

[CNN RNN]

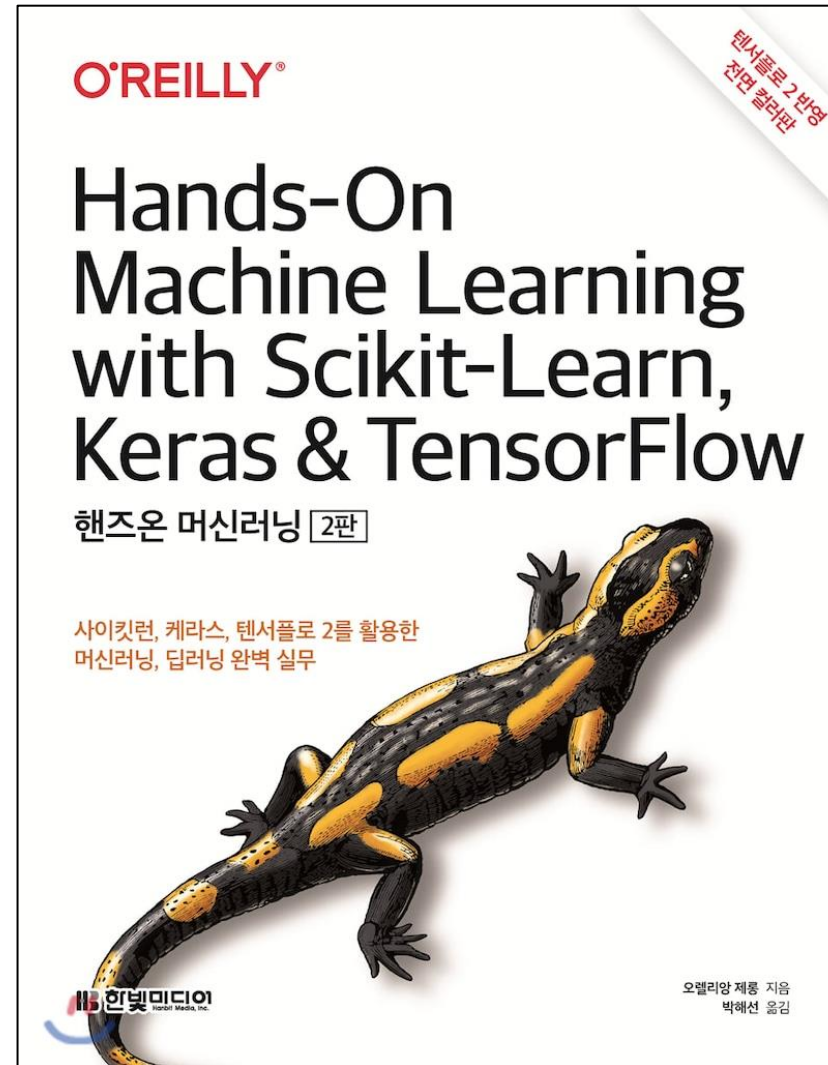
17기 사이언스 김찬

1. 개요

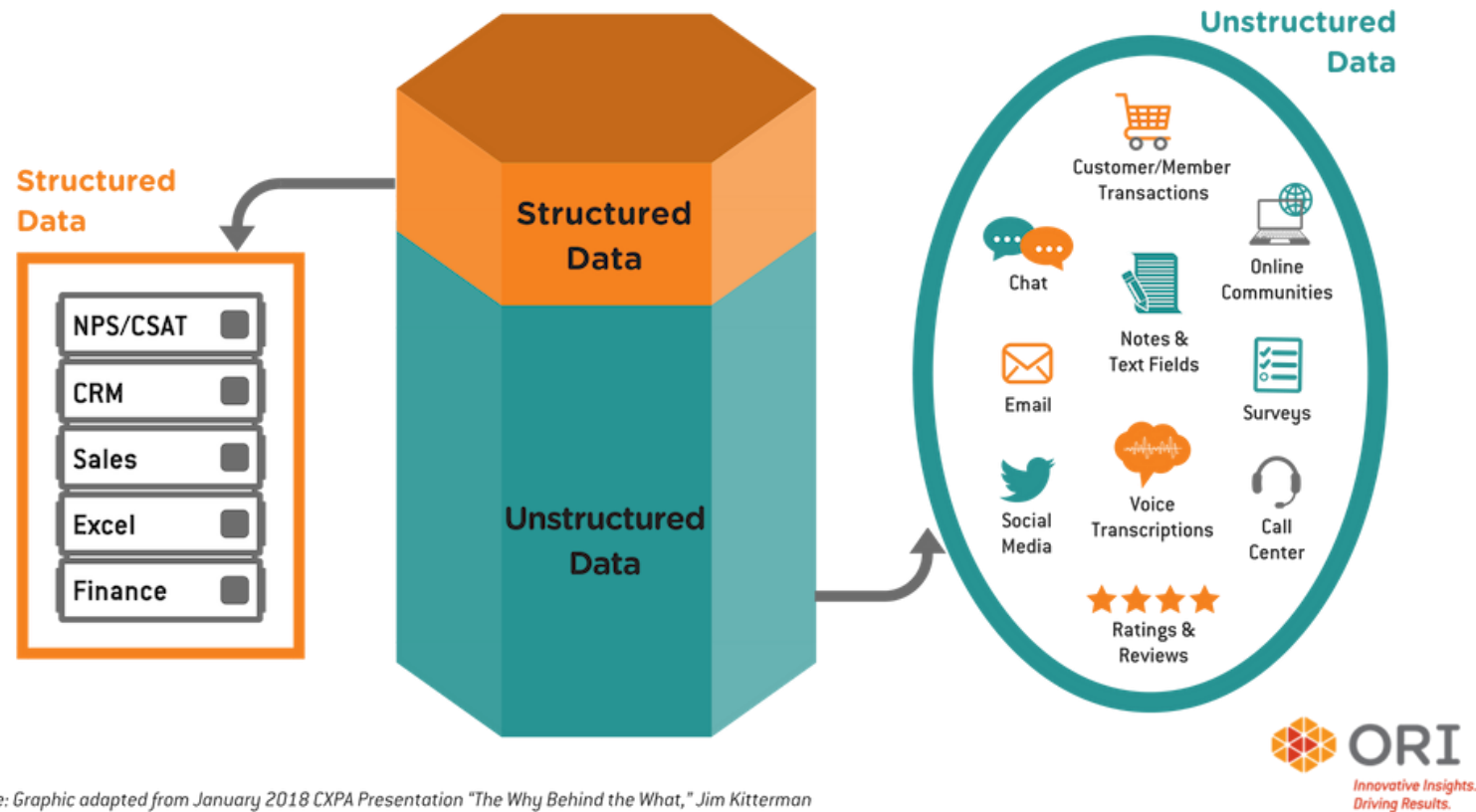
2. CNN

3. RNN

출처



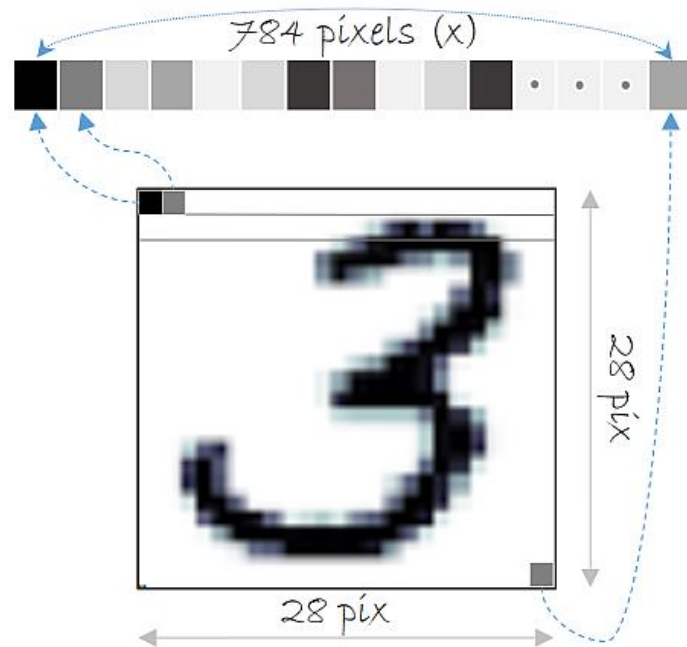
What's Hiding in Your Unstructured Data?



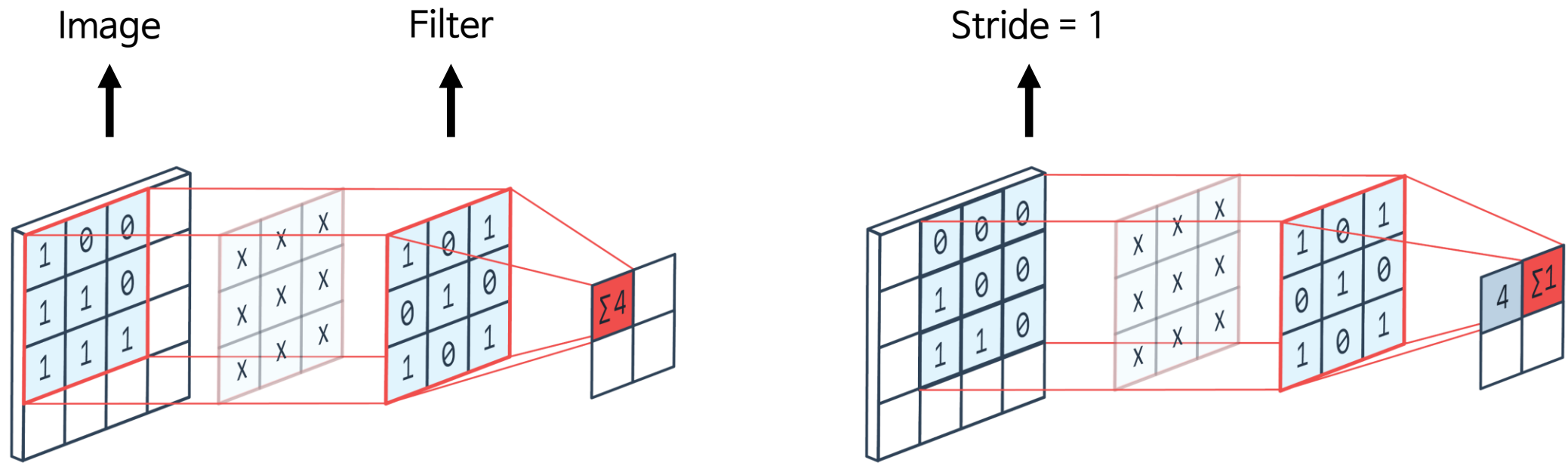
비정형 데이터 중 이미지 - CNN
시계열 - RNN 에 대해 알아보시다!

CNN

Convolutional Neural Network, 합성곱 신경망
이미지 처리에 굉장히 많이 사용
이미지 데이터의 공간 정보를 유지



단순한 FC layer를 가지고 처리하면 공간적 특성 무시
이를 유지하는 상태로 처리하는 방법 필요!



Input : 4 x 4
Output : 2 x 2
Loss!

Padding : output의 size 조절을 위해 채워넣는 값

0	0	0	0	0	0	0	0
0	3	3	4	4	7	0	0
0	9	7	6	5	8	2	0
0	6	5	5	6	9	2	0
0	7	1	3	2	7	8	0
0	0	3	7	1	8	3	0
0	4	0	4	3	2	2	0
0	0	0	0	0	0	0	0

$6 \times 6 \rightarrow 8 \times 8$

*

1	0	-1
1	0	-1
1	0	-1

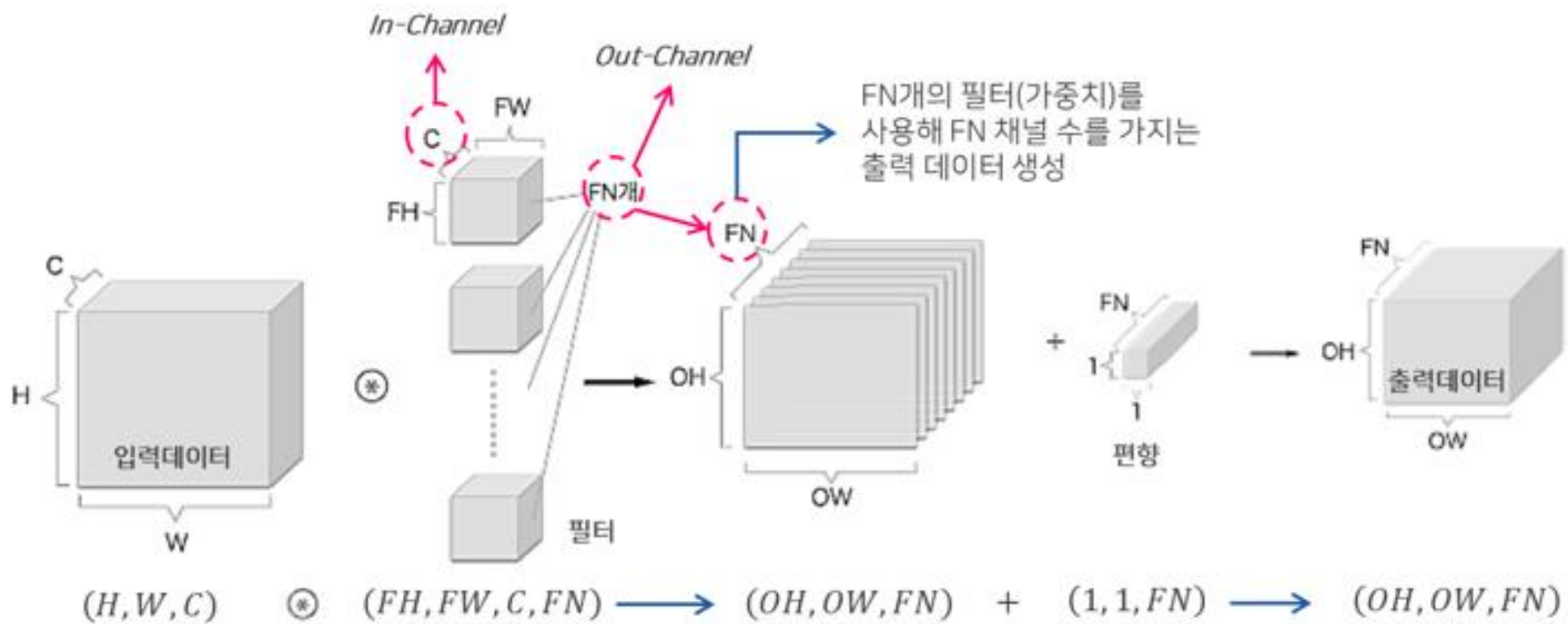
3×3

=

-10	-13	1			
-9	3	0			

6×6

<https://cs231n.github.io/assets/conv-demo/index.html>



Pooling

Convolution layer를 거쳐 나온 결과값들의 대표값 추출
최댓값을 가져오는 max pooling, 평균을 가져오는 average pooling이 있음

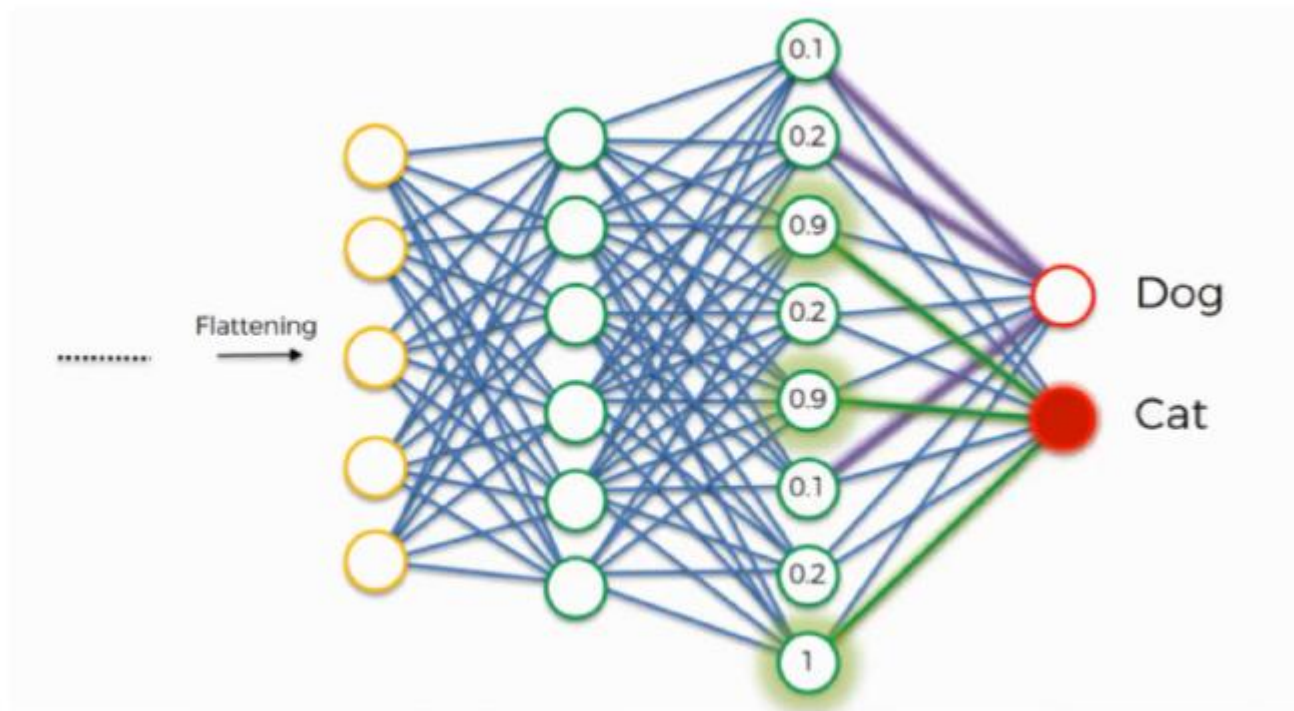
5	7	2	6
1	9	3	1
2	4	2	0
0	3	8	5

2x2 max pooling
stride 2



9	6
4	8

일반적인 flatten layer를 넣어서 최종 결과값을 구하는 layer
CNN 구조의 맨 마지막에 위치

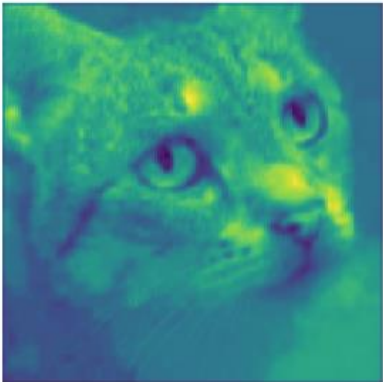


Image를 입력하면 전체를 보고 판단하는 것이 아니라, 특정 부분의 특징을 살펴봄

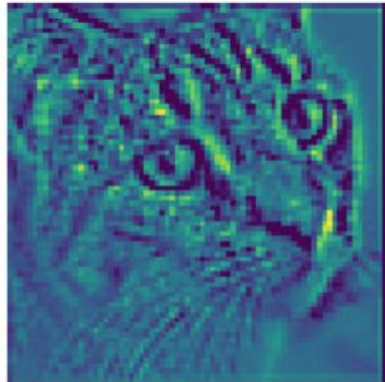
이렇게 작은 특징들을 수치화 하여 최종 확률을 계산

CNN layer를 통해 컴퓨터가 알 수 있도록 반복 연산

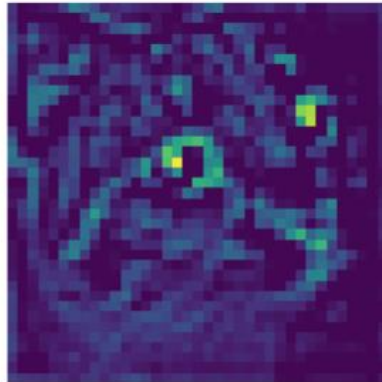
block1_conv1



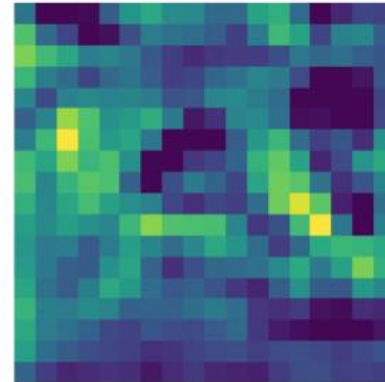
block2_conv1



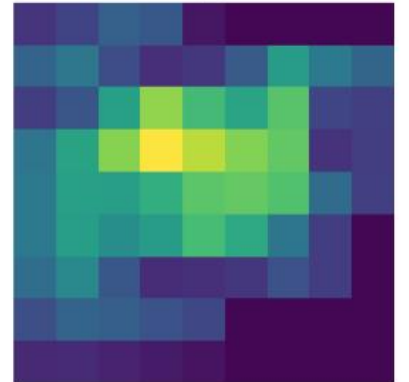
block3_conv1



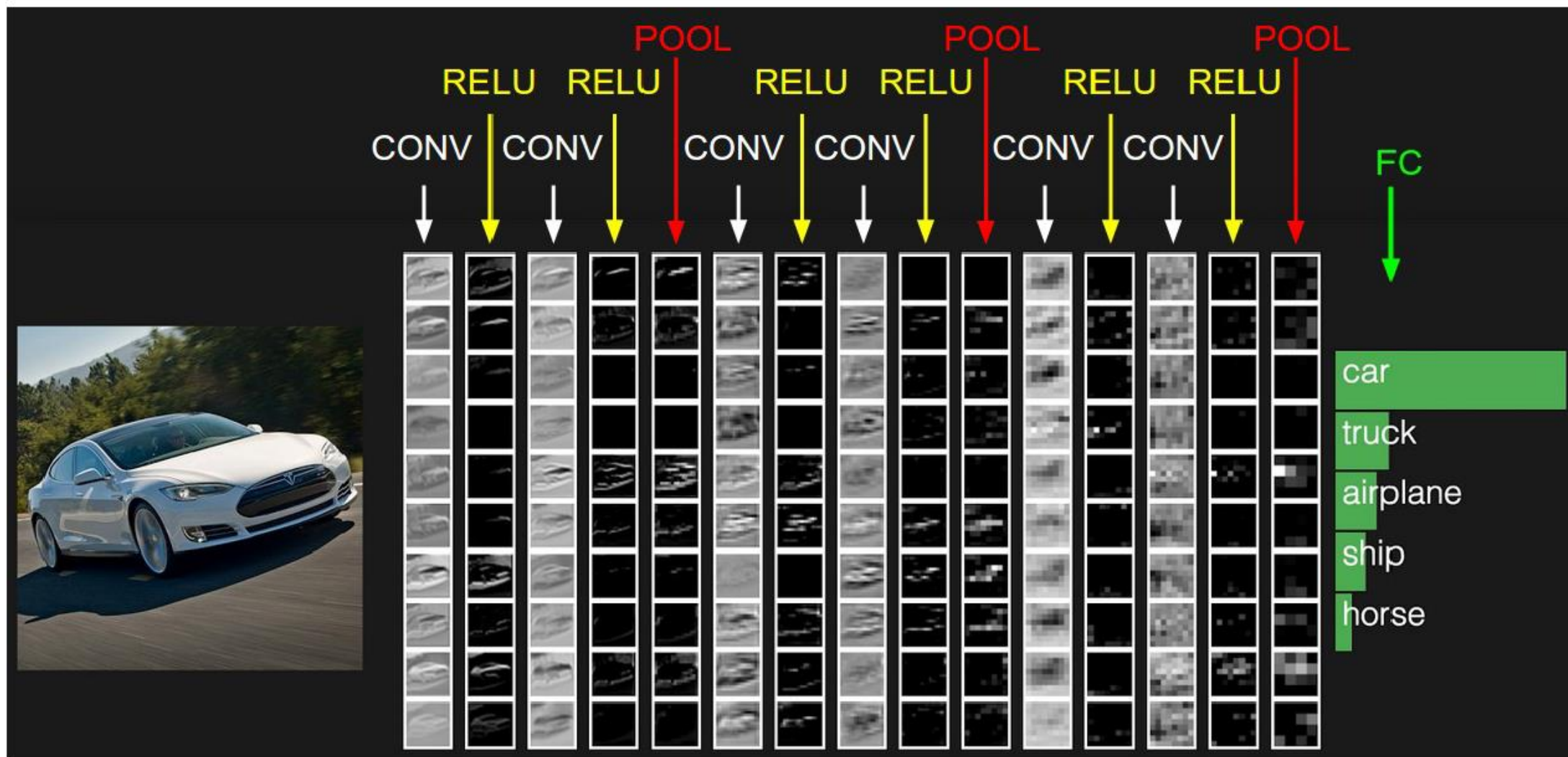
block4_conv1



block5_conv1

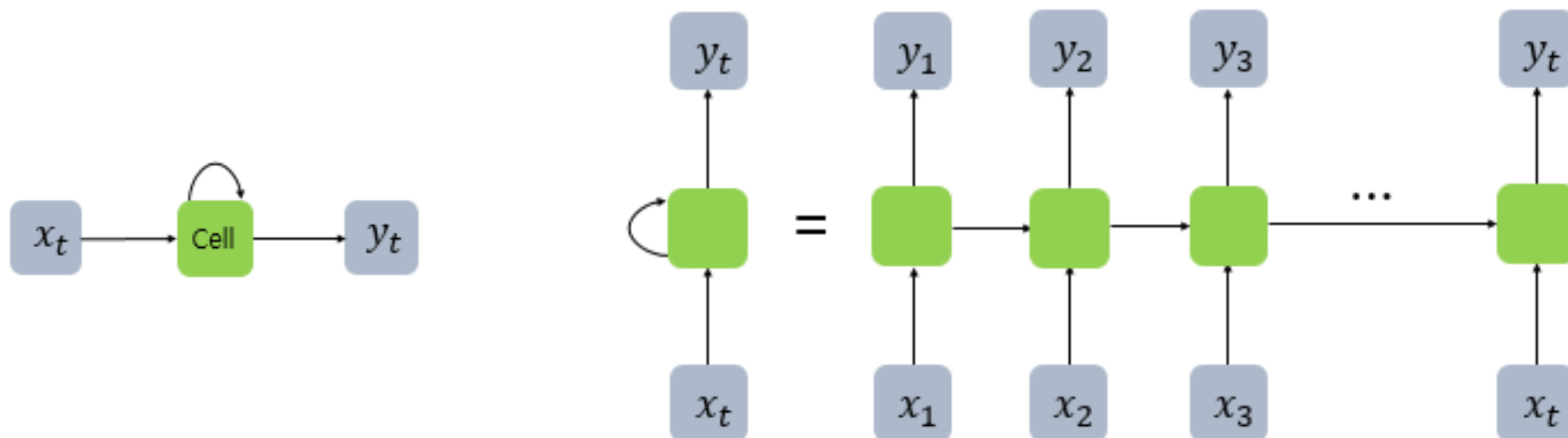


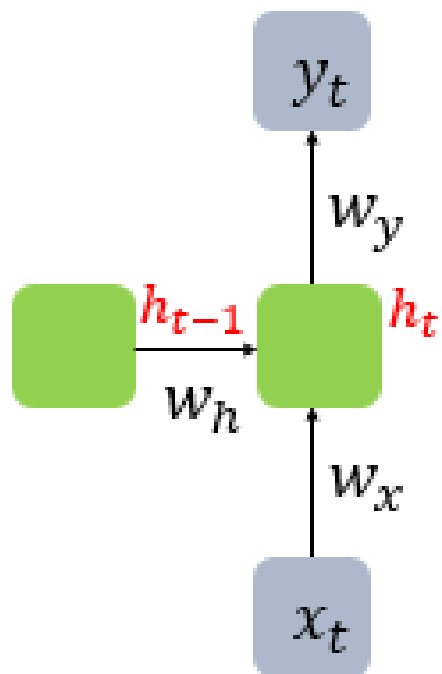
(Convolution → Activation function → pooling) 의 반복!



RNN

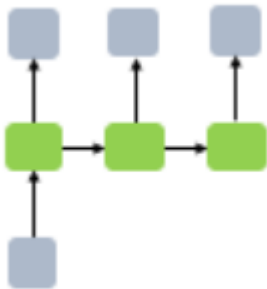
Recurrent Neural Network, 순환 신경망
입력과 출력을 시퀀스 (임의의 길이를 가진 데이터) 형태
시계열 데이터나 자연어 처리에서 매우 유용



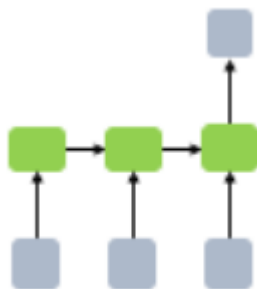


은닉층 : $h_t = \tanh(W_x x_t + W_h h_{t-1} + b)$

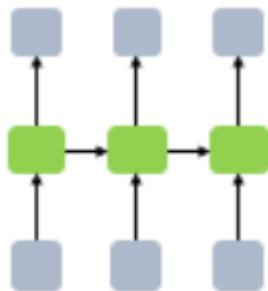
출력층 : $y_t = f(W_y h_t + b)$



벡터 투 시퀀스 (예 : 이미지를 입력하면 이미지에 대한 캡션 출력)



시퀀스 두 벡터 (예 : 영화 리뷰 입력하면 감성 점수 출력)



시퀀스 투 시퀀스 (예 : 주식가격 예측)