Homework 4

Yuanteng Chen 3039725444

2. Feature Dimension of Convolutional Neural network

(a)

(1) $\begin{cases} \text{weights: } F \cdot C \cdot k^2 \\ \text{bias: } F \end{cases}$

(2). $W_{out} = (W - K + 2P)/S + 1$

$H_{out} = (H - k + 2p)/S + 1$

$C_{out} = F$

(b). $W_{out} = (W_{in} - k)/s + 1$

$H_{out} = (H_{in} - k)/S + 1$

$C_{out} = C_{in}$

(c) receptive:

$RF_{i+1} = S_i (RF_i - 1) + K_i$

where RF_i means the receptive field of the ith layer and $S_i \rightarrow$ stride, $K_i \rightarrow$ kernel size

as stride step size $= 1$.

$\therefore$ the receptive field size of last output is $L \cdot K - (L-1) = L(K-1) + 1$

(d) $RF_{i+1} = S_i(RF_i - 1) + k_i$

kernel size $= 2$ and stride step size $= 2$

$\therefore RF_{i+1} = 2 \cdot (RF_i - 1) + 2$

$= 2 RF_i$

$\therefore$ The receptive field size increases by 2.

as the output feature resolution decreases, we reduce the amount of computation, so the number of matrix multiply operations decreases.

(e).

Layer	parameters.	dimension
Input	0	$28 \times 28 \times 1$
Conv3-10	$10 + 3 \times 3 \times 1 \times 10 = 100$	$28 \times 28 \times 10$ $(28 + 2 \times 1 - 3)/1 + 1 = 28$
pool-2	0	$14 \times 14 \times 10$
Conv3-10	$10 + 3 \times 3 \times 10 \times 10 = 910$	$14 \times 14 \times 10$
Pool-2	0	$7 \times 7 \times 10$
Flatten	0	490
FC-3	$490 \times 3 + 3 = 1473$	3

(f)

Conv2-3 → Relu → Conv2-3 → Relu → GAP → FC-3

$f(x_3) = f(x_2) = [0, 0.8, 0]^T$

$f(x_4) = f(x_1) = [0.8, 0, 0]^T$

as CNN is invariant of circular shifs.

3. Convolutional networks.

(a).

① Convolutional layer utilize weights sharing which reduces the number of parameters compared to fully connected layers.

② Convolution layers are invariant to circular shift which means they can detect features regardless of their exact positions in an image.

(b). $[1, 4, 0, -2, 3] \rightarrow [-2, 2, 11]$

assume filter $= [a, b, c]$

$$\therefore \begin{cases} a + 4b = -2 \\ 4a - 2c = 2 \\ -2b + 3c = 11 \end{cases} \Rightarrow \begin{cases} a = 2 \\ b = -1 \\ c = 3 \end{cases}$$

$\therefore$ filter $= [2, -1, 3]$

(c) the input size $= 2 \times 2$ $\begin{bmatrix} -1 & 2 \\ 3 & 1 \end{bmatrix}$

and pad $= 0$, stride $= 1$. kernel size $= 2 \times 2$

$\therefore$ the output size $= 3 \times 3$

input $\begin{bmatrix} -1 & 2 \\ 3 & 1 \end{bmatrix}$ filter $\begin{bmatrix} +1 & -1 \\ 0 & +1 \end{bmatrix}$

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

$$\begin{bmatrix} 3 & -3 \\ 0 & 3 \end{bmatrix} \quad \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} -1 & 1+2 & -2 \\ 0+3 & \begin{matrix} -1+0 \\ -3+1 \end{matrix} & 2-1 \\ 0 & 3+0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 3 & -2 \\ 3 & -5 & 1 \\ 0 & 3 & 1 \end{bmatrix}$$

4. Convolutional networks and Dilated convolutions:

(a) $[B, C, H, W] = [10, 3, 32, 32]$.

i. 3×3; strid $=1$, padding $=1$

Sol: $H' = W' = (32 + 1\times 2 - 3)/1 + 1 = 32$

$\therefore$ output $= [10, 64, 32, 32]$

ii: 4×4. strid $=2$. padding $=0$

Sol: $H' = W' = (32 + 0 - 4)/2 + 1 = 15$

$\therefore$ output $= [10, 64, 15, 15]$

(b). detects vertical edges:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

(c) 3×3 filter to blur an image:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

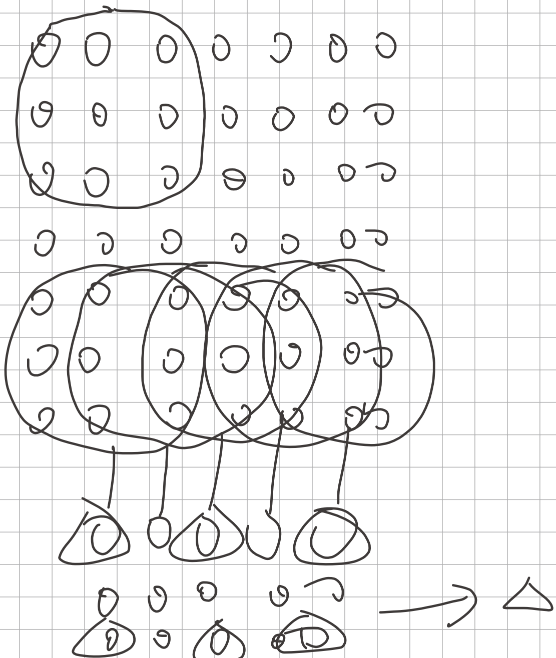
d. (i) $M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ $\quad k = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

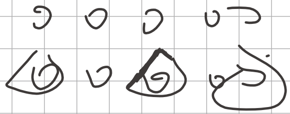
$$M' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 4 & 5 & 6 & 0 \\ 0 & 7 & 8 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\text{output} = \begin{bmatrix} 5, & 10, & 5 \\ 10, & 20, & 10 \\ 5, & 10, & 5 \end{bmatrix}$$

(ii) I assume the stride size of both layers are 1:

sol: the receptive field of DilatedConv2 is 7





5. weights and Gradients in a CNN

(a) Derive the gradient to the weight matrix: dw

Sol: $y_{i,j} = \sum_{h=1}^{k} \sum_{l=1}^{k} X_{i+h-1, j+l-1} W_{h,j}$

$$\frac{\partial L}{\partial W_{h,j}} = \sum_{i=1}^{m} \sum_{j=1}^{m} \frac{\partial L}{\partial y_{i,j}} \cdot \frac{\partial y_{i,j}}{\partial W_{h,j}}$$

$$\frac{\partial y_{i,j}}{\partial W_{h,j}} = X_{i+h-1, j+l-1}, \quad \frac{\partial L}{\partial y_{i,j}} = dy_{i,j}$$

$$\therefore \frac{\partial L}{\partial W_{h,j}} = \sum_{i=1}^{m} \sum_{j=1}^{m} dy_{i,j} \cdot X_{i+h-1, j+l-1}$$

$$\therefore dw = X \cdot dY$$

after 1 step SGD (assum learning rate $= \lambda$)

$$\therefore W_{t+1} = W_t - \lambda dw$$
$$= W_t - \lambda X \cdot dY$$

(b) $E[X_{i,j}] = 0$, $Var(X_{i,j}) = \sigma_x^2$

$E(dy_{i,j}) = 0$ $\quad Var(dy_{i,j}) = \sigma_g^2$

Sol: $\frac{\partial L}{\partial W_{h,j}} = \sum_{i=1}^{m} \sum_{j=1}^{m} dy_{i,j} \cdot X_{i+h-1, j+l-1}$

$\because X_{i,j}$ and $dy_{i,j}$ are independent

$$E\left(\frac{\partial L}{\partial w_{h,j}}\right) = \sum_{i=1}^{m}\sum_{j=1}^{m} E(dy_{i,j} \cdot X_{i+h-1, j+l-1})$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m} E(dy_{i,j}) \cdot E(X_{i+h-1, j+l-1})$$

又 $\because E(X_{i,j}) = E(dy_{i,j}) = 0$

$\therefore E\left(\frac{\partial L}{\partial w_{h,j}}\right) = 0$

$$Var\left(\frac{\partial L}{\partial w_{h,j}}\right) = E\left(\left(\frac{\partial L}{\partial w_{h,j}}\right)^2\right) - \left(E\left(\frac{\partial L}{\partial w_{h,j}}\right)\right)^2$$

$$= E\left(\left(\frac{\partial L}{\partial w_{h,j}}\right)^2\right)$$

$$= E\left(\sum_{i=1}^{m}\sum_{j=1}^{m} X^2_{i+h-1, j+l-1} \cdot dy^2_{h,l}\right)$$

又 $\because X_{i,j}$ and $dy_{i,j}$ are independent

$$\therefore = \sum_{i=1}^{m}\sum_{j=1}^{m} E(X^2_{i+h-1, j+l-1}) \cdot E(d^2 y_{h,l})$$

$\because Var(X_{i,j}) = E(X^2_{i,j}) - (E(X_{i,j}))^2 = E(X^2_{i,j}) = \sigma_x^2$

$Var(dy_{i,j}) = E(dy^2_{i,j}) - (E(dy_{i,j}))^2 = E(dy^2_{i,j}) = \sigma_g^2$

$\therefore = m^2 \sigma_x^2 \cdot \sigma_g^2$

$\because m = (n+0-k)/1+1 = (n-k)+1$

standard deviation ~~of~~ of $\frac{\partial L}{\partial w_{hj}}$

$$= \sqrt{Var\left(\frac{\partial L}{\partial w_{h,j}}\right)} = m\sigma_x\sigma_g = (n-k+1)\sigma_x\sigma_g$$

∴ the growth rate of the standard deviation of the gradient on $dW_{h,i}$ with respect to the length and width of the image n.

(c).

Sol: $y_{1,1} = X_{1,1} = \max(X_{1,1}, X_{1,2}, X_{2,1}, X_{2,2})$

$$\therefore \frac{dy_{1,1}}{dX_{1,1}} = 1 \qquad \frac{dy_{1,1}}{dX_{1,2}} = \frac{dy_{1,1}}{dX_{2,1}} = \frac{dy_{1,1}}{d_{2,2}} = 0$$

in average pooling: $y_{1,1} = \frac{1}{4}(X_{1,1}+X_{1,2}+X_{2,1}+X_{2,2})$

$$\therefore \frac{dy_{1,1}}{dX_{1,1}} = \frac{dy_{1,1}}{dX_{1,2}} = \frac{dy_{1,1}}{dX_{2,1}} = \frac{dy_{1,1}}{dX_{2,2}} = \frac{1}{4}$$

assume $i' = i/2$, $j' = j/2$

$$\therefore \frac{\partial y_{i',j'}}{\partial X_{i,j}} = \begin{cases} 1, & X_{i,j} = \max(X_{i,j}, X_{i+1,j}, X_{i,j+1}, X_{i+1,j+1}) \\ 0, & X_{i,j} \neq \max(\cdots) \end{cases}$$

$$\therefore dx_{i,j} = \sum_{y_{i'j'}} dy_{i'j'} \frac{\partial y_{i'j'}}{\partial x_{i,j}} \quad \text{(max-pooling)},$$

$$dx_{i,j} = \frac{1}{4} y_{i'j'} \quad \text{(average-pooling)}$$

(d). ① there is no learnable parameters in Max-pooling or average-pooling. So they reduce the complexity of computation by decreasing the feature size.

② Without max-pooling or average pooling, CNN will not be invariant to circular shift as they increase the size of receptive field.

8. (a) CSDN, gpt

(b). None

(c) 10 hours