balloon image dataset provided [here]. Test it on the original test image and show that you are now able to identify all the balloons.