Introduction to Machine Learning, Spring 2025
Homework 5

(Due May 5, 2025 at 11:59pm (CST))

April 21, 2025

1. Please write your solutions in English.
2. Submit your solutions to the course Gradescope.
3. If you want to submit a handwritten version, scan it clearly.
4. Late homeworks submitted within 3 days of the due date will be marked down 25% each day cumulatively Homeworks submitted more than 3 days after the due date will not be accepted unless there is a valid reason, such as a medical or family emergency.
5. You are required to follow ShanghaiTech's academic honesty policies. You are allowed to discuss problems with other students, but you must write up your solutions by yourselves. You are not allowed to copy materials from other students or from online or published resources. Violating academic honesty can result in serious penalties.

1. [15 points] [Convolutional Neural Networks]

   (a) Consider a sequential 2D convolution block: the input dimension is 4 × 64 × 64 (channel, width, height) and we use **two** continously Conv2D layer. The first layer is with 4 channels input and 8 channels output, where the kernel size is 3 × 3 (width, height). And the second layer is with 8 channels input and 16 channels output, where the kernel size is 5 × 5 (width, height). Set both the layers with stride = 1 and pad = 1. What is the output dimension after each layer? How many parameters do we have in these two layers in total? [10 points] [Hint: Do not forget the bias term.]

   (b) The convolution block is followed by a max pooling layer with 2 × 2 (width, height) filter and stride = 2. What is the output dimension of the pooling layer? How many parameters do we have in the pooling layer? [5 points]

[Hint]: You can check the pytorch document for details.

Solution

(a) Since the input image is $4 \times 64 \times 64$, so $W = 64, H = 64$.
And since stride $S = 1$, pad $P = 1$, kernel size $F = 3$,
so the output dimension is $W_{conv} = \dfrac{W + 2P - F}{S} + 1 = 64$, $H_{conv} = \dfrac{H + 2P - F}{S} + 1 = 64$.
So the dimension of the output for the first layer is $8 \times 64 \times 64$, and similarly, the dimension of the output for the second layer is $16 \times 62 \times 62$.

For the first convolution layer, the kernels have total $8 \times 4 \times 3 \times 3 = 288$ parameters. And each kernel has a bias, which is $8 \times 1 = 8$ parameter.
And for the second convolution layer, the kernels have total $16 \times 8 \times 5 \times 5 = 3200$ parameters. And each kernel has a bias, which is $16 \times 1 = 16$ parameter.
So the total number of parameters is $288 + 8 + 3200 + 16 = 3512$.

(b) Since the output dimension of the convolution layer is $16 \times 62 \times 62$.
And for the pooling layer, the filter dimension is $F' = 2$, stride $S' = 2$,
so the output dimension is $W_{pooling} = \dfrac{W_{conv} - F'}{S'} + 1 = 31$, $H_{pooling} = \dfrac{H_{conv} - F'}{S'} + 1 = 31$.
So the output dimension is $16 \times 31 \times 31$.

And since the pooling layer is a max pooling layer, so there is no parameter in this layer.

So above all, the output dimension is $16 \times 31 \times 31$, and the total number of parameters is 0.

2. [15 points] [Convolution and Cross-Correlation]

(a) Compute cross-correlation and convolution of A and B (Use B as the filter), respectively. Both the operations are with zero-padding, $P = 1$, and stride is set to be 1. [12 points]

$$A = \begin{bmatrix} 2 & 4 & -3 \\ 3 & -2 & 2 \\ -3 & 1 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 1 & -2 \\ -2 & 2 & 4 \\ 1 & 3 & -3 \end{bmatrix}$$

(b) Under what circumstances does convolution have the same result as correlation? [3 points]

[Hint: Please notice the mathmatical definition of convolution and cross-correlation.]

Solution

(a)

$$\text{Cross-Correlation Result } = \begin{bmatrix} 35 & -17 & -10 \\ -20 & 2 & 30 \\ 5 & 27 & 2 \end{bmatrix}$$
$$\text{Convolution Result } = \begin{bmatrix} -7 & 20 & 16 \\ 20 & 26 & -23 \\ -1 & -31 & 24 \end{bmatrix}$$

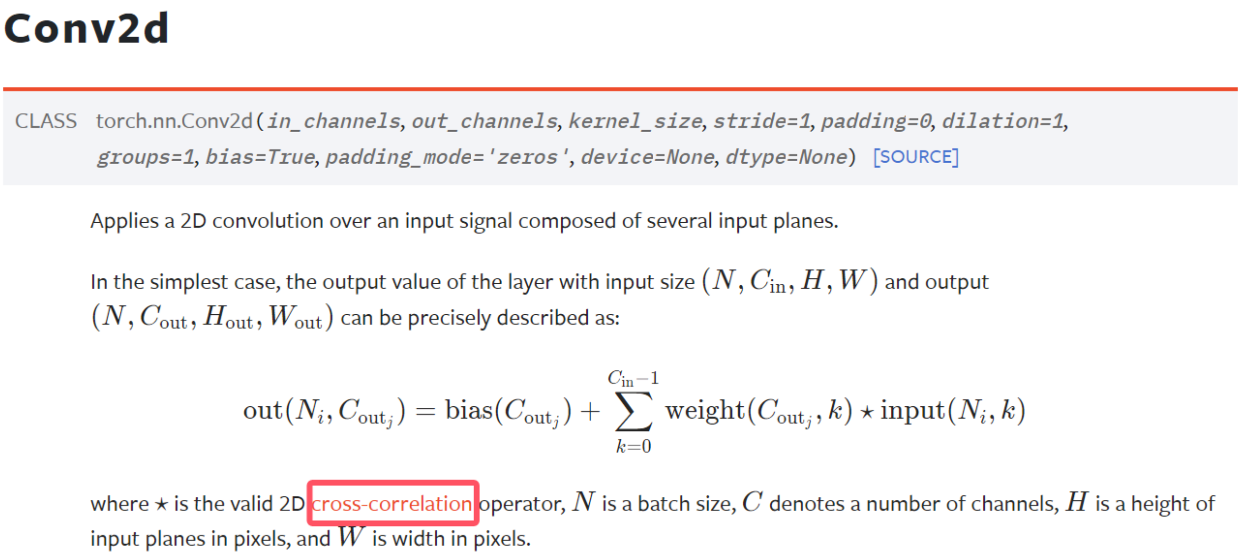
(b) Specifically, convolution and cross-correlation produce the same output when the kernel(a.k.a. filter, i.e. B) is symmetric about its midpoint and when the other input signal is unchanged.

3. [5 points] [Convolution layer implementation details]

Check the pytorch document, point out whether the convolution layer in PyTorch is using cross-correlation or convolution? And why it is defined like this? [5 points]

Solution

From the document, we can see that the convolution layer in PyTorch is using cross-correlation.



Conv2d

CLASS torch.nn.Conv2d(*in_channels*, *out_channels*, *kernel_size*, *stride=1*, *padding=0*, *dilation=1*, *groups=1*, *bias=True*, *padding_mode='zeros'*, *device=None*, *dtype=None*) [SOURCE]

Applies a 2D convolution over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C_{in}, H, W) and output $(N, C_{\text{out}}, H_{\text{out}}, W_{\text{out}})$ can be precisely described as:

$$\text{out}(N_i, C_{\text{out}_j}) = \text{bias}(C_{\text{out}_j}) + \sum_{k=0}^{C_{\text{in}}-1} \text{weight}(C_{\text{out}_j}, k) \star \text{input}(N_i, k)$$

where $\star$ is the valid 2D cross-correlation operator, N is a batch size, C denotes a number of channels, H is a height of input planes in pixels, and W is width in pixels.

The kernel in the convolution layer in PyTorch is learnable, so we don't have to worry about whether it has been flipped or not. Cross-correlation is easier to compute.