DS-GA 1008: Deep Learning - Homework 2 (Spring 2020)

Alfred Rajakumar

March 7, 2020

1 Problem - 1 Fundamentals:

1.1 Convolution:

Here,

$$\mathbf{A}_{5\times5} = \begin{bmatrix} 4 & 5 & 2 & 2 & 1 \\ 3 & 3 & 2 & 2 & 4 \\ 4 & 3 & 4 & 1 & 1 \\ 5 & 1 & 4 & 1 & 2 \\ 5 & 1 & 3 & 1 & 4 \end{bmatrix} \text{ and } \mathbf{B}_{3\times3} = \begin{bmatrix} 4 & 3 & 3 \\ 5 & 5 & 5 \\ 2 & 4 & 3 \end{bmatrix}, \tag{1}$$

a. The dimension of the output on forward propagation is: 3×3

b. General formula for output width is: $O = \left\lfloor \frac{I+2P-(K-1)-1}{S} + 1 \right\rfloor$

c. On forward propagating, we get,

$$\mathbf{C}_{3\times3} = \begin{bmatrix} 109 & 92 & 72 \\ 108 & 85 & 74 \\ 110 & 74 & 79 \end{bmatrix} \tag{2}$$

d. We have $\frac{\partial E}{\partial C_{ij}} = 1$, and by applying chain rule, we could also get,

$$\frac{\partial E}{\partial A_{i,j}} = \begin{cases} \frac{\partial E}{\partial C_{k,l}} B_{i-k+1,j-l+1}, & \text{when } 1 \leq i-k+1 \leq 3 \text{ and } 1 \leq j-l+1 \leq 3 \\ 0, & \text{when } (i-k+1 < 1 \text{ or } i-k+1 > 3) \text{ and } (j-l+1 < 1 \text{ or } j-l+1 > 3) \end{cases}$$

$$\implies \frac{\partial E}{\partial A_{i,j}} = \begin{bmatrix} 4 & 7 & 10 & 6 & 3 \\ 9 & 17 & 25 & 16 & 8 \\ 11 & 23 & 34 & 23 & 11 \\ 7 & 16 & 24 & 17 & 8 \\ 2 & 6 & 9 & 7 & 3 \end{bmatrix} \tag{3}$$

1.2 Pooling:

a. The *torch.nn* modules for 2D pooling are:

(a) ***MaxPool2d***: Applies a 2D max pooling over an input signal composed of several input planes

(b) ***AvgPool2d***: Applies a 2D average pooling over an input signal composed of several input planes

(c) ***FractionalMaxPool2d***: Applies a 2D fractional max pooling over an input signal composed of several input planes

(d) ***LPPool2d***: Applies a 2D power-averaging over an input signal composed of several input planes. On each window, the function computed is:

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

(e) ***AdaptiveMaxPool2d***: Applies a 2D adaptive max pooling over an input signal composed of several input planes

(f) ***AdaptiveAvgPool2d***: Applies a 2D adaptive average pooling over an input signal composed of several input planes

b. We know that $X^k \in \mathbb{R}^{H_{in} \times W_{in}}$ and $Y^k \in \mathbb{R}^{H_{out} \times W_{out}}$. So,

(a) Max Pooling:

$$Y^k_{i,j,max} = \max_{(s,t) \in S^k_{i,j}} X^k_{s,t}$$

(b) Average Pooling:

$$Y^k_{i,j,avg} = \frac{1}{L} \sum_{(s,t) \in S^k_{i,j}} X^k_{s,t}$$

where $L = \mathit{length}(S^k_{i,j})$

(c) LP-Pooling:

$$Y^k_{i,j,p,lpp} = \sqrt[p]{\sum_{(s,t) \in S^k_{i,j}} \left(X^k_{s,t}\right)^p}$$

c. On applying max-pooling on $\mathbf{C}$, we get,

$$MaxPool2d\big(\mathbf{C}_{3\times3}\big) = \begin{bmatrix} 109 & 92 \\ 110 & 85 \end{bmatrix} \tag{4}$$

d. (a) When $p = 1$, we have LP-Pooling = (L x Average Pooling) as shown below:

$$Y^k_{i,j,1,lpp} = \left(\sqrt[p]{\sum_{(s,t) \in S^k_{i,j}} \left(X^k_{s,t}\right)^p} \right)_{p=1} = \sum_{(s,t) \in S^k_{i,j}} X^k_{s,t} = LY^k_{i,j,avg}$$

(b) When $p \to \infty$, we have LP-Pooling = Max Pooling as shown below:

$$\lim_{p \longrightarrow \infty} Y^k_{i,j,p,lpp} = \lim_{p \longrightarrow \infty} \left(\sqrt[p]{\sum_{(s,t) \in S^k_{i,j}} \left(X^k_{s,t}\right)^p} \right) = Y^k_{i,j,max}$$