DS-GA 1008: Deep Learning, Spring 2019
Homework Assignment 2

Lekha Iyengar

February 2019

Problem 1.1 - Convolution

Part A

What is the dimensionality of the output if we forward propagate the image over the given convolution with no padding and a stride of 1?

- Padding P = 0
- Stride s = 1
- Kernel Width k = 3
- Kernel Height = 3
- Input Width I = 5
- Input Height = 5

$$Outputwidth = (\left\lfloor \frac{I + 2P - K}{S} \right\rfloor + 1)$$

$$Outputwidth = (\left\lfloor \frac{5 + 0 - 3}{1} \right\rfloor + 1)$$

$$Outputwidth = 3$$

$$OutputHeight = Outputwidth = 3$$

Output dimension is 3*3.

Part B

General formula for output width O.

- Input width I
- Kernel width K
- Padding P

$$O = (\left\lfloor \frac{I + 2P - K}{S} \right\rfloor + 1)$$

Part C

Output C. Bias term of convolution is 0.

$A =$

4	5	2	2	1
3	3	2	2	4
4	3	4	1	1
5	1	4	1	2
5	1	3	1	4

$B =$

4	3	3
5	5	5
2	4	3

$C =$

109	92	72
108	85	74
110	74	79

Part D

Gradient back-propagated from layer above is,

$$D = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Gradient (w.r.t) input image back propagated out of this layer We get this by doing a full convolution of the Kernel rotated 180 degrees with the gradient back-propagated from layer above The rotated kernel K is

$K =$

3	4	2
5	5	5
3	3	4

Full convolution of K with D gives us

$$G = \begin{bmatrix} 4 & 7 & 10 & 6 & 3 \\ 9 & 17 & 25 & 16 & 8 \\ 11 & 23 & 34 & 23 & 11 \\ 7 & 16 & 24 & 17 & 8 \\ 2 & 6 & 9 & 7 & 3 \end{bmatrix}$$

Problem 1.2 - Pooling

Part A

List torch.nn modules for 2d versions of these pooling techniques and read on they do.

1. Avg Pool2d

   This is the 2D module in torch.nn for average pooling. It applies 2D average pooling over an input signal comprised of several input planes. The parameters are kernel_size, stride, padding, ceil_mode, and count_include_pad. If the input has dimension (N, C, H_{in}, W_{in}) and the dimension of output is (N, C, H_{out}, W_{out}) then,

$$H_{out} = \left\lfloor \frac{H_{in} + 2 * padding[0] - kernel\_size[0]}{stride[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 * padding[1] - kernel\_size[1]}{stride[1]} + 1 \right\rfloor$$

   ceil_mode - When set to True, it uses ceiling instead of floor to compute H_{out} and W_{out}. The default value is False.

   count_include_pad - When set to True it includes zero padding in averaging calculations. Default value is True.

2. MaxPool2D

   This is a torch.nn module for max pooling. It applies 2D max pooling over an input signal composed of several input planes. The parameters are kernel_size, stride, padding, dilation, return_indices and ceil_mode. If the input has dimension (N, C, H_{in}, W_{in}) and the dimension of output is (N, C, H_{out}, W_{out}) then,

$$H_{out} = \left\lfloor \frac{H_{in} + 2 * padding[0] - dilation[0] * (kernel\_size[0] - 1) - 1}{stride[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 * padding[1] - dilation[1] * (kernel\_size[1] - 1) - 1}{stride[1]} + 1 \right\rfloor$$

dilation controls the spacing between kernel points. It controls the stride of elements in the window.

return_indices - If this parameter is set to True, it returns max indices along with output. It is useful for unpooling.

3. LPPool2D

This is the torch.nn module for LP Pooling, it applies 2D power averaging over an input signal of several input planes. The parameters are norm_type, kernel_size, stride, and ceil_mode. The power average function is computed as,

$$f(x) = \sqrt[p]{\sum_{x \in X} x^p}$$

If the input has dimension (N, C, H_{in}, W_{in}) and the dimension of output is (N, C, H_{out}, W_{out}) then,

$$H_{out} = \left\lfloor \frac{H_{in} + 2 * padding[0] - dilation[0] * (kernel\_size[0] - 1) - 1}{stride[0]} + 1 \right\rfloor$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2 * padding[1] - dilation[1] * (kernel\_size[1] - 1) - 1}{stride[1]} + 1 \right\rfloor$$

ceil_mode - When set to true it uses ceiling instead of floor to compute H_{out} and W_{out}.

Part B

X^k is k^{th} input feature. $X^k \in R^{H_{in} * W_{in}}$, where H_{in} and w_{in} are input height and width respectively. Y^k is k^{th} output feature $Y^k \in R^{H_{out} * W_{out}}$ where H_{out} and W_{out} are output height and width respectively. S^k_{ij} is list of indices of elements in subregion of X^k used for generating Y^k_{ij} the $(i, j)^{th}$ entry of Y^k.

1. Max pool

$$Y^k_{ij} = max(X^k[s]) \; \forall \; s \in S^k_{ij}$$

2. Avg pool

$$Y_{ij}^{k} = \frac{\sum_{s \in Sij^k} X^k[s]}{\mid S_{i,j}^{k} \mid}$$

3. LP pool

$$Y_{ij}^{k} = \sqrt[p]{\sum_{s \in Sij^k} (X^k[s])^p}$$

Part C

$$C = \begin{array}{|c|c|c|} \hline 109 & 92 & 72 \\ \hline 108 & 85 & 74 \\ \hline 110 & 74 & 79 \\ \hline \end{array}$$

Applying max pooling with kernel size 2 and stride 1 on C we get

109	92
110	85

Part D

Max pooling and Average pooling can be expressed in terms of LP pooling.

$$f(x) = \sqrt[p]{\sum_{x \in X} x^p}$$

At $p = \infty$, one gets max pooling.
At $p = 1$, one gets sum pooling, which is proportional to average pooling.

Links

- Read only link to Overleaf Project