Non-Local Neural Networks

발표자 : 백인성

2019.07.24.

Data Mining
Quality Analytics
hcai
Human-Centered Artificial Intelligence Center

목차

1. Introduction

Introduction

- ❖ 오늘날 이미지 분석에 딥러닝 모델을 적용하여 좋은 결과를 얻고 있음
- ❖ 이미지 분석 뿐 아니라 영상 분석에서도 좋은 결과를 얻고 있음

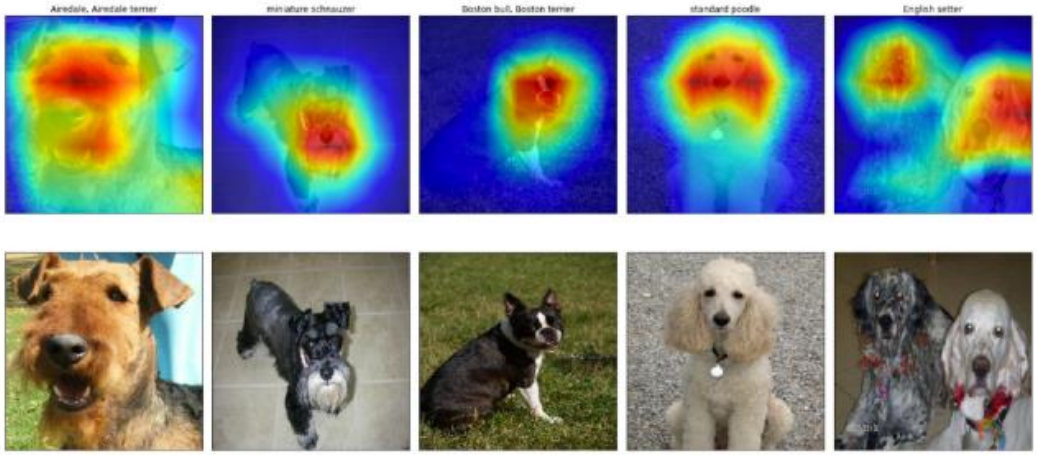
<아마존고 자동 재고 파악>
→ 선반 내 물건 인식

<자율 주행 차량>
→ 도로 위 객체 탐지

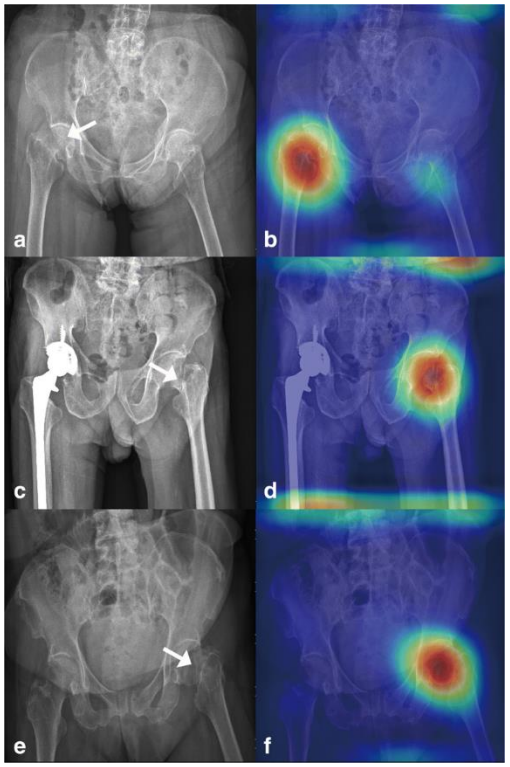
출처: https://www.youtube.com/watch?v=6NN8kXh-Tpk
HMG Journal, https://news.hmgjournal.com/Tech/deep-learning-carfuture

Introduction

❖ 이미지 분석 후 예측 결과를 해석하는 분야도 많이 발전 되고 있음



<분류 모델 원인 해석>
→ 예측 모델 결과에 대한 신뢰성

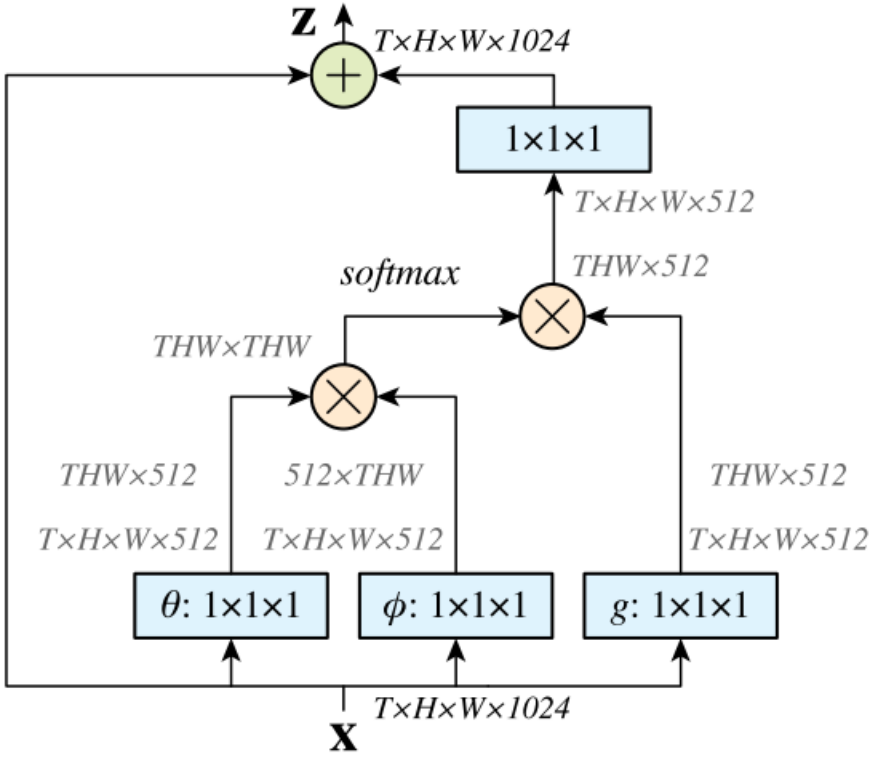


<고관절 골절 탐지>
→ 병 원인 진단

출처: https://alexisbcook.github.io/2017/global-average-pooling-layers-for-object-localization/
Cheng, Chi-Tung, et al. "Application of a deep learning algorithm for detection and visualization of hip fractures on plain pelvic radiographs." European radiology (2019): 1-9.

Introduction

- Non-local neural networks preview
  - **CNN 모델의 성능을 더 높일** 수 있는 방안을 고려
  - Self-Attention Mechanism을 활용하여 **해석이 가능**하게 함



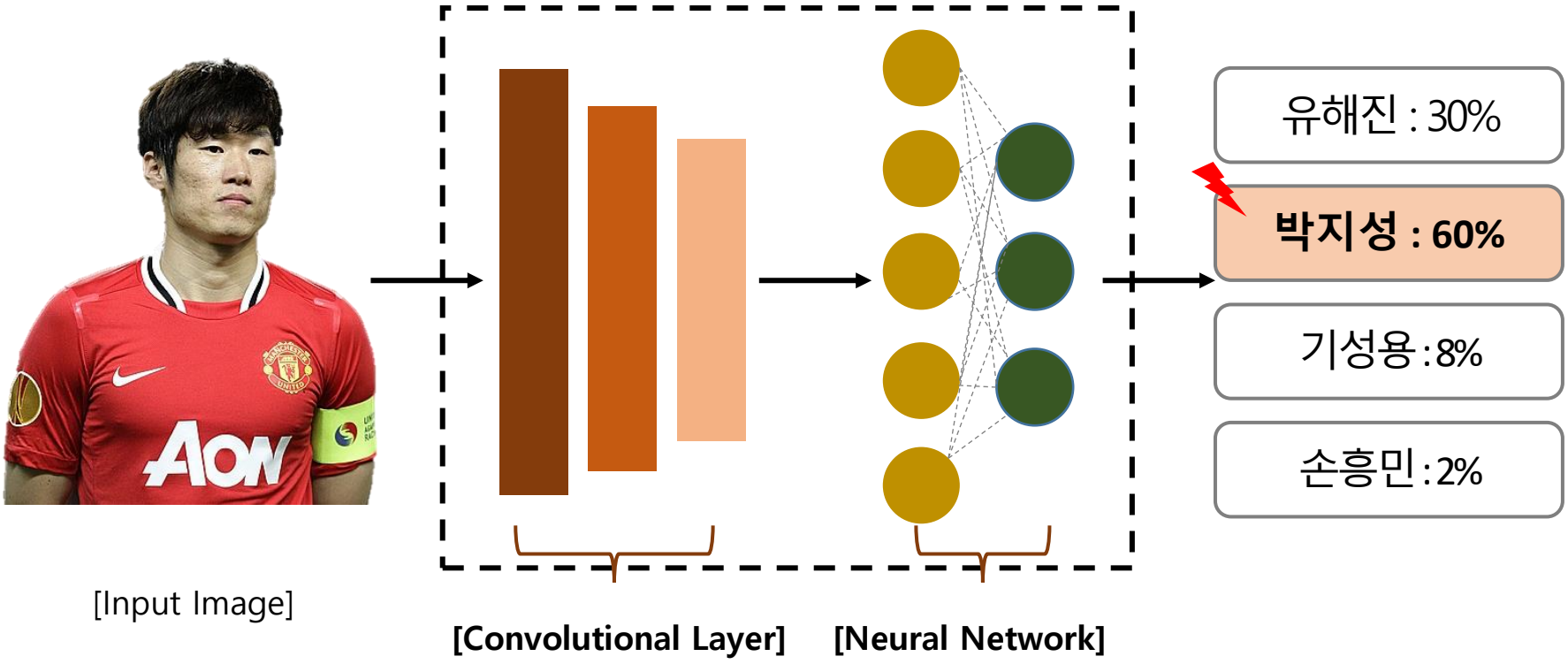
출처: Wang, Xiaolong, et al. "Non-local neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

2. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN)

❖ Convolutional Neural Network 기본 구조

- ➢ Neural Network에 Convolution Layer를 사용한 방법론
- ➢ Object Detection, Classification 등 Visual Task에서 좋은 Performance 보임

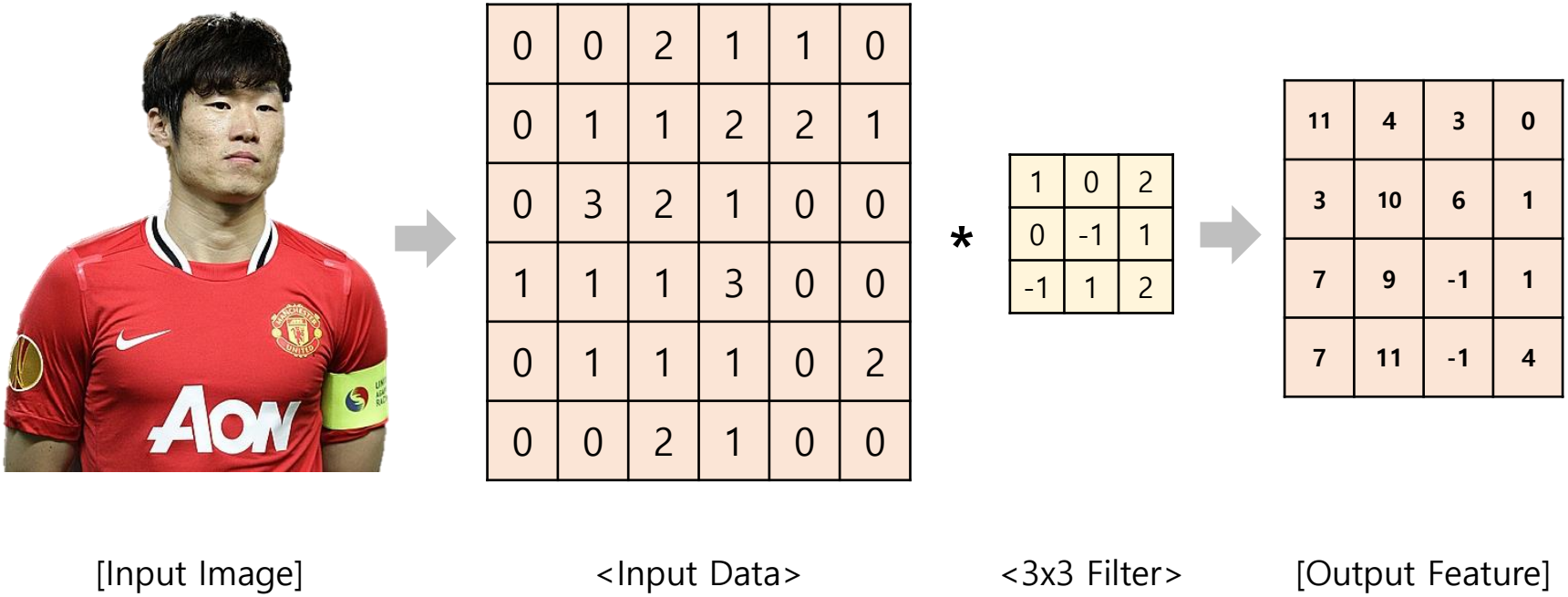


출처: http://www.sisaweek.com/news/articleView.html?idxno=28851

Convolutional Neural Network (CNN)

❖ 3x3 Filter를 사용한 CNN

- ➢ CNN 구조에서는 입력 값에서 중요한 정보를 요약하기 위해 필터 사용
- ➢ Convolutional Filter를 사용하면 입력 크기를 줄이며 중요 정보 요약 가능



[Input Image] <Input Data> <3x3 Filter> [Output Feature]

Convolutional Neural Network (CNN)

❖ 3x3 Filter를 사용한 CNN

➢ 하지만 3x3 Filter를 사용하면 필터 내 parameter가 학습을 진행할 때, 입력 데이터 내 모든 정보를 고려할 수 없다는 한계점이 존재함

[사례1]

0	0	2	1	1	0
0	1	1	2	2	1
0	3	2	1	0	0
1	1	1	3	0	0
0	1	1	1	0	2
0	0	2	1	0	0

<Input Data>

*

1	0	2
0	-1	1
-1	1	2

<3x3 Filter>

11	**4**	**3**	**0**
3	**10**	**6**	**1**
7	**9**	**-1**	**1**
7	**11**	**-1**	**4**

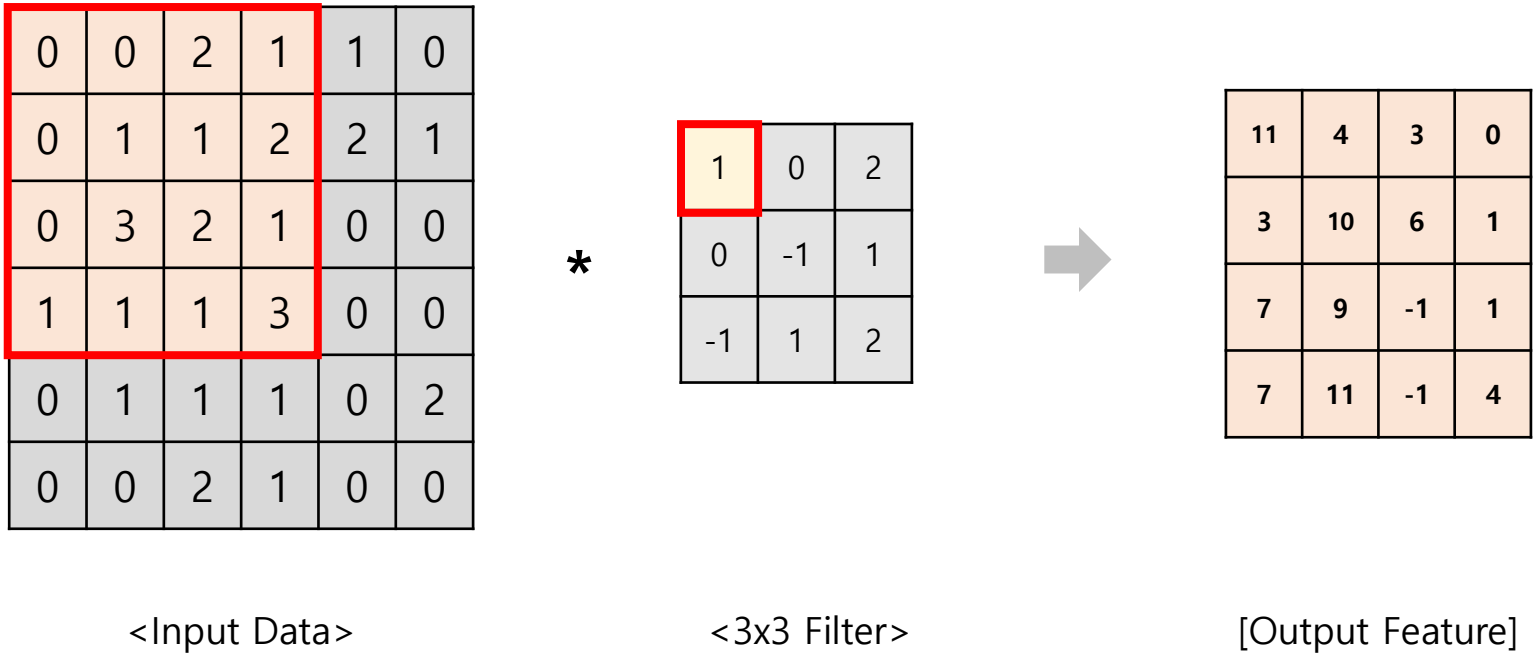
[Output Feature]

Convolutional Neural Network (CNN)

❖ 3x3 Filter를 사용한 CNN

➢ 하지만 3x3 Filter를 사용하면 필터 내 parameter가 학습을 진행할 때, 입력 데이터 내 모든 정보를 고려할 수 없다는 한계점이 존재함

[사례1]



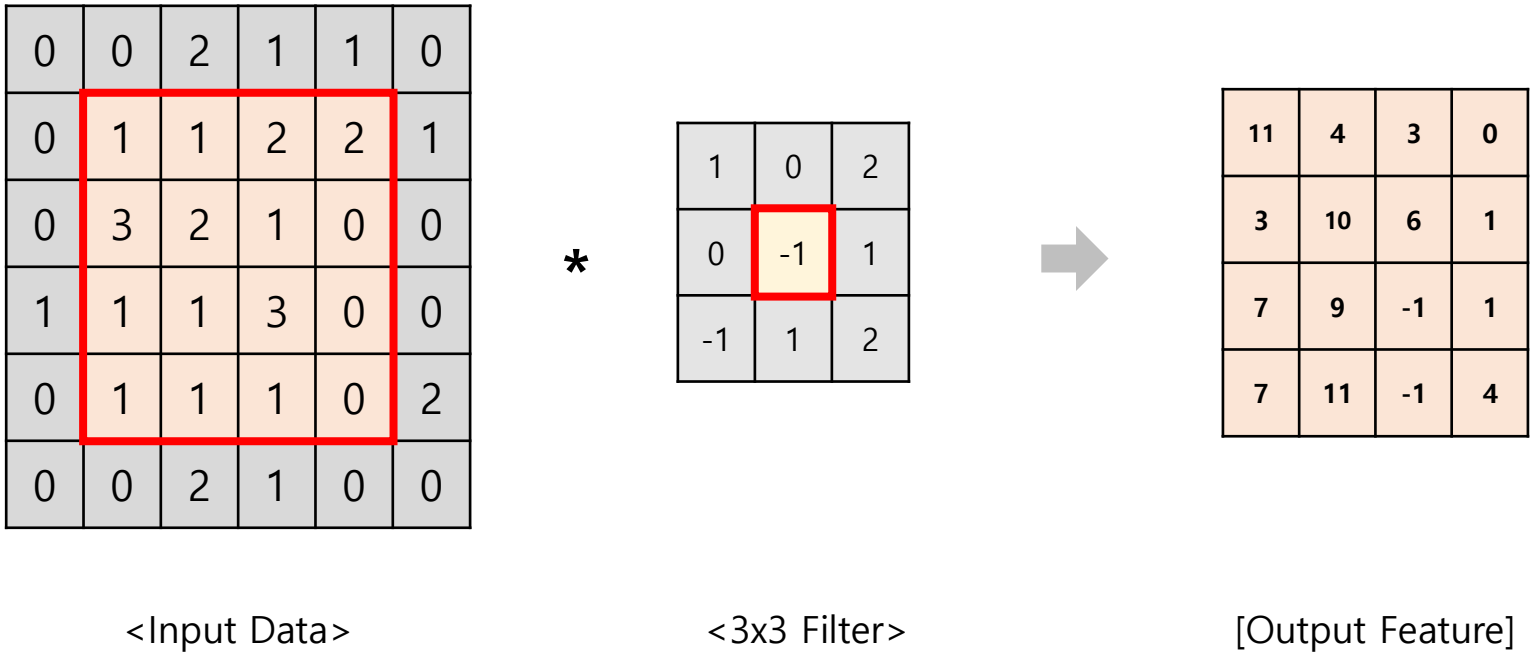
<Input Data> <3x3 Filter> [Output Feature]

Convolutional Neural Network (CNN)

❖ 3x3 Filter를 사용한 CNN

➢ 하지만 3x3 Filter를 사용하면 필터 내 parameter가 학습을 진행할 때, 입력 데이터 내 모든 정보를 고려할 수 없다는 한계점이 존재함

[사례2]



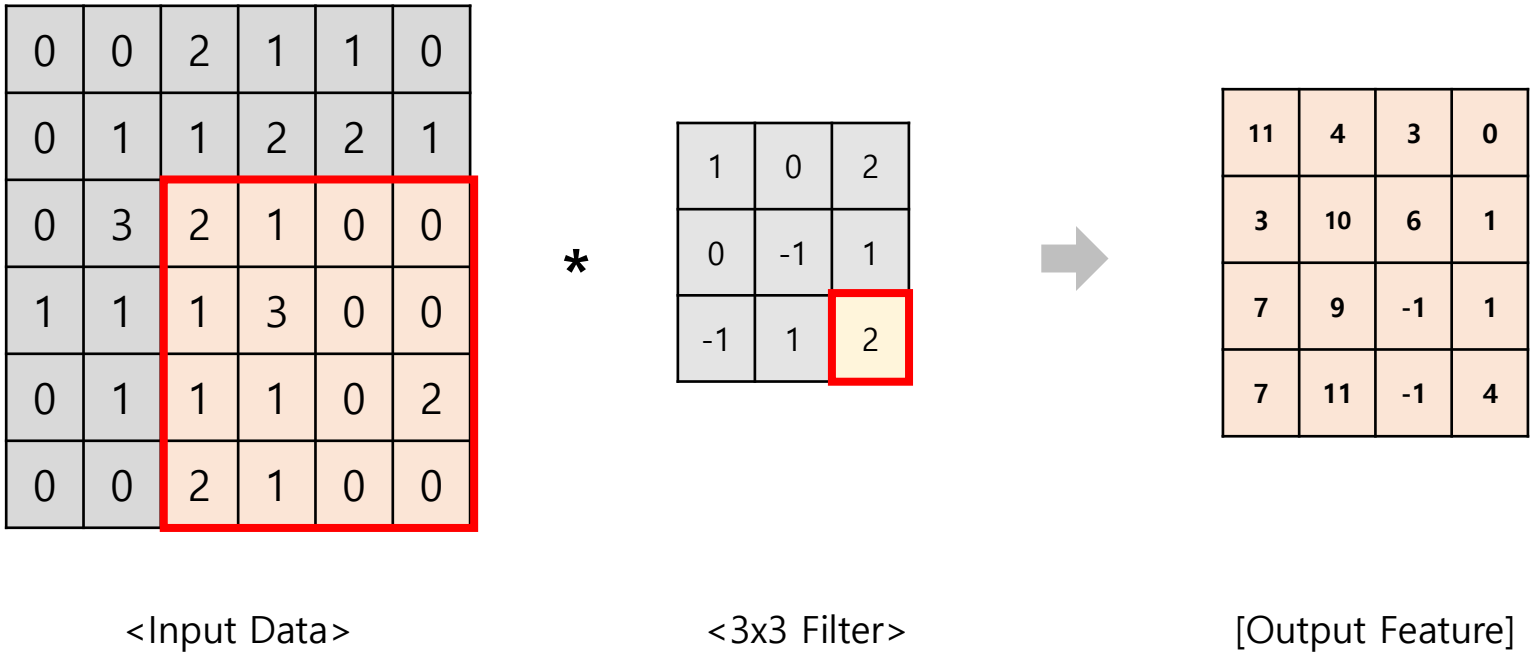
<Input Data> <3x3 Filter> [Output Feature]

Convolutional Neural Network (CNN)

❖ 3x3 Filter를 사용한 CNN

➢ 하지만 3x3 Filter를 사용하면 필터 내 parameter가 학습을 진행할 때, 입력 데이터 내 모든 정보를 고려할 수 없다는 한계점이 존재함

[사례3]

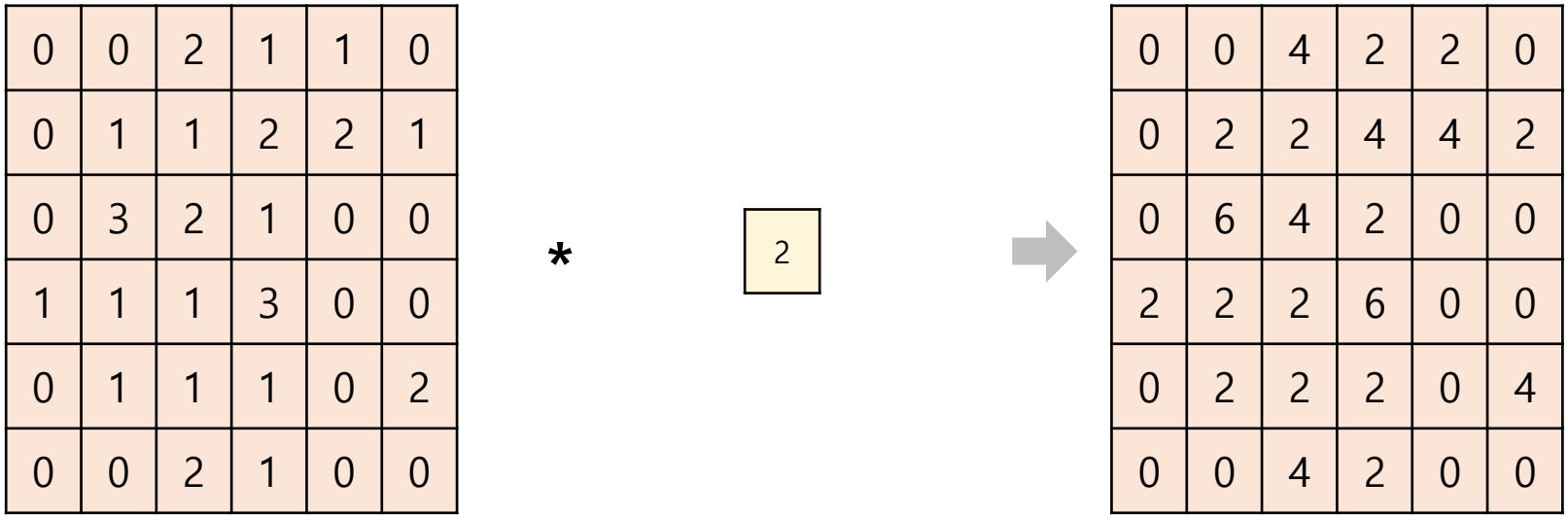


<Input Data> <3x3 Filter> [Output Feature]

Convolutional Neural Network (CNN)

❖ 1x1 Convolutional Filter를 사용했을 때 장점①

➢ 1x1 Convolutional Filter를 사용하면 입력 데이터 내 모든 정보를 고려하며 필터 내 parameter가 학습을 진행할 수 있다는 장점이 존재

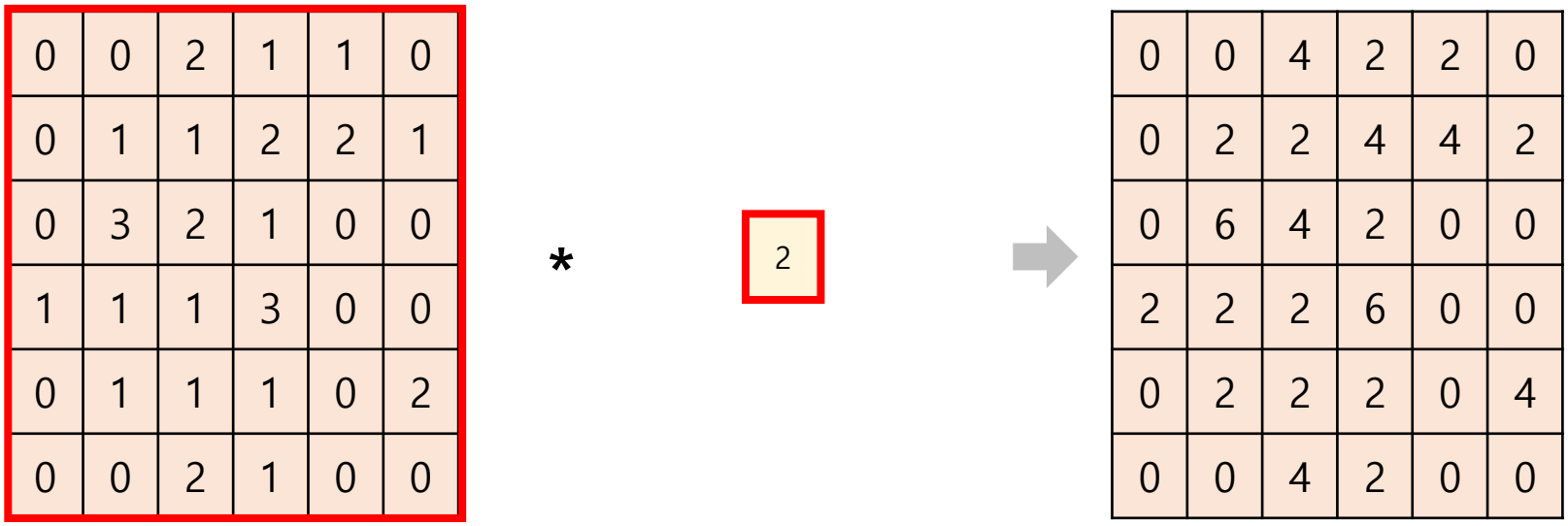


<Input Data> <1x1 Filter> [Output Feature]

Convolutional Neural Network (CNN)

❖ 1x1 Convolutional Filter를 사용했을 때 장점①

➢ 1x1 Convolutional Filter를 사용하면 **입력 데이터 내 모든 정보를 고려**하며 필터 내 parameter가 학습을 진행할 수 있다는 장점이 존재

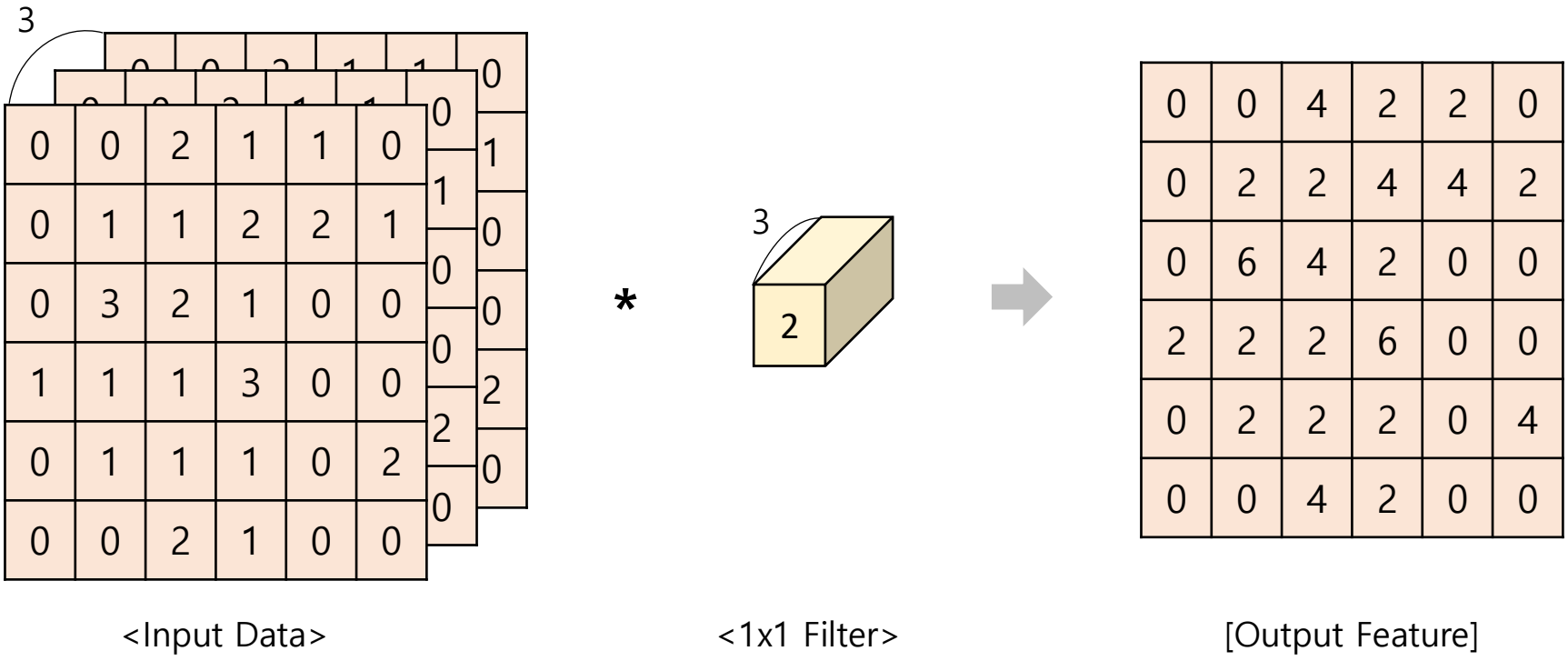


<Input Data> <1x1 Filter> [Output Feature]

Convolutional Neural Network (CNN)

❖ 1x1 Convolutional Filter를 사용했을 때 장점②

- ➢ 1x1 Filter 개수에 따라 **Output 채널 수를 (=차원) 쉽게 조절 가능**함
- ➢ 아래 사례는 Output 채널이 1개일 때 예시임

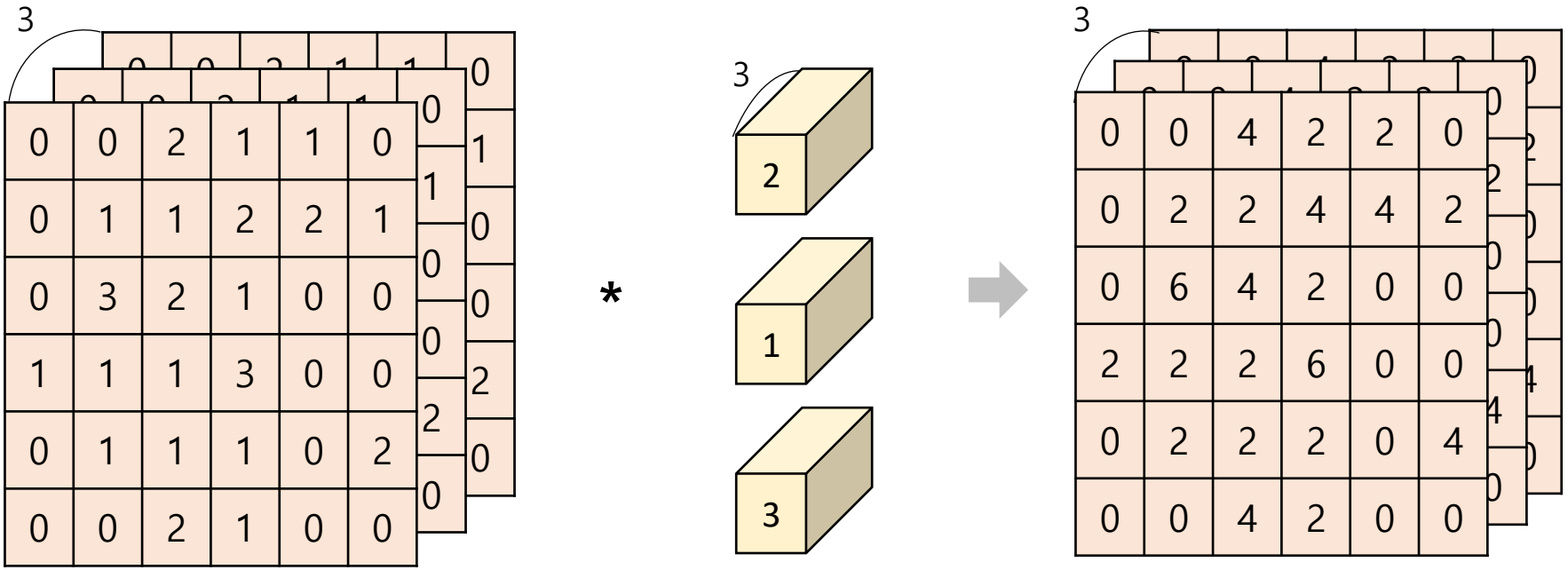


<Input Data> <1x1 Filter> [Output Feature]

Convolutional Neural Network (CNN)

❖ 1x1 Convolutional Filter를 사용했을 때 장점②

- ➢ 1x1 Filter 개수에 따라 **Output 채널 수를 (=차원) 쉽게 조절 가능**함
- ➢ 아래 사례는 Output 채널 수가 Input 채널 수와 동일할 때 예시임

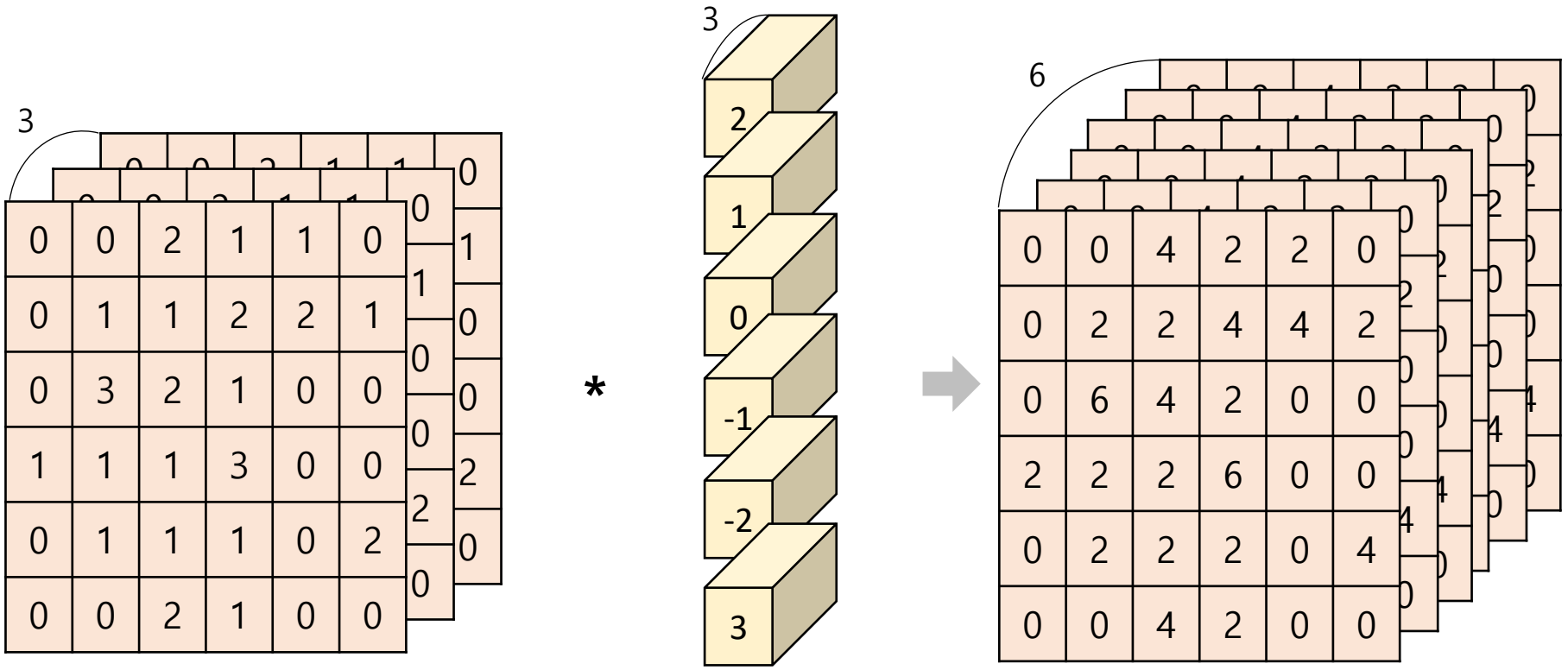


<Input Data> <1x1 Filter> [Output Feature]

Convolutional Neural Network (CNN)

❖ 1x1 Convolutional Filter를 사용했을 때 장점②

➢ 1x1 Filter 개수에 따라 **Output 채널 수를 (=차원) 쉽게 조절 가능**함

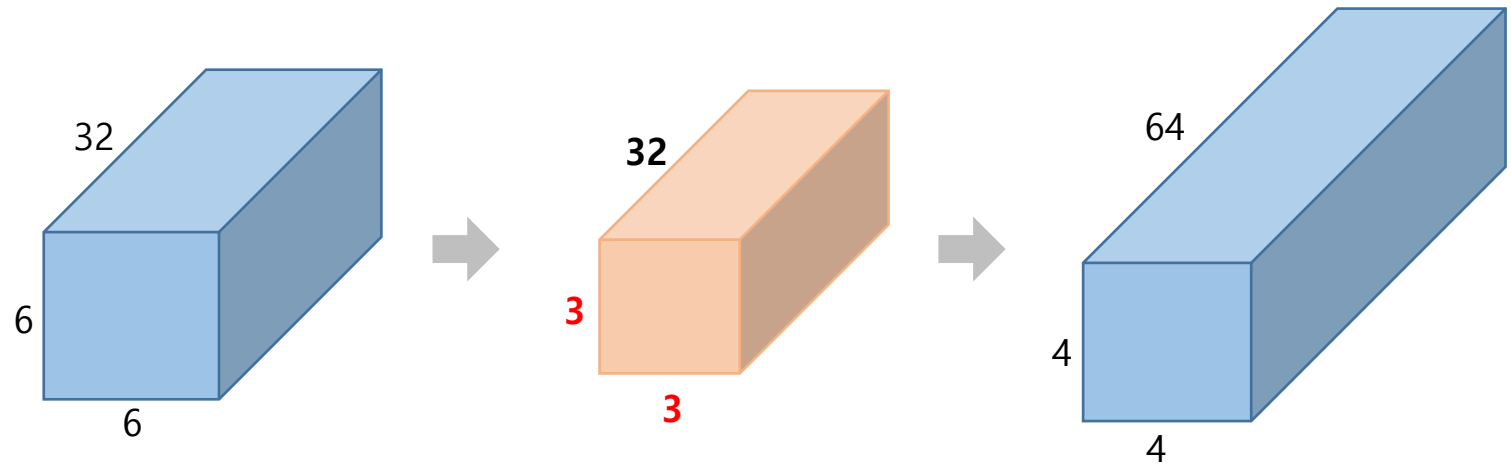


<Input Data> <1x1 Filter> [Output Feature]

Convolutional Neural Network (CNN)

❖ 1x1 Convolutional Filter를 사용했을 때 장점③ (3x3 vs 1x1 계산 비용 비교)

➢ 1x1 Convolutional Filter를 사용할 때 **계산 비용이 매우 작아져 효율적임**



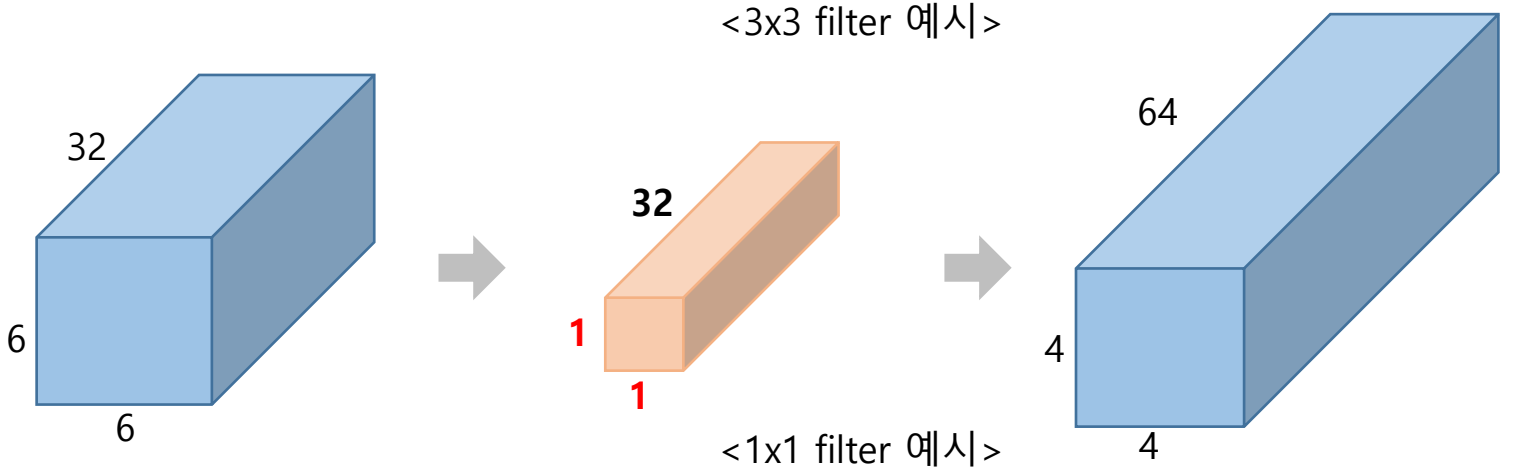
계산 비용

3x3x32x4x4x64

=

294,912

<3x3 filter 예시>



1x1x32x4x4x64

=

32,768

<1x1 filter 예시>

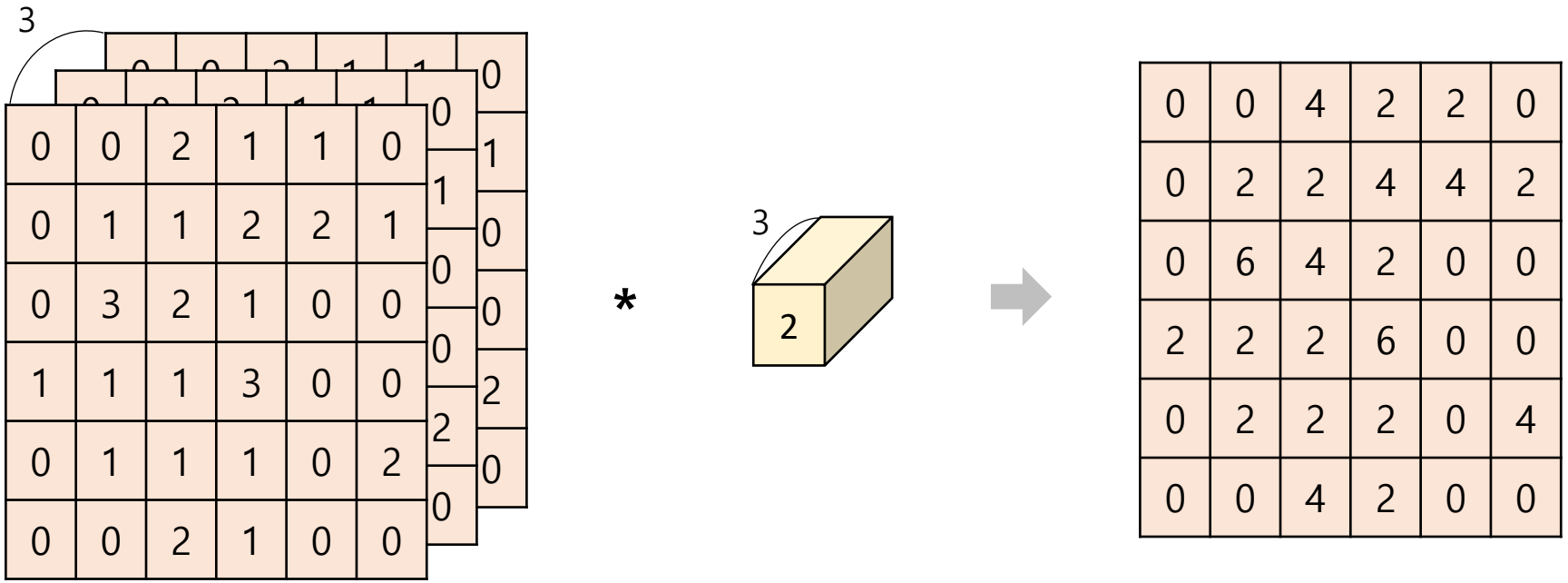
Convolutional Neural Network (CNN)

❖ 1x1 Convolutional Filter를 사용했을 때 장점 요약

① 입력 데이터 내 **모든 정보를 고려해** 학습이 진행됨

② 1x1 Filter 개수에 따라 **Output 채널 수 (=차원) 조절이 쉽게 가능**함

③ 계산 비용이 작아져 **효율성이 높아짐**



<Input Data> <1x1 Filter> [Output Feature]

3. Non-local neural networks

Non-local neural networks (NLNN)

- Non-local neural networks (2018)
  - CNN이 필터를 통해 Local한 영역만 요약하던 단점을 보완
  - 2018년도 CVPR (Computer Vision and Pattern Recognition)에서 발표
  - 1,063회 인용 (20.04.14 기준) → **1,413회 인용 (20.07.23 기준)**

Non-local neural networks

X Wang, R Girshick, A Gupta... - Proceedings of the IEEE ..., 2018 - openaccess.thecvf.com

Both convolutional and recurrent operations are building blocks that process one local neighborhood at a time. In this paper, we present non-local operations as a generic family of building blocks for capturing long-range dependencies. Inspired by the classical non-local ...

☆ 99 1431회 인용 관련 학술자료 전체 14개의 버전 »

/PR paper is the Open Access version, provided by the Computer Vision Foundation.
xcept for this watermark, it is identical to the version available on IEEE Xplore.

Non-local Neural Networks

Xiaolong Wang[1,2*] Ross Girshick[2] Abhinav Gupta[1] Kaiming He[2]
[1]Carnegie Mellon University [2]Facebook AI Research

Abstract

Both convolutional and recurrent operations are building blocks that process one local neighborhood at a time. In this paper, we present non-local operations as a generic family of building blocks for capturing long-range dependencies. Inspired by the classical non-local means method [4] in computer vision, our non-local operation computes the response at a position as a weighted sum of the features at all positions. This building block can be plugged into many computer vision architectures. On the task of video classification, even without any bells and whistles, our non-local models can compete or outperform current competition

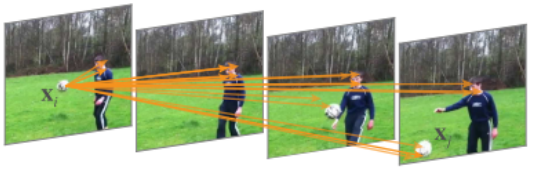


Figure 1. A spacetime ***non-local*** operation in our network trained for video classification in Kinetics. A position $\mathbf{x}_i$'s response is computed by the weighted average of the features of *all* positions $\mathbf{x}_j$ (only the highest weighted ones are shown here). In this example computed by our model, note how it relates the ball in the first frame to the ball in the last two frames. More examples are in Figure 3.

출처: Wang, Xiaolong, et al. "Non-local neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Non-local neural networks (NLNN)

❖ Non-local neural networks 장점 preview

① 데이터 내 Local한 영역이 아닌 **전체 영역을 고려**하며 학습이 가능

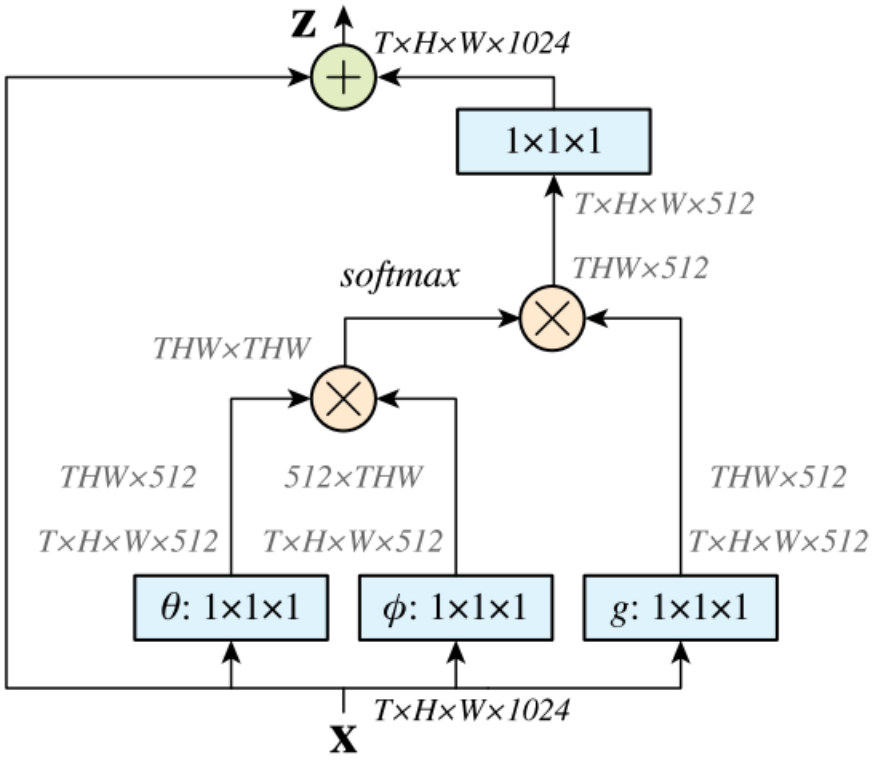
② **기존 CNN 네트워크 구조에 쉽게 적용**이 가능함

③ 예측 결과에 대한 **해석이 가능**함 (Visualization)

Non-local neural networks (NLNN)

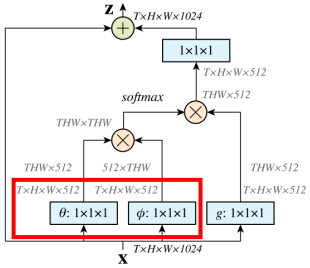
❖ Non-local neural networks 구조

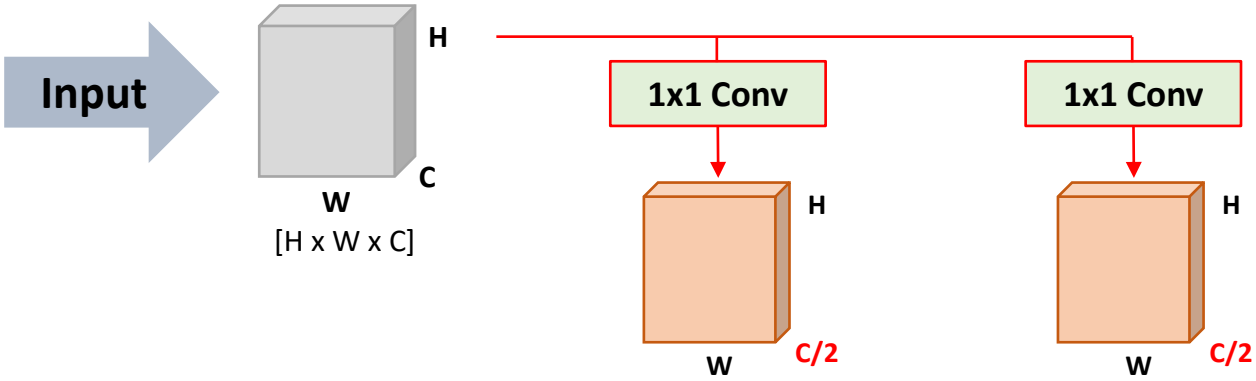
- ➢ 1x1x1 Filter를 활용하여 Input Feature 내 중요 정보를 추출함
- ➢ 최종적으로 Input Size (TxHxWx1024)와 같은 Size를 갖는 Output 도출



출처: Wang, Xiaolong, et al. "Non-local neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

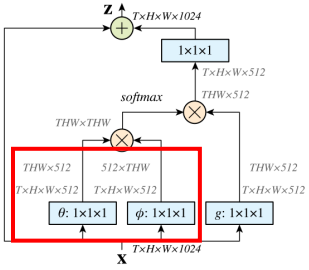
Non-local neural networks (NLNN)

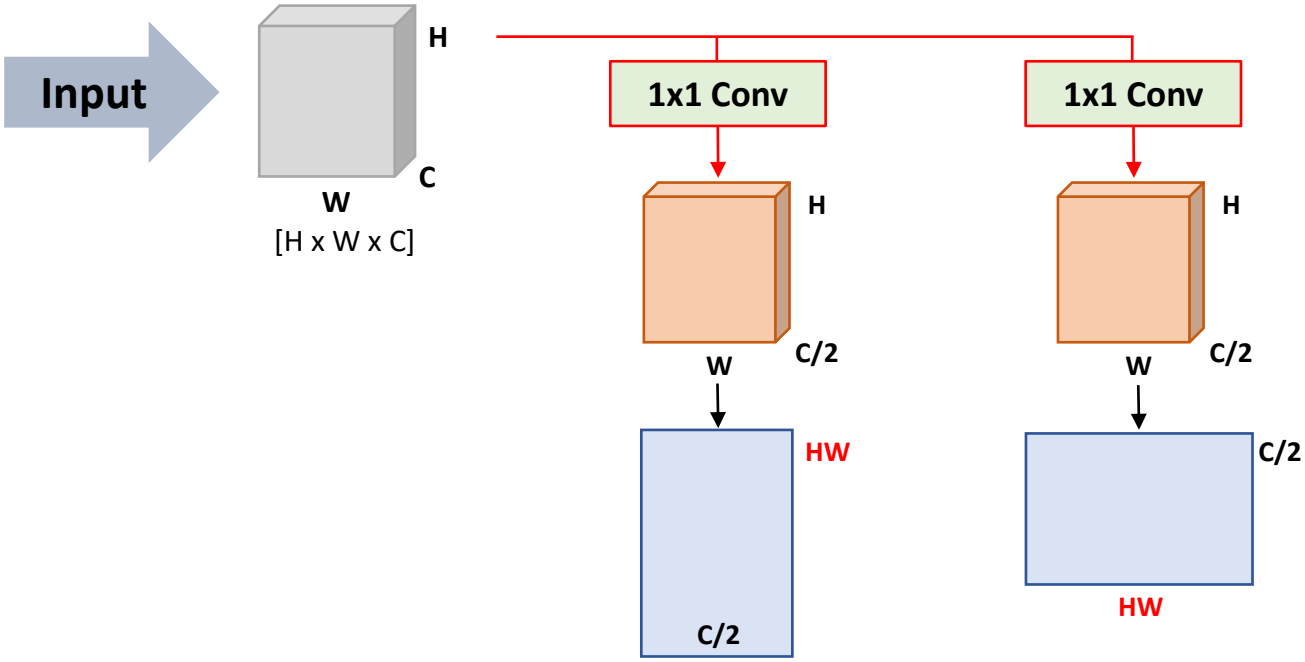




- 1x1 Convolutional Filter를 활용하여 채널을 절반으로 줄임
- 2개 Filter를 활용하여 Feature map 2개 생성

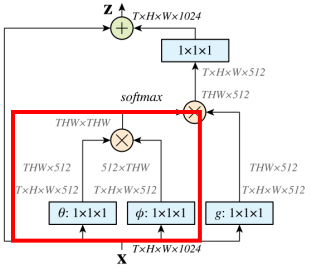
Non-local neural networks (NLNN)

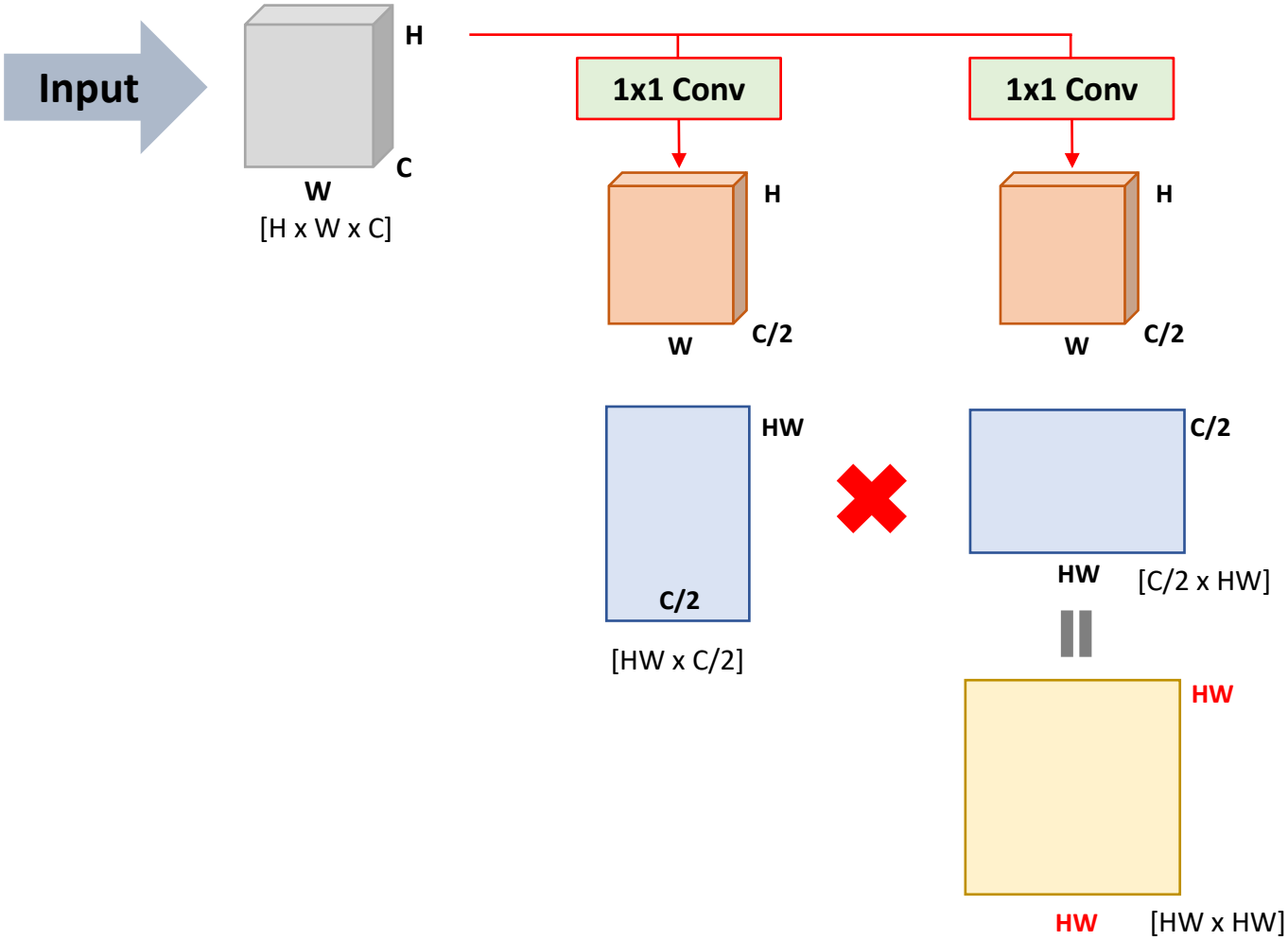




- 2 feature map을 각각 (HW x Channel), (Channel x HW)로 reshape 진행

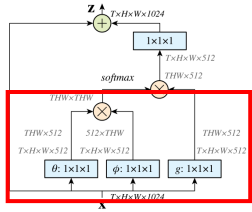
Non-local neural networks (NLNN)

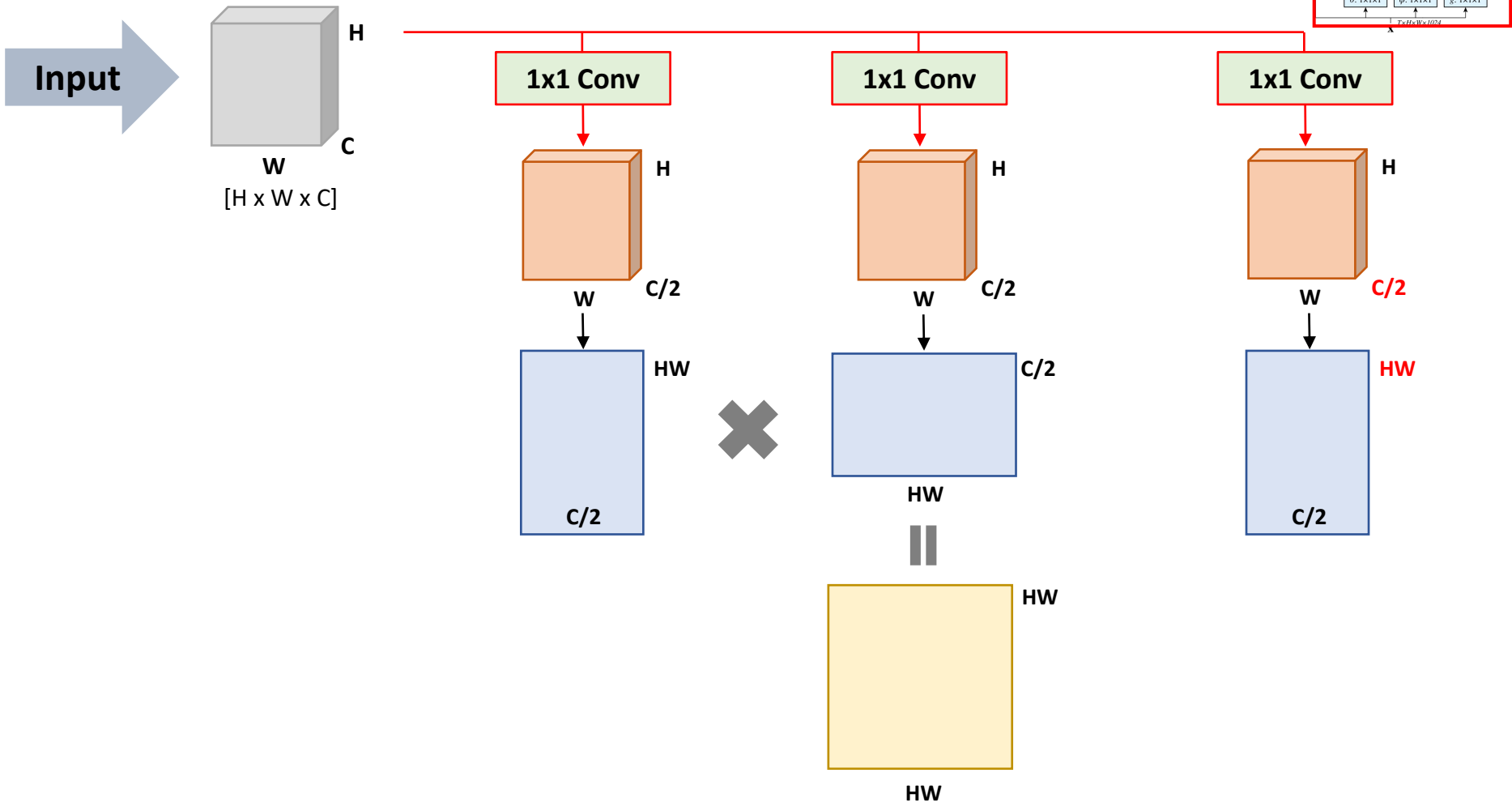




- 2 feature map에 행렬곱 연산을 진행하여 (HW x HW) feature map 생성
- HW x HW feature map은 1x1 filter로 요약 된 모든 pixel 간의 관계 고려함

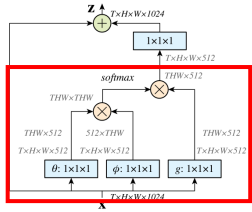
Non-local neural networks (NLNN)

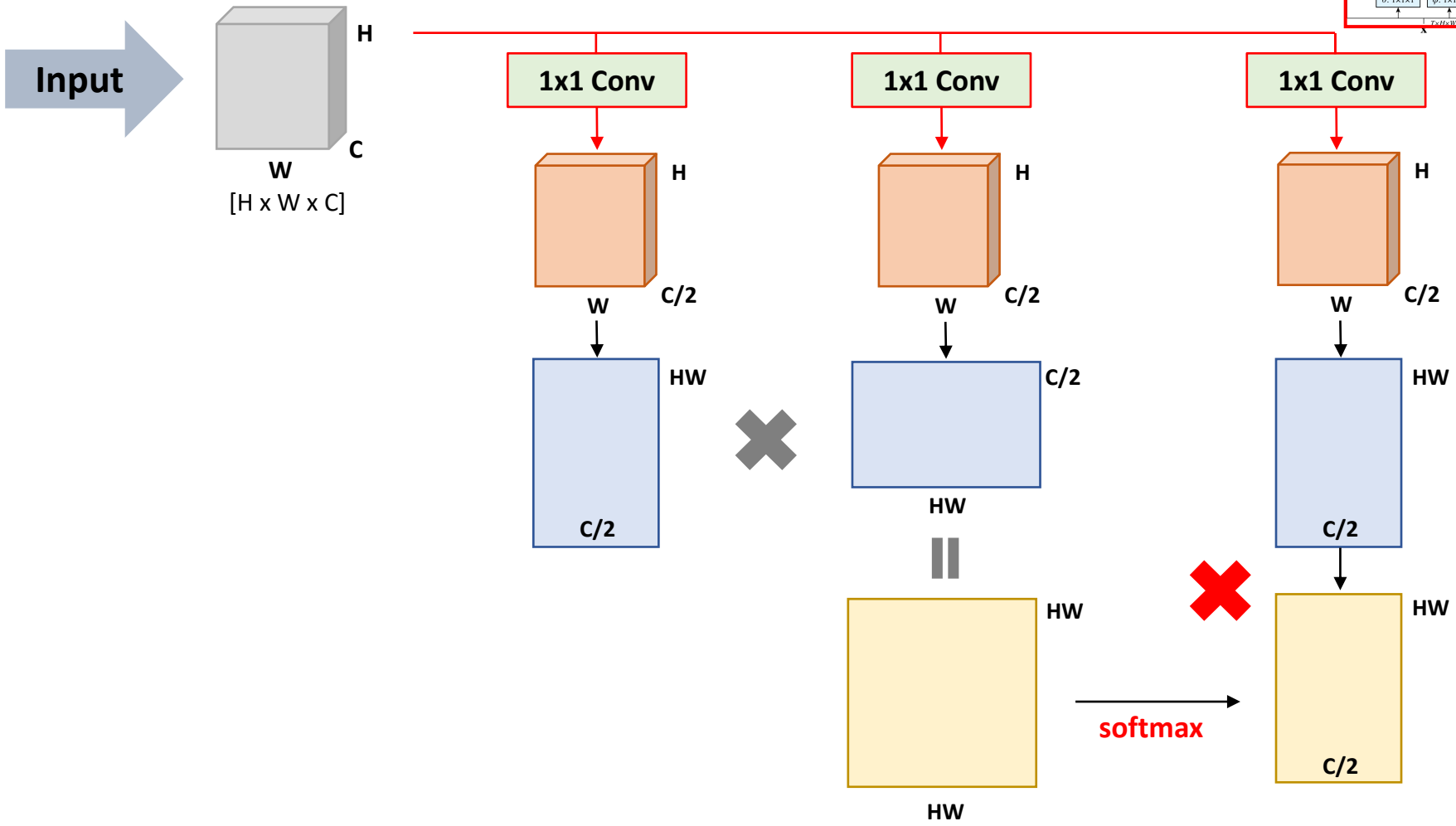




- 1x1 Convolutional Filter를 추가로 더 활용하여 채널 수를 절반으로 요약
- 이후 (Channel x HW)로 Reshape 진행

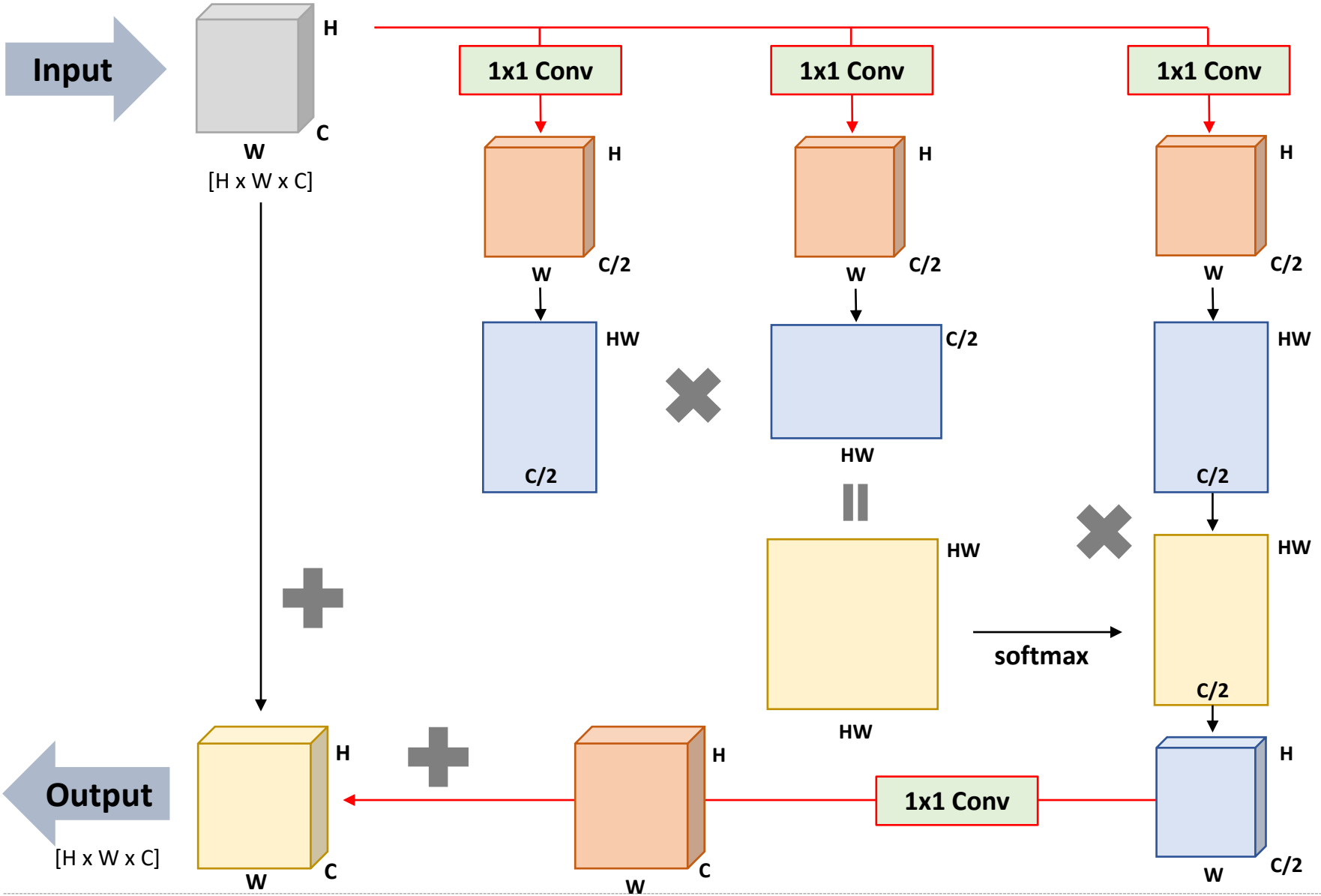
Non-local neural networks (NLNN)



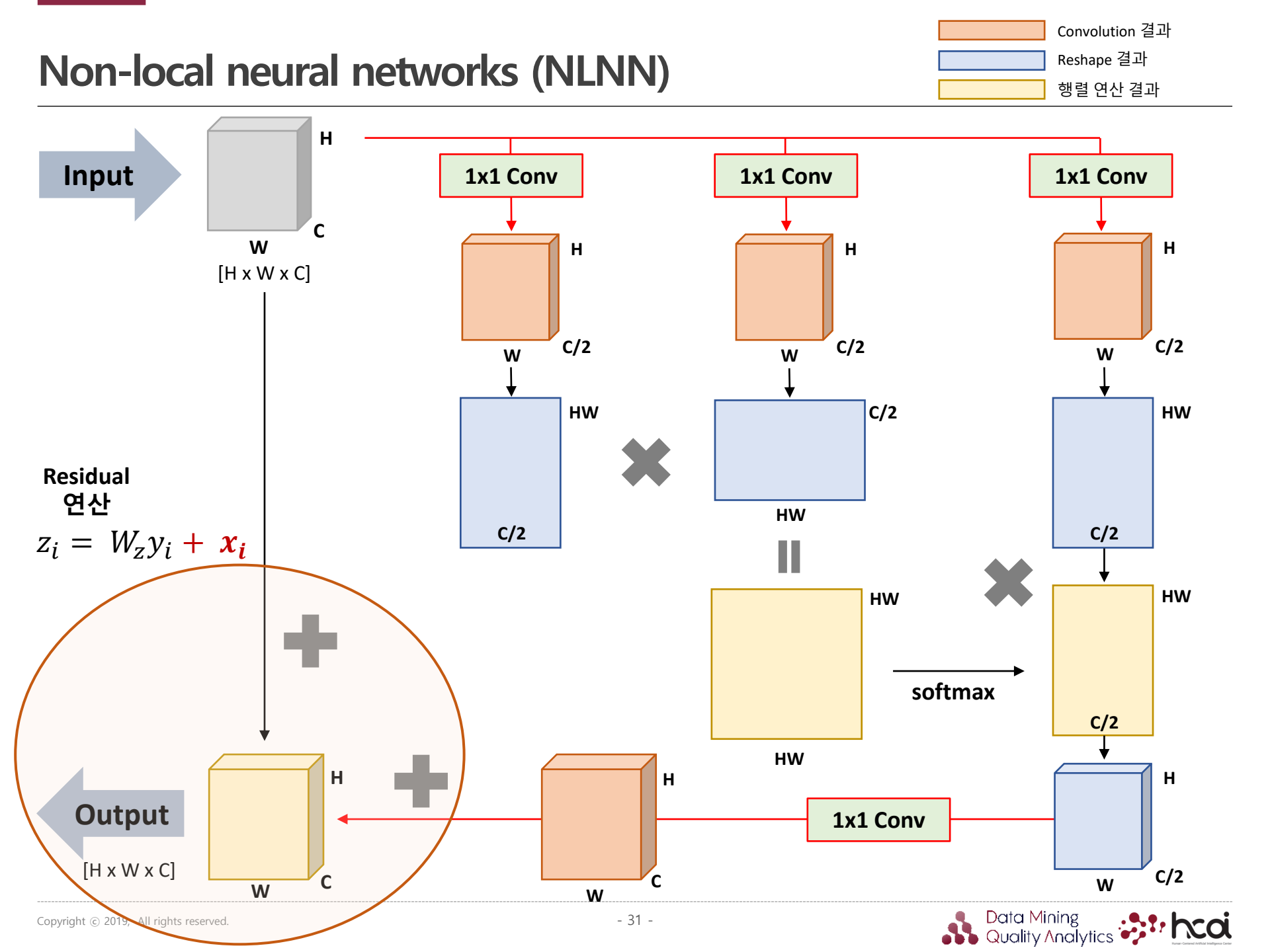


- (HW x HW) feature map에 row별 softmax를 취해 확률값 도출
- (Channel x HW) feature map과 행렬곱 연산 진행

Non-local neural networks (NLNN)



Non-local neural networks (NLNN)



Non-local neural networks (NLNN)

$x_i, x_j \rightarrow Input\ data$

$W \rightarrow 1 \times 1\ Convlutional\ Filter$

- 1x1 filter를 통해 학습된 Weight에 input값을 곱함

$\theta(x_i) = W_\theta x_i \qquad \phi(x_j) = W_\phi x_j$

-Embedded Gaussian-

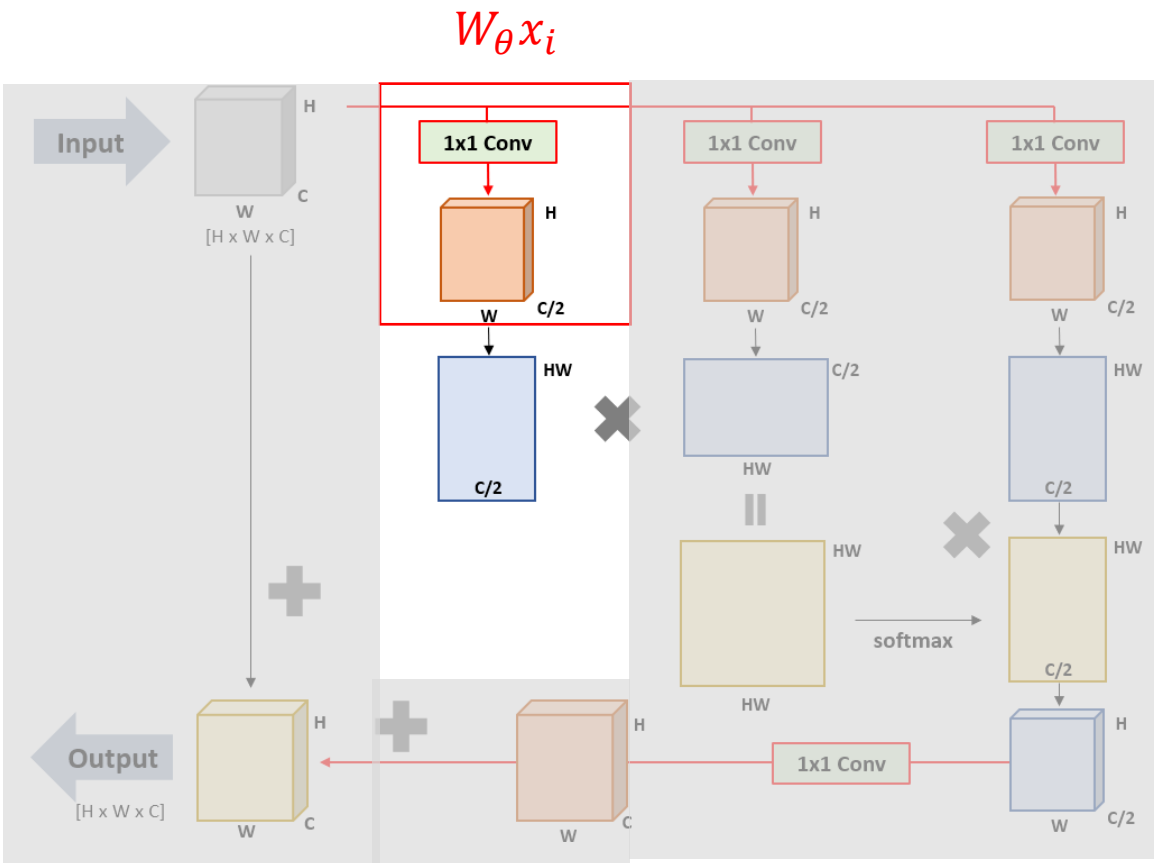
$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}$$

$$g(x_j) = W_g x_j$$

$$y_i = \frac{1}{C(x)} f(x_i, x_j) g(x_j)$$

$$set\ C(x) = \sum_{\forall j} f(x_i, x_j)$$

$$z_i = W_z y_i + x_i$$



Non-local neural networks (NLNN)

$x_i, x_j \rightarrow Input\ data$

$W \rightarrow 1 \times 1\ Convlutional\ Filter$

- 1x1 filter를 통해 학습된 Weight에 input값을 곱함

$$\theta(x_i) = W_\theta x_i \qquad \phi(x_j) = W_\phi x_j$$

-Embedded Gaussian-

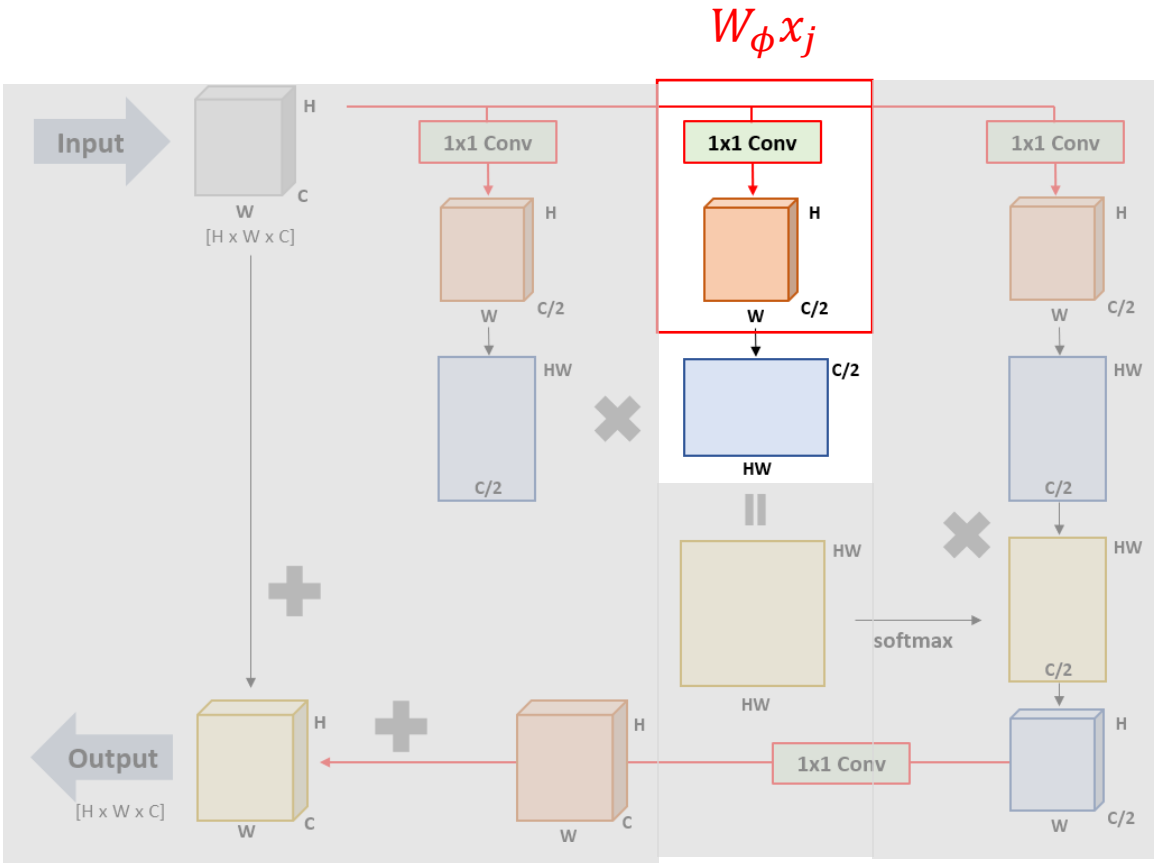
$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}$$

$$g(x_j) = W_g x_j$$

$$y_i = \frac{1}{C(x)} f(x_i, x_j) g(x_j)$$

$$set\ C(x) = \sum_{\forall j} f(x_i, x_j)$$

$$z_i = W_z y_i + x_i$$



Non-local neural networks (NLNN)

$x_i, x_j \rightarrow Input\ data$

$W \rightarrow 1 \times 1\ Convlutional\ Filter$

- 모든 pixel간 관계를 계산

$$\theta(x_i) = W_\theta x_i \qquad \phi(x_j) = W_\phi x_j$$

-Embedded Gaussian-

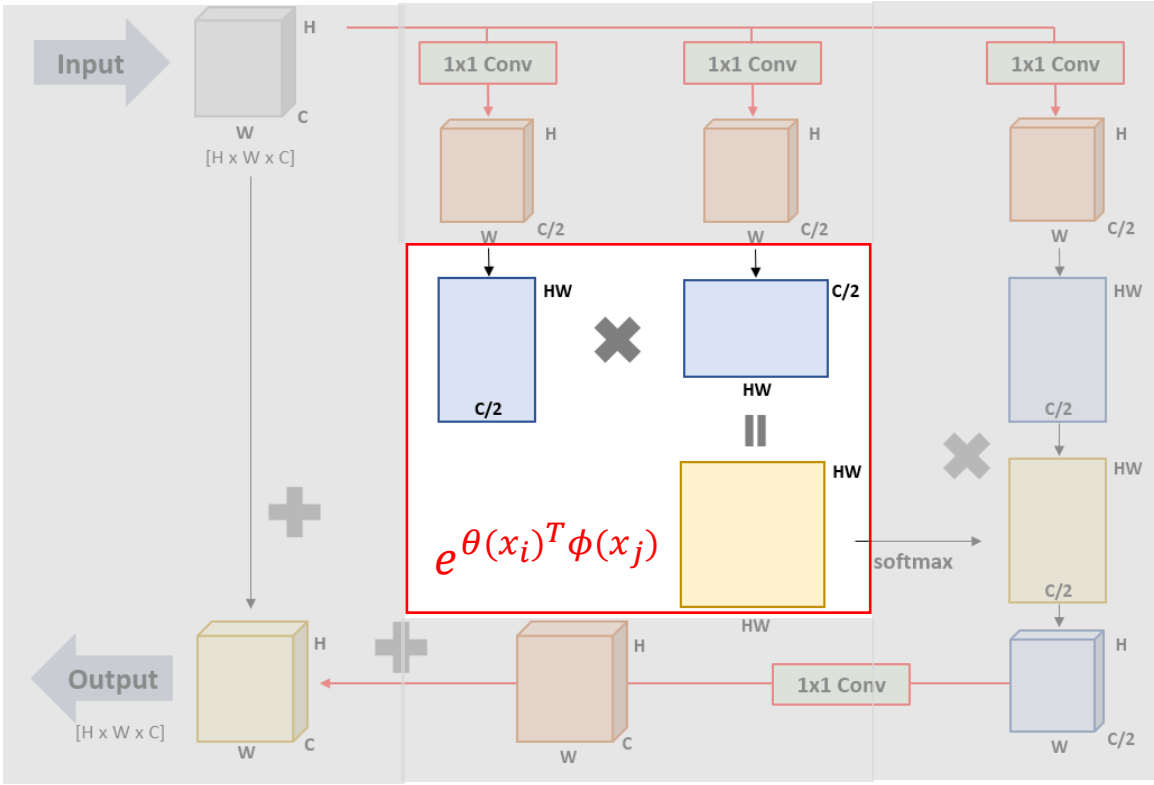
$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}$$

$$g(x_j) = W_g x_j$$

$$y_i = \frac{1}{C(x)} f(x_i, x_j) g(x_j)$$

$$set\ C(x) = \sum_{\forall j} f(x_i, x_j)$$

$$z_i = W_z y_i + x_i$$



Non-local neural networks (NLNN)

$x_i, x_j \rightarrow Input\ data$

$W \rightarrow 1 \times 1\ Convlutional\ Filter$

- 새로운 1x1 filter 내 weight와 input data를 곱함

$$\theta(x_i) = W_\theta x_i \qquad \phi(x_j) = W_\phi x_j$$

-Embedded Gaussian-

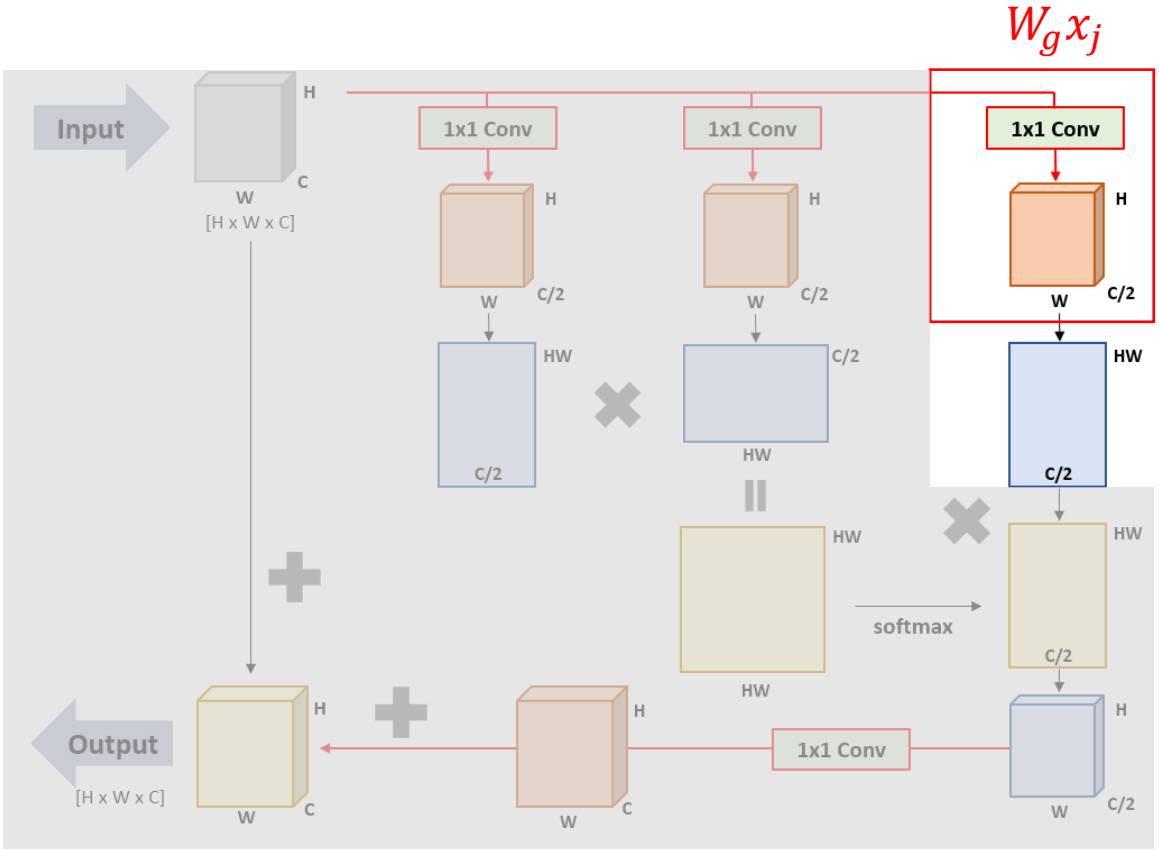
$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}$$

$$g(x_j) = W_g x_j$$

$$y_i = \frac{1}{C(x)} f(x_i, x_j) g(x_j)$$

$$set\ C(x) = \sum_{\forall j} f(x_i, x_j)$$

$$z_i = W_z y_i + x_i$$



Non-local neural networks (NLNN)

$x_i, x_j \rightarrow Input\ data$

$W \rightarrow 1 \times 1\ Convlutional\ Filter$

- Softmax 함수식을 통해 Row별 확률값 도출

$$\theta(x_i) = W_\theta x_i \qquad \phi(x_j) = W_\phi x_j$$

-Embedded Gaussian-

$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}$$

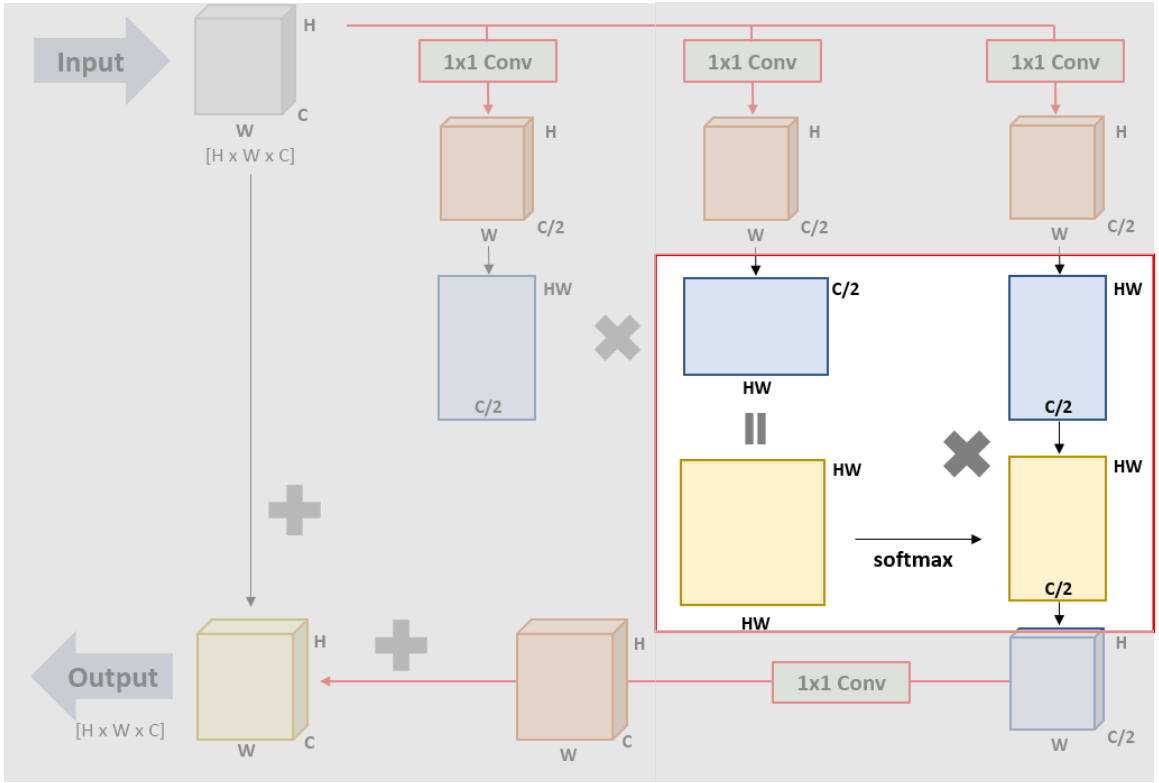
$$g(x_j) = W_g x_j$$

SoftMax function

$$y_i = \frac{1}{C(x)} f(x_i, x_j) g(x_j)$$

$$set\ C(x) = \sum_{\forall j} f(x_i, x_j)$$

$$z_i = W_z y_i + x_i$$



Non-local neural networks (NLNN)

$x_i, x_j \rightarrow Input\ data$

$W \rightarrow 1 \times 1\ Convlutional\ Filter$

- 이전까지 수식을 모두 정리해 y_i 형태를 확인

$$\theta(x_i) = W_\theta x_i \qquad \phi(x_j) = W_\phi x_j$$

-Embedded Gaussian-

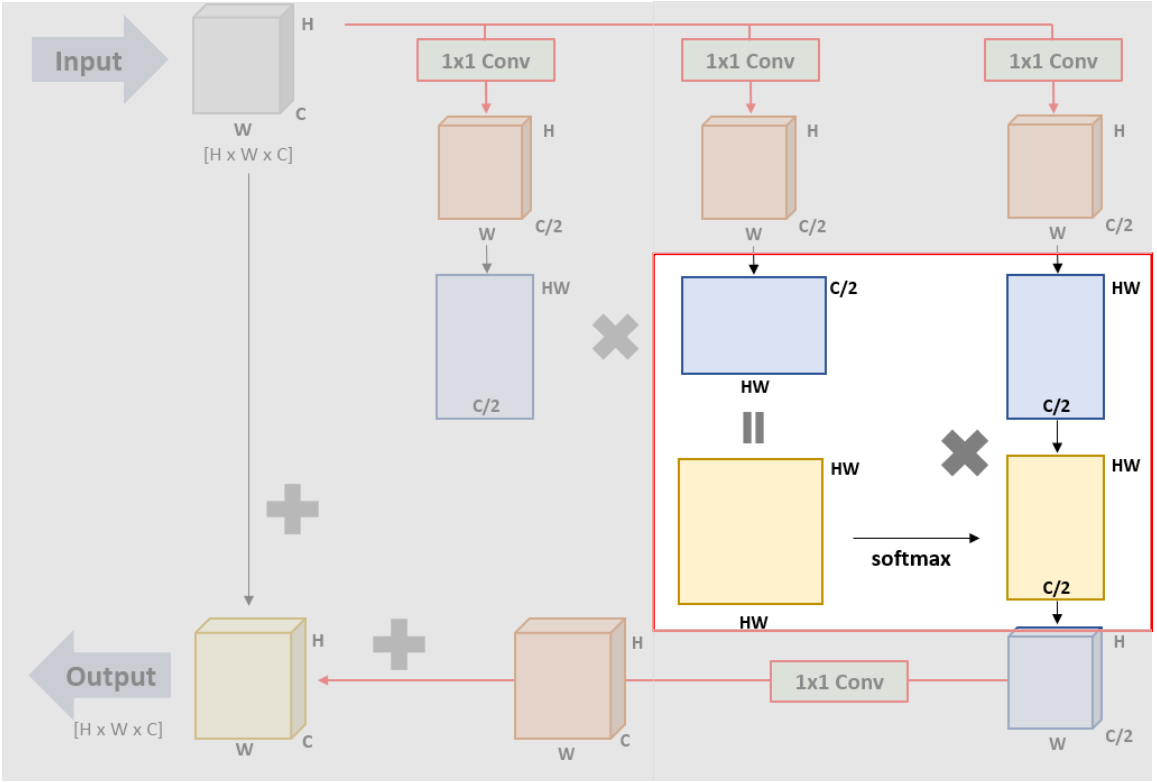
$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}$$

$$g(x_j) = W_g x_j$$

SoftMax function

$$y_i = \frac{1}{C(x)} f(x_i, x_j) g(x_j)$$

$$set\ C(x) = \sum_{\forall j} f(x_i, x_j)$$



Non-local neural networks (NLNN)

$x_i, x_j \rightarrow Input\ data$

$W \rightarrow 1 \times 1\ Convlutional\ Filter$

- 이전까지 수식을 모두 정리해 y_i 형태를 확인

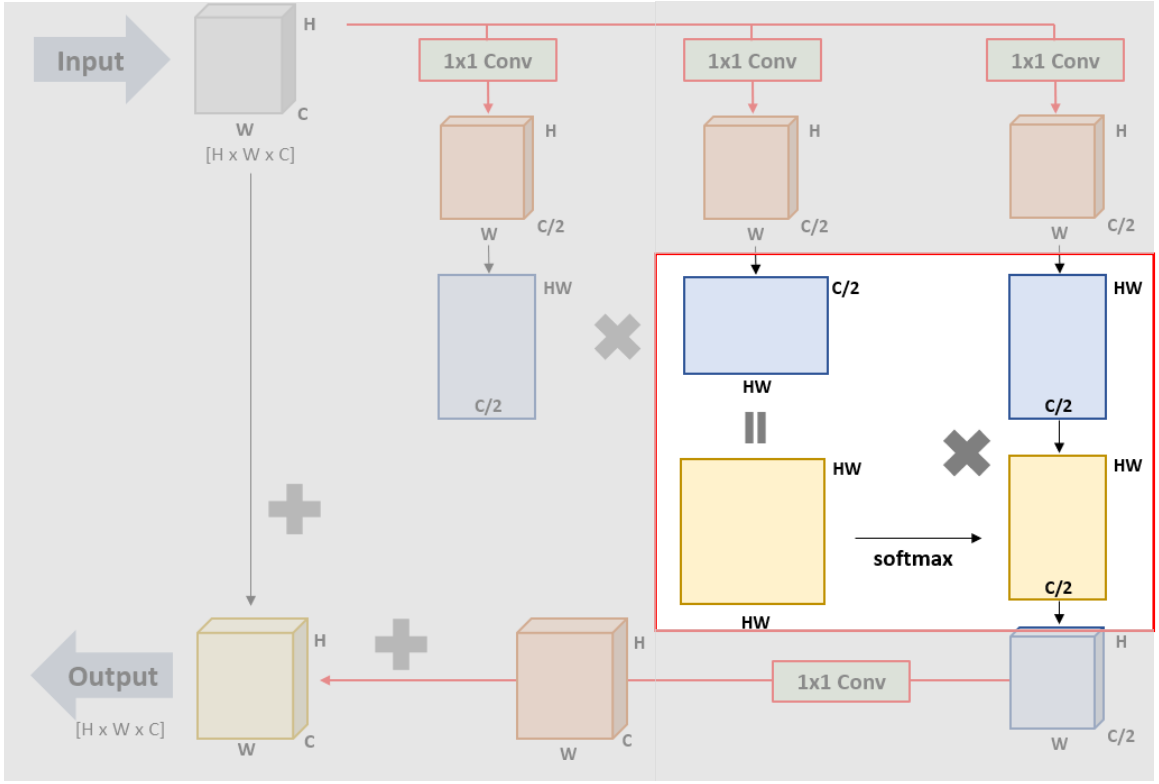
-Embedded Gaussian-

$$f(x_i, x_j) = e^{(W_\theta x_i)^T W_\phi x_j}$$

$$g(x_j) = W_g x_j$$

SoftMax function

$$y_i = \frac{1}{C(x)} f(x_i, x_j) g(x_j)$$



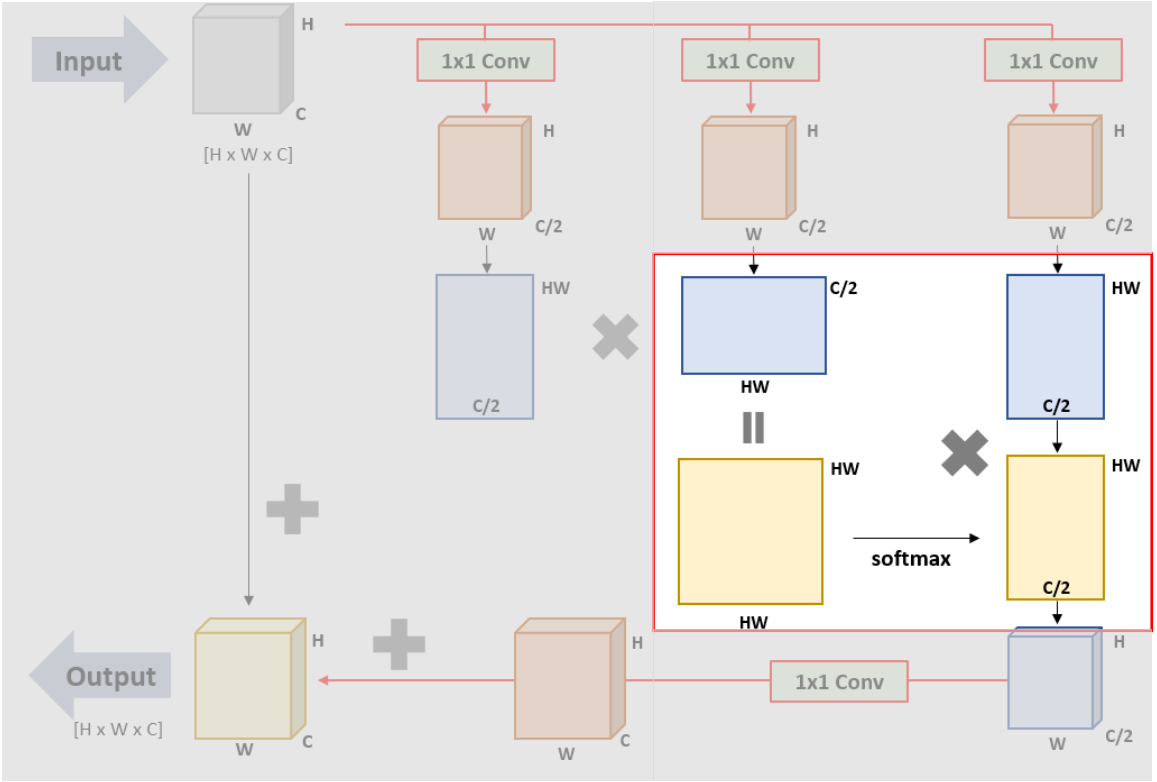
Non-local neural networks (NLNN)

$x_i, x_j \rightarrow Input\ data$

$W \rightarrow 1 \times 1\ Convlutional\ Filter$

$$y_i = \frac{1}{C(x)} e^{(W_\theta x_i)^T W_\phi x_j} W_g x_j \quad \Rightarrow \quad Y = softmax(X^T W_\theta^T W_\phi X) W_g X$$

$$set\ \ C(x) = \sum_{\forall j} f(x_i, x_j)$$



출처: 강현규"Visual Attention"-DMQA세미나, http://jalammar.github.io

Non-local neural networks (NLNN)

$x_i, x_j \rightarrow Input\ data$

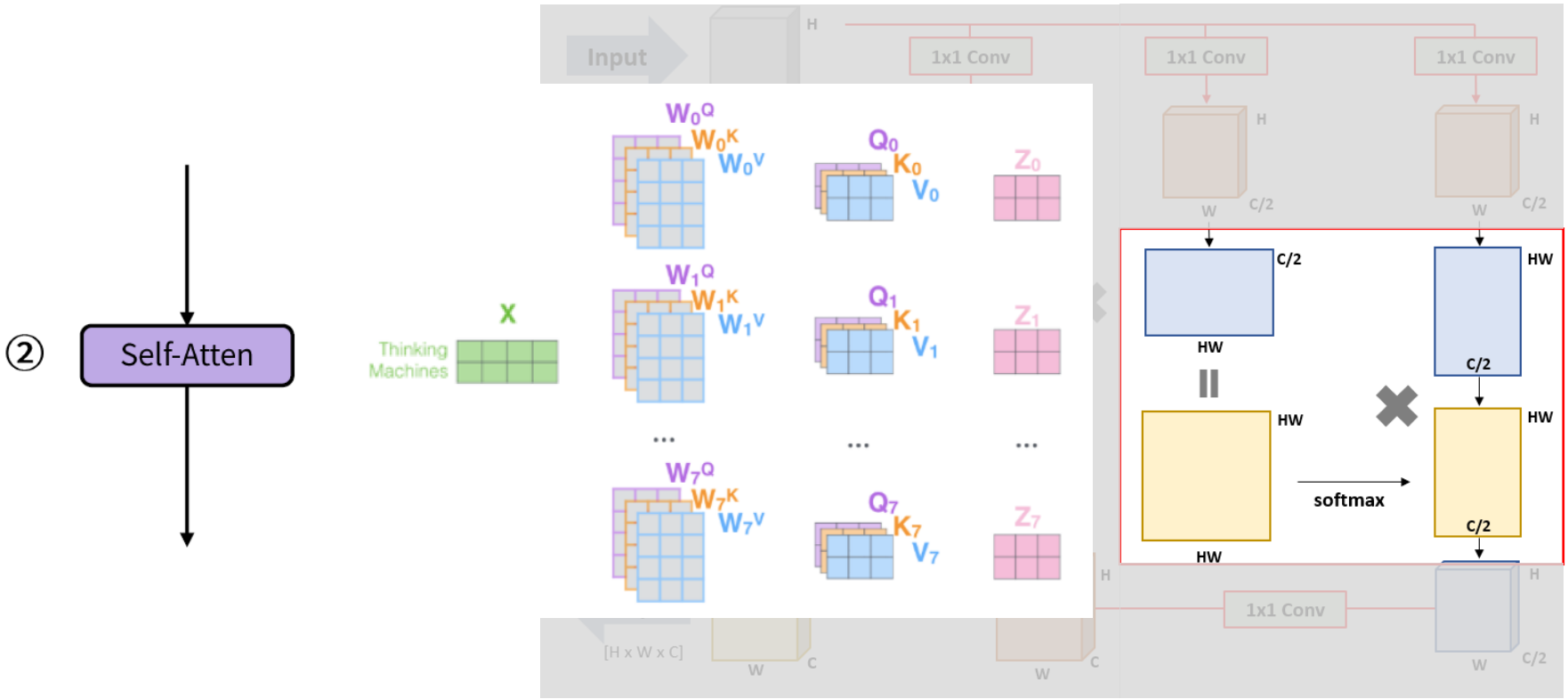
$W \rightarrow 1 \times 1\ Convlutional\ Filter$

$$y_i = \frac{1}{C(x)} e^{(W_\theta x_i)^T W_\phi x_j} W_g x_j \quad \Rightarrow \quad Y = softmax(X^T W_\theta^T W_\phi X) W_g X$$

$$set\ \ C(x) = \sum_{\forall j} f(x_i, x_j)$$

Self Attention

$$A(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



출처: 강현규"Visual Attention"-DMQA세미나, http://jalammar.github.io

Non-local neural networks (NLNN)

$x_i, x_j \rightarrow Input\ data$

$W \rightarrow 1 \times 1\ Convlutional\ Filter$

$$\theta(x_i) = W_\theta x_i \qquad \phi(x_j) = W_\phi x_j$$

-Embedded Gaussian-

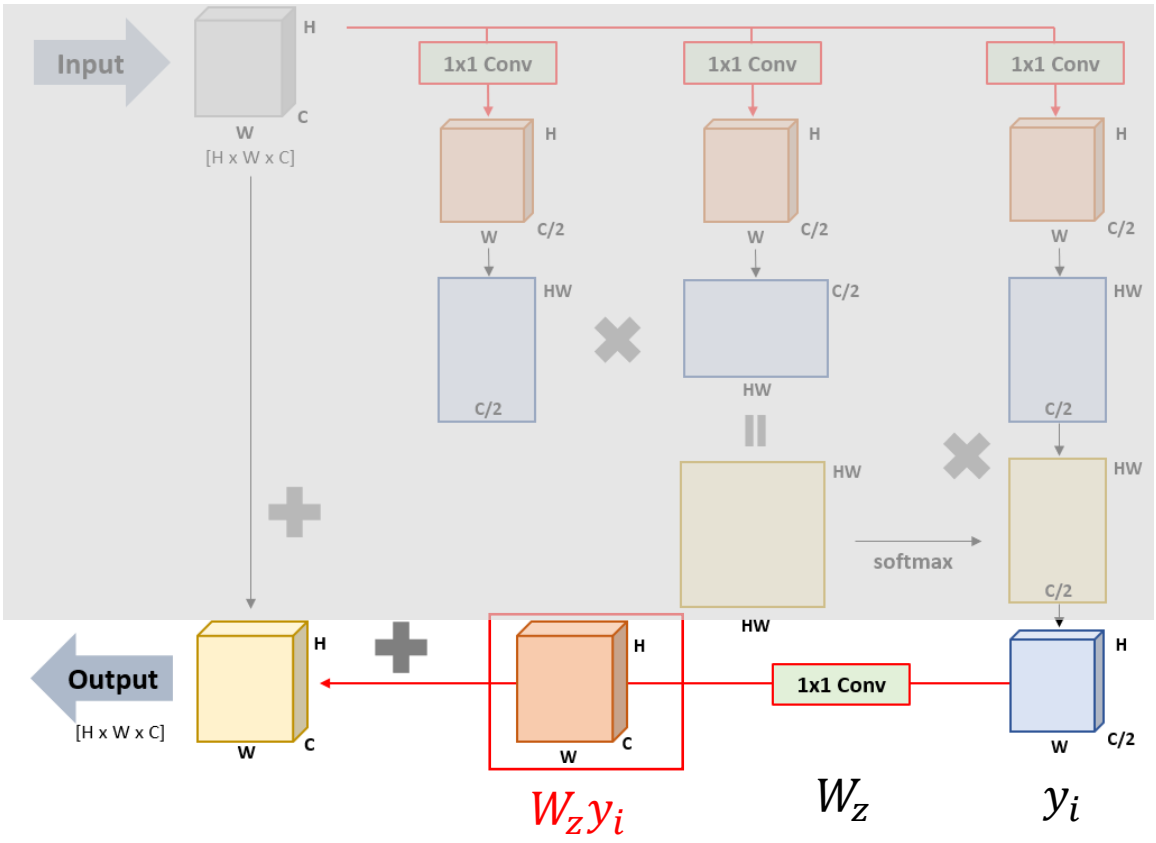
$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}$$

$$g(x_j) = W_g x_j$$

$$y_i = \frac{1}{C(x)} f(x_i, x_j) g(x_j)$$

$$set\ C(x) = \sum_{\forall j} f(x_i, x_j)$$

$$z_i = W_z y_i + x_i$$



Non-local neural networks (NLNN)

$x_i, x_j \rightarrow Input\ data$

$W \rightarrow 1 \times 1\ Convlutional\ Filter$

$$\theta(x_i) = W_\theta x_i \qquad \phi(x_j) = W_\phi x_j$$

-Embedded Gaussian-

$$f(x_i, x_j) = e^{\theta(x_i)^T \phi(x_j)}$$

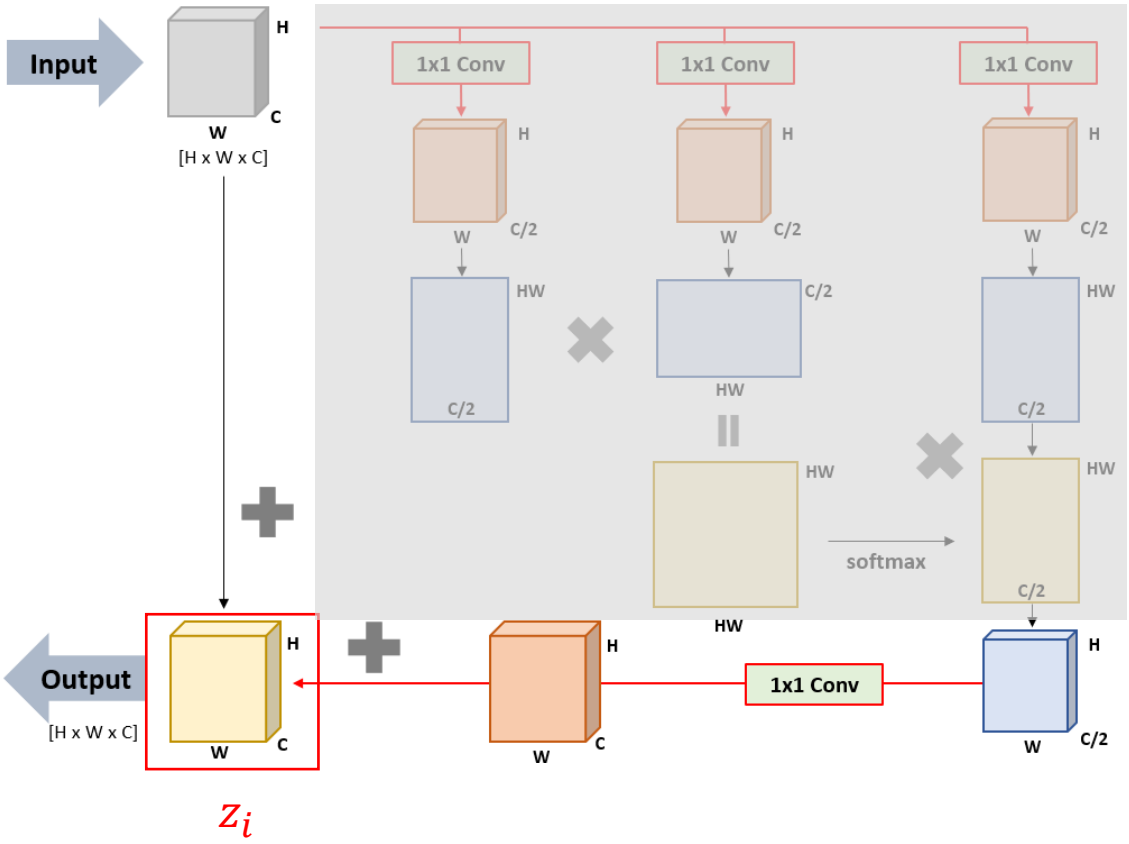
$$g(x_j) = W_g x_j$$

$$y_i = \frac{1}{C(x)} f(x_i, x_j) g(x_j)$$

$$set\ C(x) = \sum_{\forall j} f(x_i, x_j)$$

$$z_i = W_z y_i + x_i$$

-Residual 연산-



Non-local neural networks (NLNN)

❖ Non-local neural networks 장점

① 데이터 내 Local한 영역이 아닌 **전체 영역을 고려**하며 학습이 가능

→ **1x1 Convolutional Filter**를 다양하게 활용

② **기존 CNN 네트워크 구조에 쉽게 적용**이 가능함

→ Non-local neural networks 내 **Input 크기와 Output 크기가 동일**함

③ 예측 결과에 대한 **해석이 가능**함 (Visualization)

→ **Self-Attention**과 동일한 구조가 네트워크 내에 존재

Non-local neural networks (NLNN)

❖ Non-local neural networks 결과

- ➢ Pixel간 관계를 구하는 형식을 다양하게 적용한 결과 큰 차이 없음 – (a)
- ➢ Resnet-50을 기준으로 Layer 중간에 다양하게 넣어 실험 진행 – (b)

model, R50	top-1	top-5
C2D baseline	71.8	89.7
Gaussian	72.5	90.2
Gaussian, embed	72.7	**90.5**
dot-product	**72.9**	90.3
concatenation	72.8	**90.5**

(a) **Instantiations**: 1 non-local block of different types is added into the C2D baseline. All entries are with ResNet-50.

model, R50	top-1	top-5
baseline	71.8	89.7
res_2	72.7	90.3
res_3	**72.9**	90.4
res_4	72.7	**90.5**
res_5	72.3	90.1

(b) **Stages**: 1 non-local block is added into different stages. All entries are with ResNet-50.

출처: Wang, Xiaolong, et al. "Non-local neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Non-local neural networks (NLNN)

❖ Non-local neural networks 결과

➢ Resnet-50과 Resne-101 모델 사이에 1개, 5개, 10개 Non-local blocks을 넣어 실험 진행

model		top-1	top-5
R50	baseline	71.8	89.7
	1-block	72.7	90.5
	5-block	73.8	91.0
	10-block	**74.3**	**91.2**
R101	baseline	73.1	91.0
	1-block	74.3	91.3
	5-block	**75.1**	**91.7**
	10-block	**75.1**	91.6

(c) **Deeper non-local models**: we compare 1, 5, and 10 non-local blocks added to the C2D baseline. We show ResNet-50 (top) and ResNet-101 (bottom) results.

출처: Wang, Xiaolong, et al. "Non-local neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Non-local neural networks (NLNN)

❖ Non-local neural networks 결과

➢ 이미지(C2D = 2D-CNN)분석 뿐 아니라 영상 분석(I3D = 3D-CNN)에서도 Non-local block을 넣었을 때 더 성능이 좋아짐을 확인

	model	top-1	top-5
R50	C2D baseline	71.8	89.7
	I3D	73.3	90.7
	NL I3D	**74.9**	**91.6**
R101	C2D baseline	73.1	91.0
	I3D	74.4	91.1
	NL I3D	**76.0**	**92.1**

(f) **Non-local 3D ConvNet**: 5 non-local blocks are added on top of our best I3D models. These results show that non-local operations are complementary to 3D convolutions.

	model	top-1	top-5
R50	C2D baseline	73.8	91.2
	I3D	74.9	91.7
	NL I3D	**76.5**	**92.6**
R101	C2D baseline	75.3	91.8
	I3D	76.4	92.7
	NL I3D	**77.7**	**93.3**

(g) **Longer clips**: we fine-tune and test the models in Table 2f on the 128-frame clips. The gains of our non-local operations are consistent.

출처: Wang, Xiaolong, et al. "Non-local neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Non-local neural networks (NLNN)

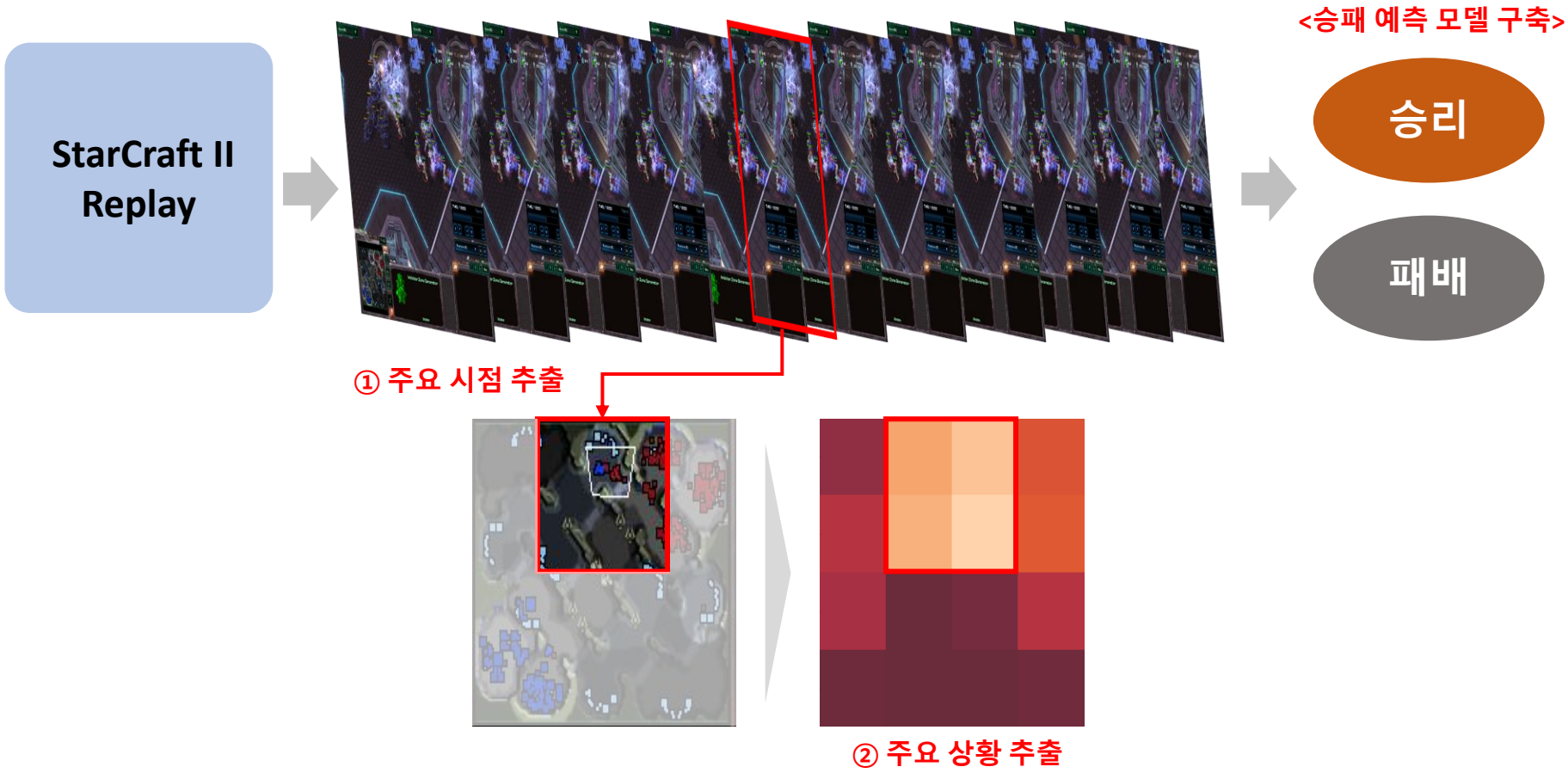
- Non-local neural networks 결과
  - 영상 내 Attention score를 활용하여 Visualization한 결과

Figure 3. Examples of the behavior of a non-local block in res_3 computed by a 5-block non-local model trained on Kinetics. These examples are from held-out validation videos. The starting point of arrows represents one $\mathbf{x}_i$, and the ending points represent $\mathbf{x}_j$. The 20 highest weighted arrows for each $\mathbf{x}_i$ are visualized. The 4 frames are from a 32-frame input, shown with a stride of 8 frames. These visualizations show how the model finds related clues to support its prediction.

출처: Wang, Xiaolong, et al. "Non-local neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

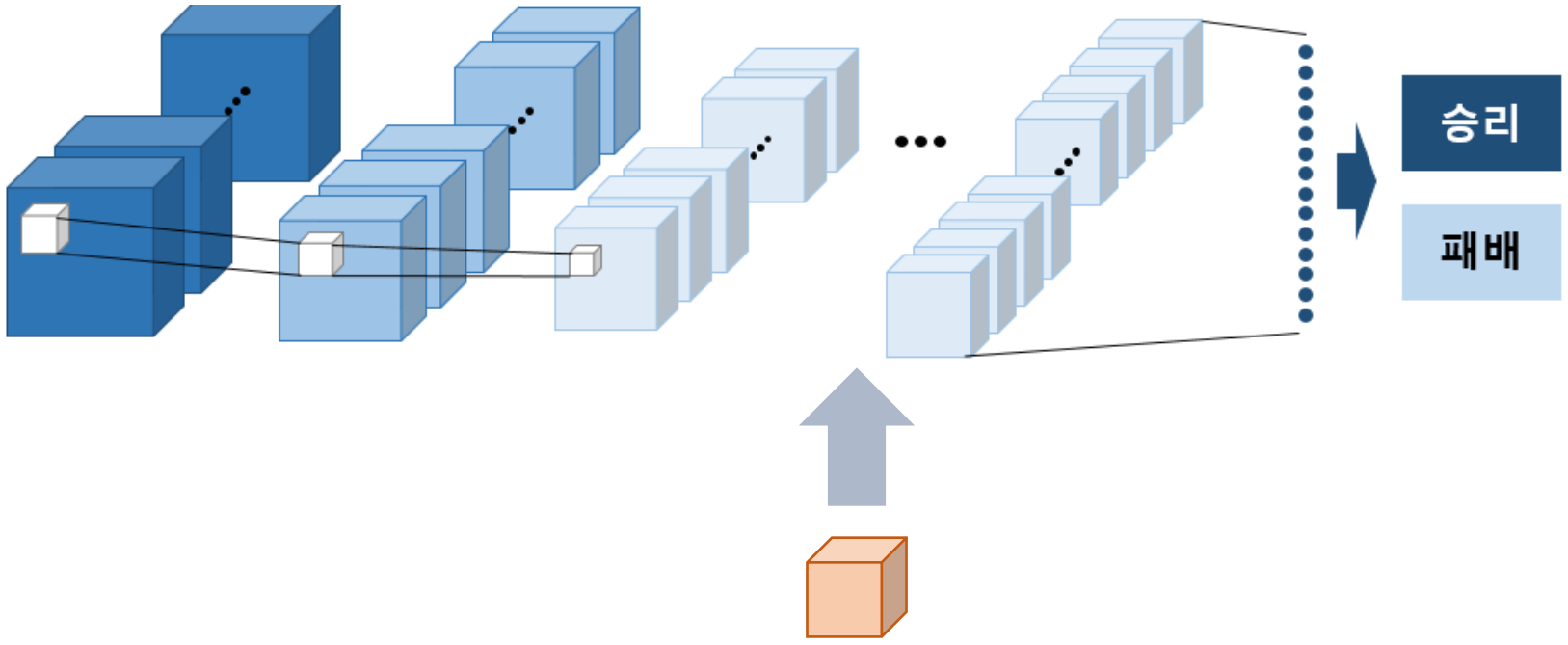
Non-local neural networks – 실제 적용 사례

- ❖ StarCraft II 경기 내 주요 시점 및 주요 상황 추출 연구
- ❖ 리플레이 데이터를 활용하여 승패 예측 모델 구축 후 주요 요소 추출



Non-local neural networks – 실제 적용 사례

❖ 기존 구축했던 3D-Resnet 모델의 마지막 Layer전에 Non-local block 추가



Non-local neural networks – 실제 적용 사례

- ❖ 게임 승패 예측 모델 적용시 3D-Resnet 대비 약 0.5% 좋은 성능을 보임
- ❖ Non-local block을 모델 맨 마지막에 넣은 결과가 가장 좋음

  → 적절한 위치 및 개수는 실험을 통해 찾아야 함

Model	Non-local위치	학습 정확도 (5회 평균)	학습 정확도 (최대)	Epoch
3D-Resnet	X	88.6% (±0.001)	88.8%	23
Non-local neural networks (Batch Norm 사용)	마지막	88.4% (±0.005)	89.0%	43
Non-local neural networks (Batch Norm 제외)	마지막	89.1% (±0.004)	89.6%	20
Non-local neural networks (Batch Norm 제외)	중간과 마지막	82.7% (±0.007)	83.4%	17

Conclusion

❖ Conclusion

① Non-local neural networks는 데이터 내 local한 부분이 아닌 전체 부분을 고려해서 학습할 수 있는 1x1 Convolutional Filter를 적극적으로 활용

② 1x1 Convolutional Filter를 활용해 효율적으로 네트워크 구축이 가능

③ Input Size와 Output Size가 같기 때문에 기존 모델에 쉽게 적용해 성능 향상을 도모할 수 있음

④ Self-Attention과 같은 구조를 지니고 있어 Visualization을 통해 해석이 가능함

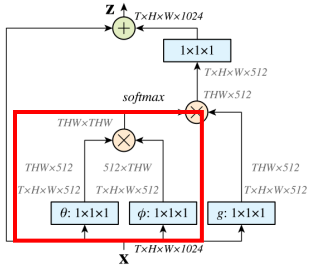
감사합니다.

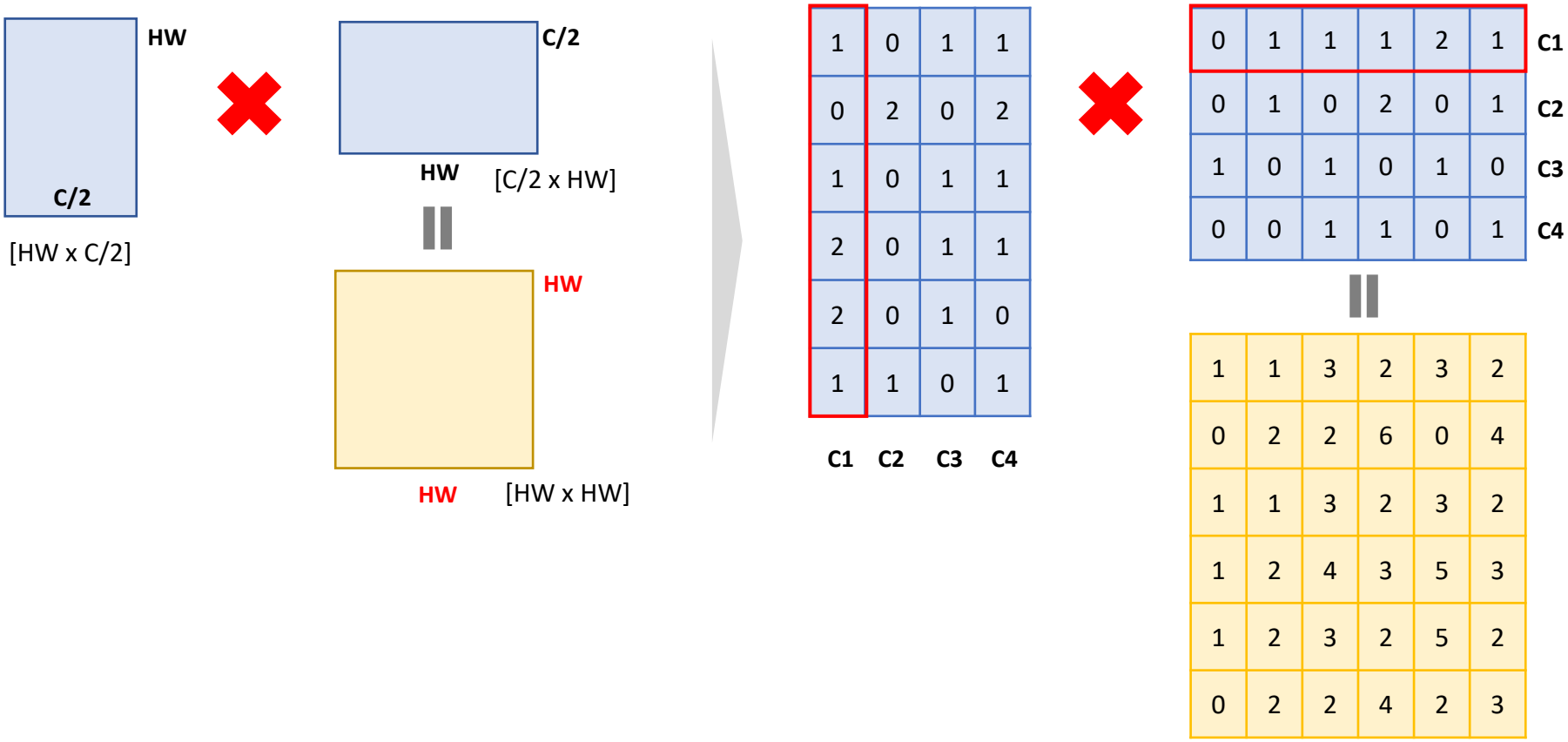
❖ 출처

- Wang, Xiaolong, et al. "Non-local neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- https://www.youtube.com/watch?v=6NN8kXh-Tpk
- HMG Journal, https://news.hmgjournal.com/Tech/deep-learning-carfuture
- https://alexisbcook.github.io/2017/global-average-pooling-layers-for-object-localization/
- Cheng, Chi-Tung, et al. "Application of a deep learning algorithm for detection and visualization of hip fractures on plain pelvic radiographs." European radiology (2019): 1-9.
- https://hwiyong.tistory.com/45
- https://blog.lunit.io/2018/01/19/non-local-neural-networks/
- https://www.youtube.com/watch?v=Dvi5_YC8Yts
- https://www.youtube.com/watch?v=vcp0XvDAX68 – Andrew Ng
- https://www.youtube.com/watch?v=C86ZXvgpejM – Andrew Ng

Appendix

Non-local neural networks (NLNN)





- HW x HW feature map은 1x1 filter로 요약 된 모든 pixel 간의 관계 고려함
- Channer1 (C1) 내 값들이 서로 다른 pixel 값과 연산 되고 있음을 확인

Non-local neural networks (NLNN)

❖ Non-local neural networks 결과

➢ Pixel간 관계를 찾기 위한 다양한 식

model, R50	top-1	top-5
C2D baseline	71.8	89.7
Gaussian	72.5	90.2
Gaussian, embed	72.7	**90.5**
dot-product	**72.9**	90.3
concatenation	72.8	**90.5**

(a) **Instantiations**: 1 non-local block of different types is added into the C2D baseline. All entries are with ResNet-50.

Gaussian. Following the non-local mean [4] and bilateral filters [47], a natural choice of f is the Gaussian function. In this paper we consider:

$$f(\mathbf{x}_i, \mathbf{x}_j) = e^{\mathbf{x}_i^T \mathbf{x}_j}. \tag{2}$$

Embedded Gaussian. A simple extension of the Gaussian function is to compute similarity in an embedding space. In this paper we consider:

$$f(\mathbf{x}_i, \mathbf{x}_j) = e^{\theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j)}. \tag{3}$$

Dot product. f can be defined as a dot-product similarity:

$$f(\mathbf{x}_i, \mathbf{x}_j) = \theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j). \tag{4}$$

Concatenation. Concatenation is used by the pairwise function in Relation Networks [40] for visual reasoning. We also evaluate a concatenation form of f:

$$f(\mathbf{x}_i, \mathbf{x}_j) = \text{ReLU}(\mathbf{w}_f^T [\theta(\mathbf{x}_i),\ \phi(\mathbf{x}_j)]). \tag{5}$$

출처: Wang, Xiaolong, et al. "Non-local neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Non-local neural networks (NLNN)

Non Local Neural Network 핵심 컨셉

- ❖ 기존 CNN 모델에 특정 부분에 Non Local Block을 넣음
- ❖ Non Local Block을 통해 Feature내 모든 값을 고려한 Weight 계산

1	0	2	3
0	-1	0	0
0	1	2	0
0	0	2	1

*

1	0	2
0	-2	0
0	-1	1

2	0	2	3
1	0	0	0
0	1	2	0
0	0	2	1

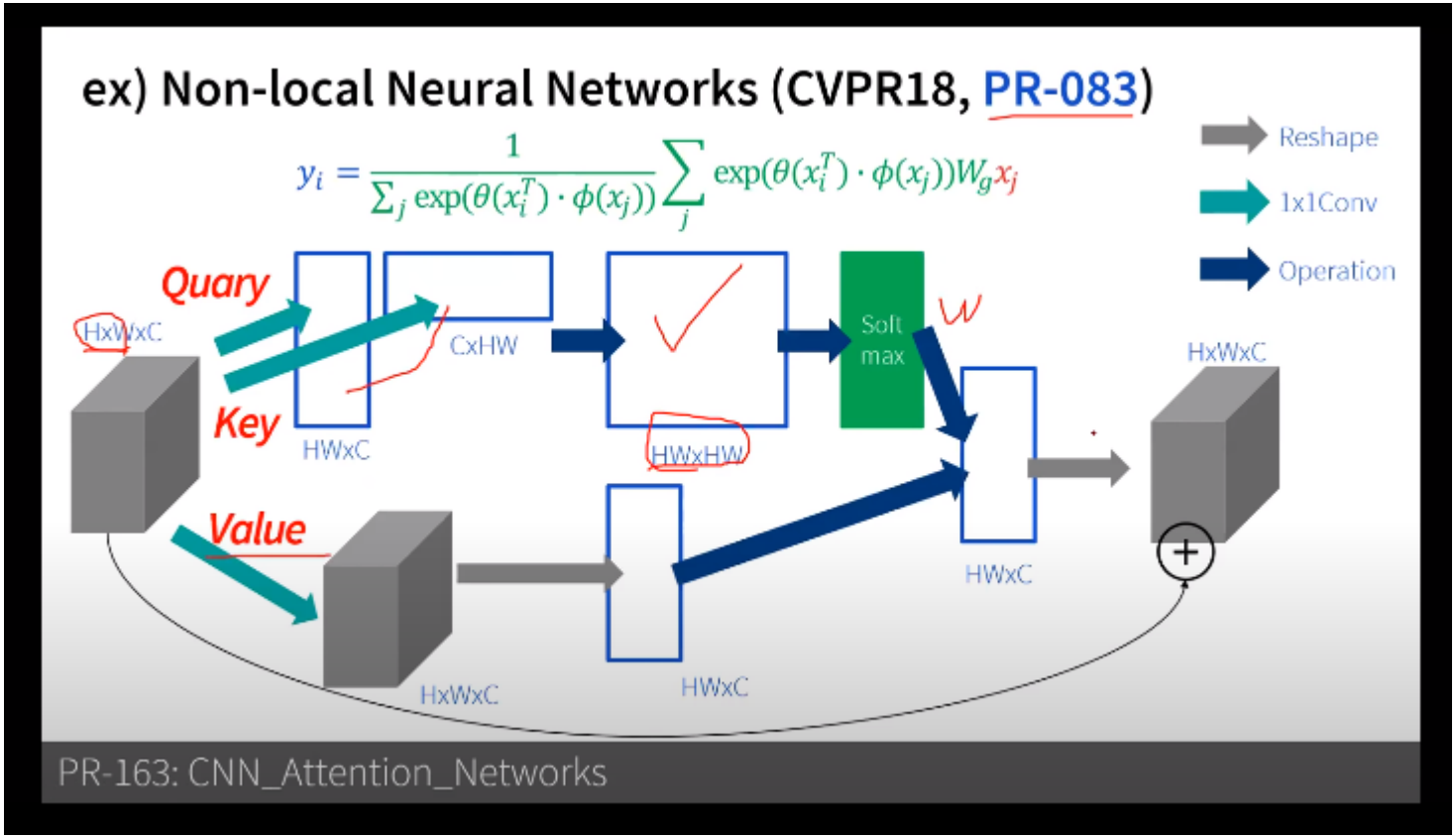
*

2	0	4	6
2	0	0	0
0	2	4	0
0	0	4	2

Feature 내 모든 값에 대한 가중치 고려

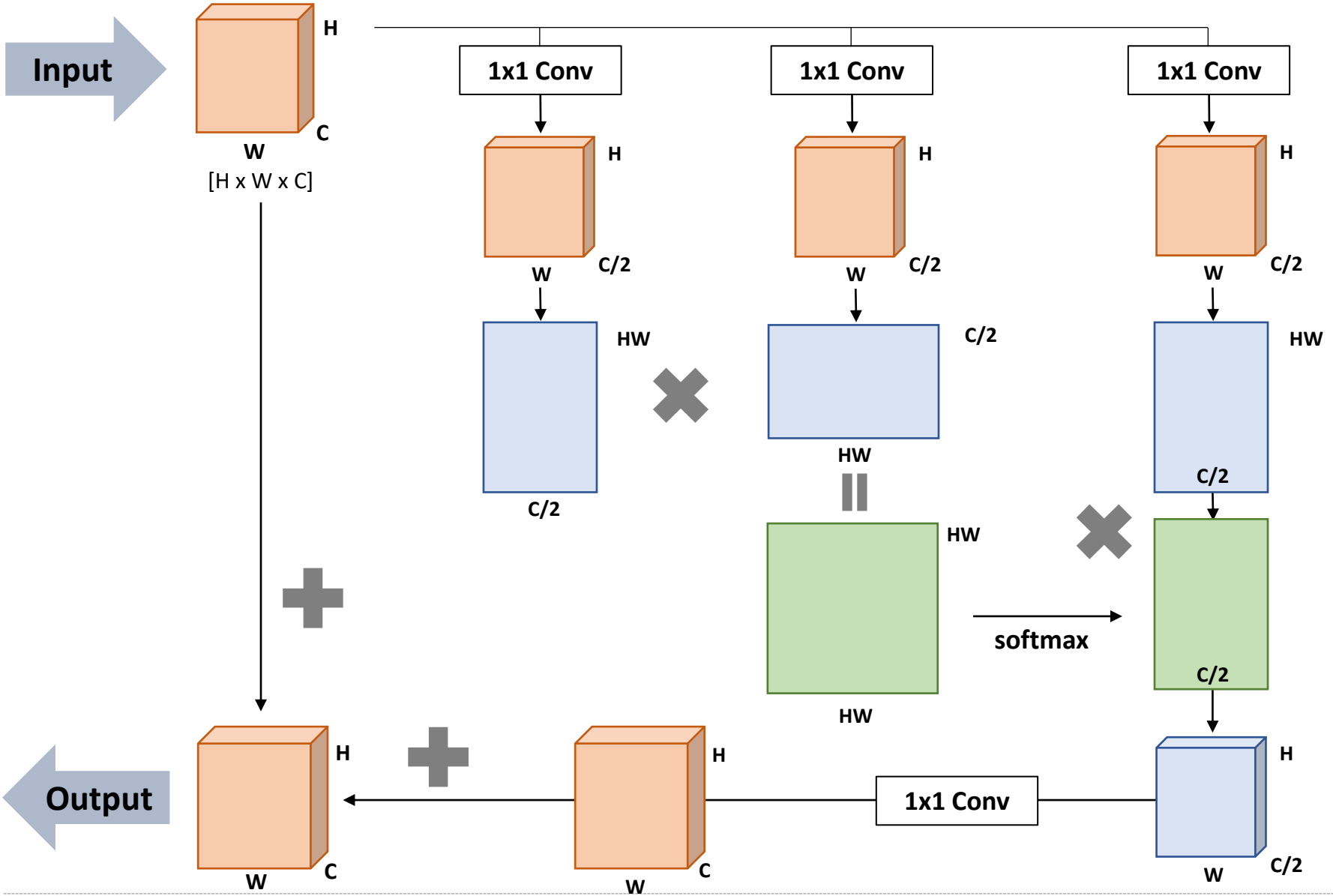
Non-local neural networks (NLNN)

❖ Non-local neural networks (NLNN)CNN 구조



출처: https://www.youtube.com/watch?v=Dvi5_YC8Yts

Non-local neural networks (NLNN)



Non-local neural networks – 실제 적용 사례

❖ 프로토스가 초반 포토 (공격 건물) 러쉬로 승리를 잡은 경기

Non-local neural networks – 실제 적용 사례

- Non-local neural network 모델 내 Attention score 결과
- 프로토스가 공격하는 지역(우측 상단)과 본진(좌측 하단)의 Attention score가 높음을 확인할 수 있음

<원본 미니맵>

<Attention>