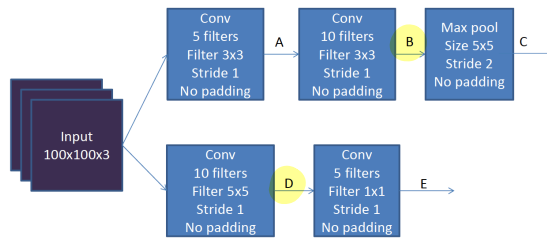


T1. Given the following CNN architecture specifications, determine the size of all the intermediate feature maps.



$$A : 5 \times 98 \times 98$$

$$B : 10 \times 96 \times 96$$

$$C : 10 \times 46 \times 46$$

$$D : 10 \times 96 \times 96$$

$$E : 5 \times 96 \times 96$$

OT1. Note how at point B and D we have feature maps of the same size. If you think carefully, having 3x3 followed by 3x3 will cover a region of 5x5 in the input. This is why in Inception v2, the 5x5 convolution is replaced by two 3x3 convolutions. Compute the amount of multiplies to compute A and B. Compare it to the amount of multiplies to compute D. Compute the amount of parameters in the path to A and B. Compare it to the amount of parameters in path to D.

The amount of multiplies to compute A

$$\left. \begin{array}{l} \text{compute B} \\ \text{compute C} \end{array} \right\} \oplus = 585 \times 98^2$$

compute B

$$96 \times 96 \times 5 \times 3 \times 3 \times 10$$

compute C

$$96 \times 96 \times 5 \times 5 \times 5 \times 10 = 750 \times 98^2$$

\therefore # multiply A-B < # multiply C.

The amount of parameters to compute A

$$\left. \begin{array}{l} \text{compute B} \\ \text{compute C} \end{array} \right\} = 585$$

compute B

$$5 \times 3 \times 3 \times 10$$

compute C

$$3 \times 5 \times 5 \times 10 = 750$$

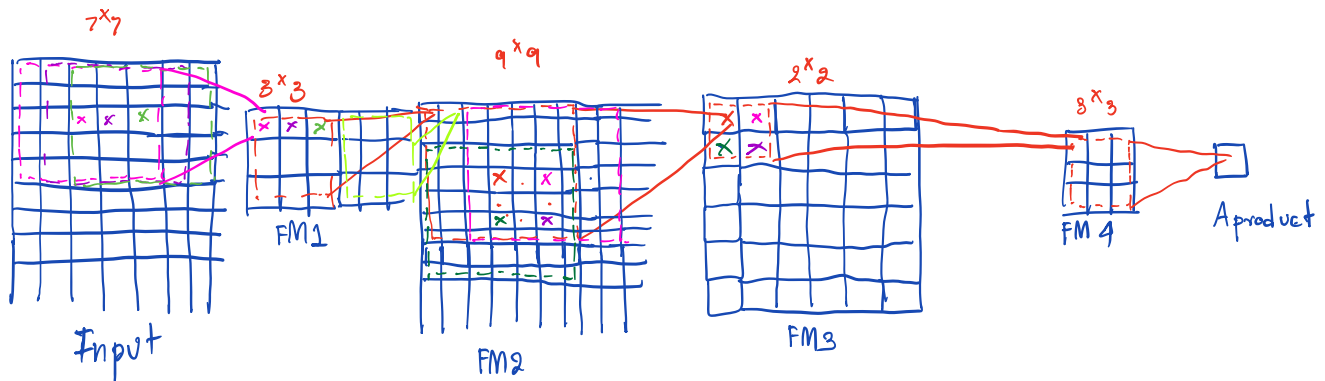
\therefore # parameter A-B < # parameter C.

T2. Consider a CNN with the following layers:

1. conv2d layer with 5x5, stride 1
2. maxpooling of size 3x3 (stride is usually the same as the size for pooling layers)
3. conv2d layer with 7x7, stride 2
4. max pooling of size 2x2
5. conv2d layer with 3x3, stride 1

What is the receptive field of this network?

image FM : Feature map



1 pixel product → 3x3 pixels FM4

3x3 pixels FM4 → (3x2) x (3x2) FM3

Overlap (3x2) x (3x2) pixels FM3 → 17x17 pixels FM2 (3+2+2+2+2+2+2+3) x

17x17 pixels FM2 → (17x3) x (17x3) pixels FM1

Overlap (17x3) x (17x3) pixels FM1 → (17x3+2+2) x (17x3+2+2) pixels Input

နိဂုံးချုပ်ချက်

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$

n_{in} : number of input features
 n_{out} : number of output features
 k : convolution kernel size
 p : convolution padding size
 s : convolution stride size

→ 55x55 pixels Input.

$$s(n_0 - 1) = n_{in} + 2p - k$$

$$s(n_0 - 1) + k - 2p = n_{in}$$

$$1(1 - 1) + 3 - 2(0) = n_{in}$$

$$3 = n_{in}$$

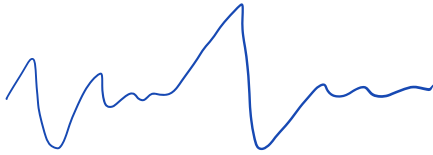
$$2(3 - 1) + 2 - 2(0) = n_{in} = 6$$

$$2(6 - 1) + 7 - 2(0) = n_{in} = 17$$

$$3(17 - 1) + 3 = n_{in} = 51$$

$$1(51 - 1) + 5 = n_{in} = 55$$

T3. Can we use CNN on a time-related input (stride over time steps)?
 Discuss the different inductive biases imposed by the model when you use a single CNN layer on a time-related input vs. when you use a single GRU layer on a time-related input. Come up with an example when one might be better than the other.



เราสามารถนำ CNN กับงานประเภท Time related ได้เนื่องจาก Convolution ภายใน CNN สามารถทำการ Convolute กับ Wave หนึ่งได้ โดยใช้ Convolution มา Convolute กับข้อมูลที่เรียงกันตามเวลา แล้วเอาหน้า Feature ที่ได้มาหันทำการหาค่าบางอย่าง ซึ่งขนาด Filter ที่นำมาจะต้องมีน้อยกว่าขนาดข้อมูล

ซึ่ง CNN จะเหมือนกับการดู Pattern ของข้อมูลนั้น ดูกราฟหรือการเปลี่ยนแปลงหนึ่งภายในข้อมูลย้อนหลัง ที่ส่งผลให้การหาค่าบางอย่างออกมาได้จุดประสงค์ของข้อมูล

แต่ถ้าเป็น GRU จะเป็นการเรียนรู้จาก Input ที่ให้เข้าไปแล้วนำผลของการหาค่ามาเป็น Input ให้กับการเรียนรู้ครั้งต่อไปต่อไปเรื่อย ๆ

โดยตัวอย่างการนำไปใช้งาน ยกตัวอย่างเช่น ถ้าข้อมูลส่งค่ามาเป็นราคาหุ้น และต้องการหาค่าราคา CNN อาจจะใช้ในการลักษณะหรือ Pattern ที่เกิดขึ้นก่อน แล้วส่งผลเป็นราคา แล้วอาจพิจารณาตามการเปลี่ยนแปลงของเวลา เหมือนกับการหาค่าราคาหุ้นโดยดูลักษณะ Candlestick แต่ถ้าเป็น GRU จะดูข้อมูลหรือ Input แล้วทำการหาค่าออกมาได้ สุดท้ายจะมีการตัดสินใจการเลือกผลในอดีต เพื่อผลที่จะทำการหาค่าได้ เหมือนกับการดู EMA ซึ่งทั้งสองวิธีเป็นวิธีที่คนเทรดใช้ในการหาค่าราคาหุ้นเหมือนกัน