50.021 Artificial Intelligence
Theory Homework 3

Due: every Monday, 4PM before class starts

[Q1]. Consider the following CNN that has:

1. Input of 10×10, with 30 channels.
2. A convolutional layer C with 12 filters, each of size 4×4. The convolution zero-padding is 1 and the stride is 2.
3. A max pooling layer P that is applied over each of the C's output feature maps, using 3×3 receptive fields and stride 1.

What is the total size of P's output feature map?

Solution:
$3 \times 3 \times 12 = 108$

Explanation:
The output from **each** filter on layer C has a size of 5×5 because the input has a size of 10×10, and we apply a filter of size 4×4 with stride 2. Then we feed this output per filter to P, using 3×3 receptive fields and stride 1, hence, having an output of 3×3. Therefore since we have 12 filters, the size of each channel of P's output feature map is $3 \times 3 \times 12$.

Now we want to compute the overhead of the above CNN in terms of floating point operation (FLOP). FLOP can be used to measure computer's performance. A decent processor nowadays can perform in Giga-FLOPS, that means billions of FLOP per second. Assume the inputs are all scalars (we have $10 \times 10 \times 30$ scalars as input), we have the computational cost of:

1. 1 FLOP for a single scalar multiplication $x_i \cdot x_j$
2. 1 FLOP for a single scalar addition $x_i + x_j$
3. $(n-1)$ FLOPs for a max operation over n items: $\max\{x_1, ..., x_n\}$

How many FLOPs layer C and P cost in total to do one forward pass?

Solution:
For layer C we have $12 \times (5 \times 5) \times (4 \times 4 \times 30 \times 2)$ FLOPs.

Explanation:
Per channel per filter, C gives an output of size 5×5 (discussed above). So for each of this 25 outputs, it comes from a 4×4 filter. So far this gives $(5 \times 5) \times (4 \times 4)$. The inner product of a filter with the input consists of 2 FLOPs: addition and multiplication, therefore multiplied by two: $(5 \times 5) \times (4 \times 4) \times 2$. Lastly, we need to do this per filter and per channel, and finally we should multiply by 30 and 12.

Note: Actually we have 1 less FLOP for the addition, so the precise answer is $12 \times (5 \times 5) \times (4 \times 4 \times 30 - 1 + 4 \times 4 \times 30)$*, but for simplicity, either answer is accepted.*

For layer P, we have $8 \times 9 \times 12$ FLOPs

Explanation:
The size of P's receptive fields is $3 \times 3 = 9$. Therefore, we are comparing a max over 9 items, which is equal to 8 FLOPs per receptive fields. The output of P layer per filter is $3 \times 3 = 9$,

and there are 12 filters in total coming from C. Therefore the total FLOPs for layer P is $8 \times 9 \times 12$.

[Q2]. Consider the following 3×3 filter,

$$w = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

This filter w is applied to a grayscale image shown in Figure 1. Assume that the dimension the image in Figure 1 is way larger than 3×3. We can express the image in terms of matrix M, each element is numbered between 0 to 1 (0 being completely black and 1 being completely white). We are applying convolution of the filter w to M : $(M * w)$. Answer the following questions,

1. For which part of the image will the filter return a number thats furthest possible from zero (very positive or very negative)? (ignore the arrow and the words, thats for the next question) Give a max of 2 sentences explanation.
2. Will the convolution output at the location indicated in Figure 1 be positive, negative, or zero in value? Give a max of 2 sentences explanation.

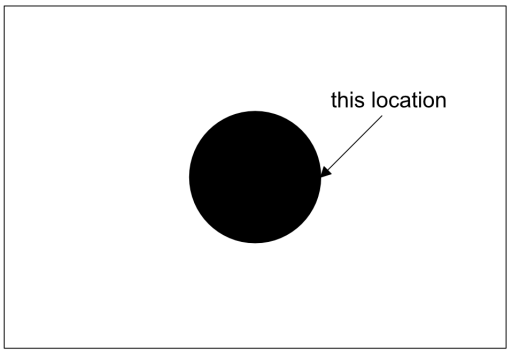

Figure 1: Figure for Q2

Solution:

1. The top or bottom part of the circle will return a value furthest from zero. Anyway, it will return a value furthest from zero on any horizontal edges. This is because the product with the first row is not canceled with the third row.
2. The marked part is a vertical edge. So it will return zero value, since the first and the third row of the filter will always cancel out.