

Agenda

1. Limitation of DNN

2D 데이터를 1D로 평탄화하여 DNN 적용할 수 있지만

위치에 상관없이 동일한 수준의 중요도를 갖는다(6과 9)

1. 전체 글자가 조금만 이동하면 새로운 학습데이터로 처리해야 한다

Translation Invariance 특성 보장X

2. 글자 크기가 변하거나, 회전, 변형하면 좋은 결과를 기대하기 어렵다

Scale / Rotation / Distortion Invariance 특성 보장X

- ① 학습 시간(training time): 크기, 회전, 이동 등 모든 데이터를 학습해야 하고
입력영상이 커지고, Layer가 깊어지면
- ② DNN 모델의 크기(network size): 모델의 크기가 커지고
- ③ 변수의 개수(number of parameters) 파라미터의 수도 많아진다

2. 합성곱신경망(CNN. Convolutional Neural Networks)

Kernel(filter, weight)

Receptive field

Stride

Padding

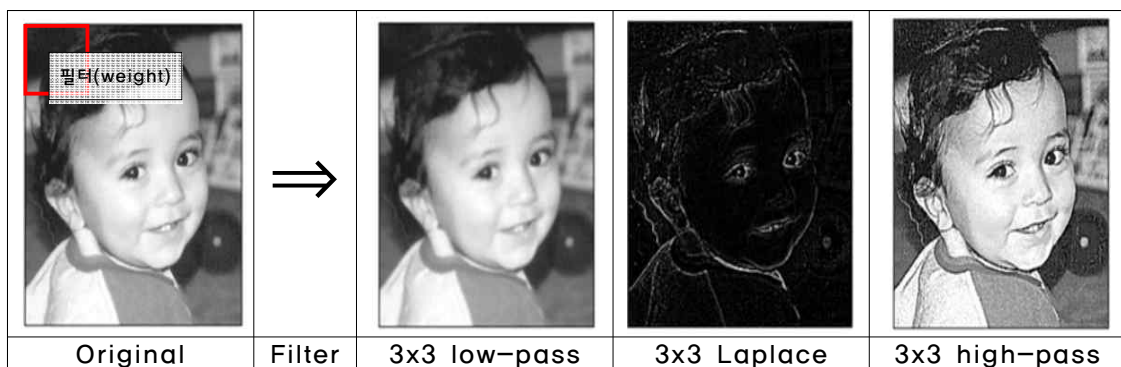
Pooling: Max / Average Pooling

합성곱신경망(CNN. Convolutional Neural Network)

변환 불변성에 기초하여 이미지를 분석에 사용하는 깊은 인공신경망의 한 종류

필터 역할을 스스로 학습하여 상대적으로 전처리를 거의 사용하지 않는다

Convolution: filter 연산에서 사용되어 영상에서 feature를 추출할 때 사용

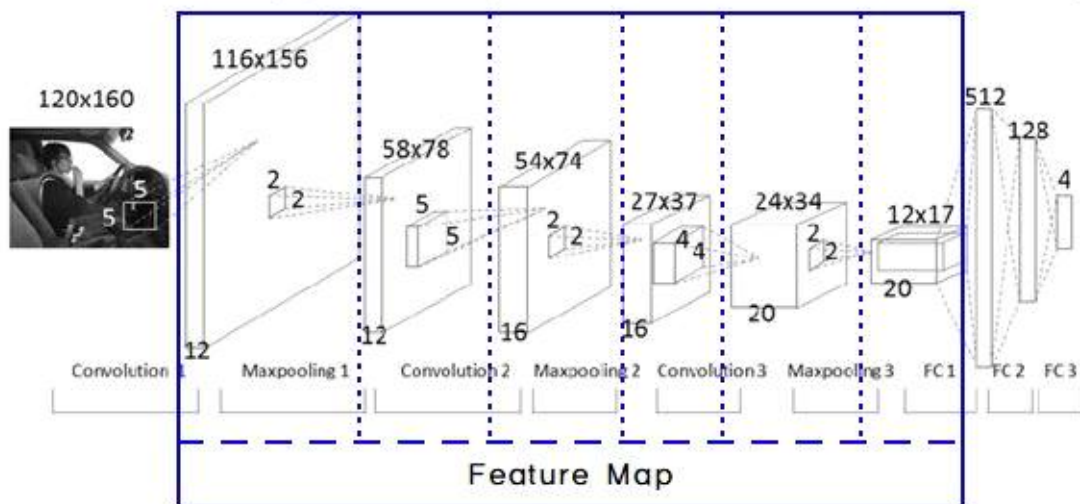


Convolution의 과정	<table><tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table> * <table><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr></table> (Filter, Kernel, Weight)	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1		
1	1	1	0	0																																	
0	1	1	1	0																																	
0	0	1	1	1																																	
0	0	1	1	0																																	
0	1	1	0	0																																	
1	0	1																																			
0	1	0																																			
1	0	1																																			
	Image		Convolved Feature																																		
	<table><tr><td>1 x1</td><td>1 x0</td><td>1 x1</td><td>0</td><td>0</td></tr><tr><td>0 x0</td><td>1 x1</td><td>1 x0</td><td>1</td><td>0</td></tr><tr><td>0 x1</td><td>0 x0</td><td>1 x1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	1 x1	1 x0	1 x1	0	0	0 x0	1 x1	1 x0	1	0	0 x1	0 x0	1 x1	1	1	0	0	1	1	0	0	1	1	0	0	⇒	<table><tr><td>4</td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	4								
1 x1	1 x0	1 x1	0	0																																	
0 x0	1 x1	1 x0	1	0																																	
0 x1	0 x0	1 x1	1	1																																	
0	0	1	1	0																																	
0	1	1	0	0																																	
4																																					
	<table><tr><td>1</td><td>1 x1</td><td>1 x0</td><td>0 x1</td><td>0</td></tr><tr><td>0</td><td>1 x0</td><td>1 x1</td><td>1 x0</td><td>0</td></tr><tr><td>0</td><td>0 x1</td><td>1 x0</td><td>1 x1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table>	1	1 x1	1 x0	0 x1	0	0	1 x0	1 x1	1 x0	0	0	0 x1	1 x0	1 x1	1	0	0	1	1	0	0	1	1	0	0	⇒	<table><tr><td>4</td><td>3</td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	4	3							
1	1 x1	1 x0	0 x1	0																																	
0	1 x0	1 x1	1 x0	0																																	
0	0 x1	1 x0	1 x1	1																																	
0	0	1	1	0																																	
0	1	1	0	0																																	
4	3																																				
	⋮																																				
	Green: 영상 이미지 Yellow: Convolution이 일어나는 영역 Red: Convolution Kernel																																				

Convolution + NN : Convolution을 사용하는 신경망 연산

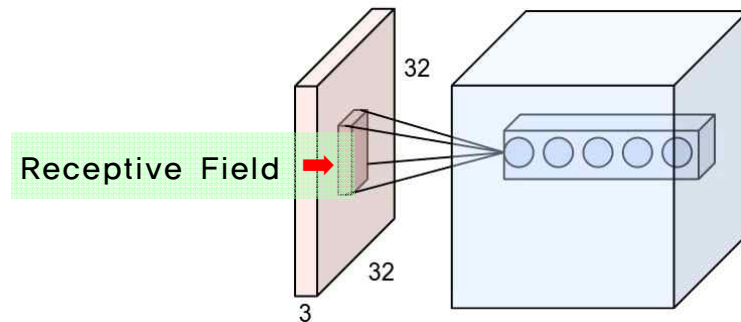
2개 이상의 CNN layer

& 입력 영상뿐만 아니라 중간 Feature map에도 Convolution 적용



Receptive Field(수용 영역)


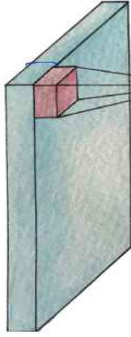

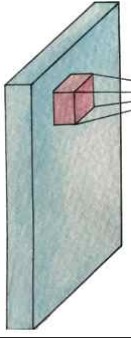
- : 출력 layer의 뉴런 하나에 영향을 미치는 입력 뉴런들의 공간 크기
- 외부 자극이 일부 영역에만 영향을 미친다(전체에 영향 X)
- 영상에서 특정 위치에 있는 픽셀들은 주변에 있는 일부 픽셀들과만 correlation이 높다 → 거리가 멀수록 영향이 감소한다

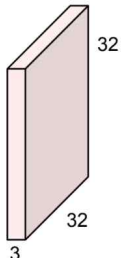
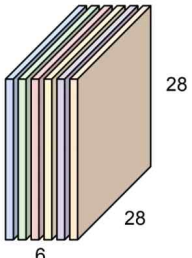


CNN의 장점

- 영상을 2D에서 1D로 평탄화하지 않으므로, 형상을 유지한다
- ∴ 입출력 모두 3차원 데이터로 처리하기 때문에 공간적 정보를 유지할 수 있다

DNN	CNN
<p>input layer hidden layer 1 hidden layer 2 output layer</p>	<p>depth height width</p>
<p>input 1 3072 Wx 10 x 3072 weights activation 1 10</p> <p>1 number: the result of taking a dot product between a row of W and the input (a 3072-dimensional dot product)</p>	<p>32x32x3 image 5x5x3 filter w activation map 28 28 1</p> <p>1 number: the result of taking a dot product between the filter and a small 5x5x3 chunk of the image (i.e. $5 \times 5 \times 3 = 75$-dimensional dot product + bias)</p> <p>$w^T x + b$</p>
10x3072 weights	5x5x3 filter w
많은 수의 가중치	<p>적은 수의 가중치</p> <p>– 부분의 합으로 판단하는 브레인 모사</p>

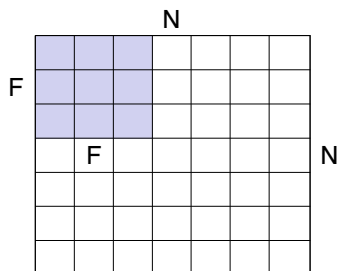
 <p>32x32x3 image</p>	<p>Image: width x height x depth depth = 1:1 채널 = 흑백 depth = 3:3 채널 = 칼라(RGB)</p>
 <p>32x32x3 image</p>	 <p>5x5x3 filter</p> <p>image * filter = $Wx + b$ = one number (W = filter, b = bias)</p>
<p>⋮</p>	
 <p>⋮</p>	

How many Numbers?	
<p>stride = 1,</p> 	<p>5x5x3 filters 6개</p> <p>activation maps</p>  <p>Convolution Layer</p>
<p>stride = 1, 32x32x3 image * 5x5x3 filters 6개 = 28x28x6 activation maps</p>	

Stride: Filter가 움직이는 정보

stride = n : n칸씩 움직인다

Output size 구하기



$$(N - F) / \text{stride} + 1$$

ex) $N = 7, F = 3$

stride = 1 : $(7-3)/1+1 = 5$ (ok)

stride = 2 : $(7-3)/2+1 = 3$ (ok)

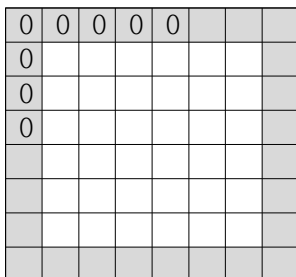
stride = 3 : $(7-3)/3+1 = 2.33$ (X)

stride = 4 : $(7-3)/4+1 = 2$ (ok)

Padding: 영상 사이즈 유지하기

ex) input 7x7 image

pad with 1 pixel border(회색)



if. pad의 값 = 0 : zero pad

⇐ zero pad with 1

Kernel에 따른 Padding의 크기

= $(F / 2)$ 의 소수점 첫째 자리 올림

ex)

$F = 3$: zero pad with 1

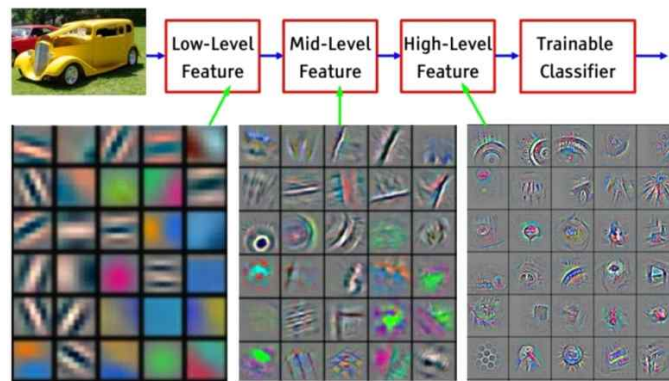
$F = 5$: zero pad with 2

$F = 7$: zero pad with 3

Convolution 결과의 Size 구하기

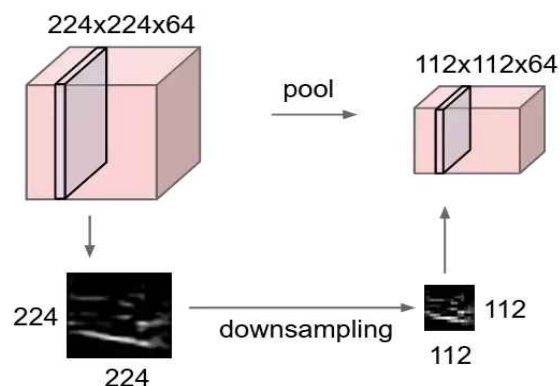
Input: $W_1 \times H_1 \times D_1$	Parameter Filter의 개수: K_1 Filter의 한 변 크기: F_1 Stride: F Zero Pad의 개수: S
Output: $W_2 \times H_2 \times D_2$	$W_2 = (W_1 - F + 2P) / S + 1$ $H_2 = (H_1 - F + 2P) / S + 1$ $D_2 = K$ → filter 개수로 output의 depth 정할 수 있다
with parameter sharing, $F \cdot F \cdot D_1$ weight per filter total $\Rightarrow (F \cdot F \cdot D_1) \cdot K$ weights and K biases	

1. Visualization of Activation Map(Feature Map)



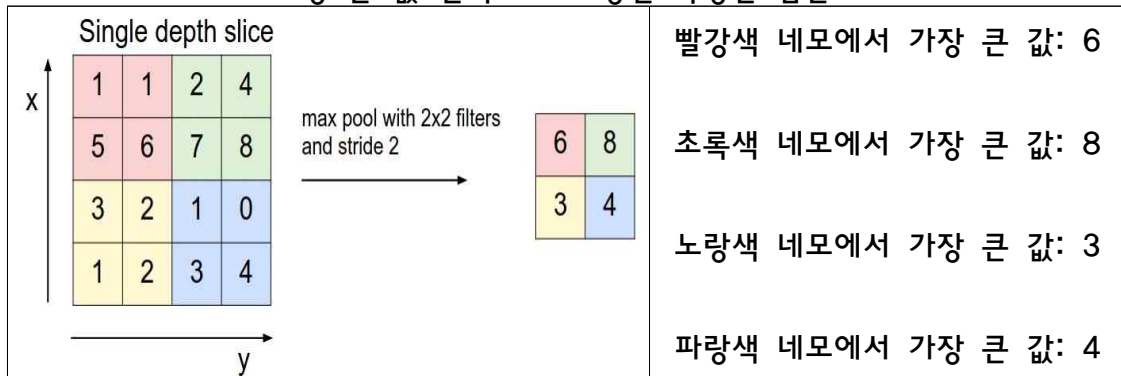
2. Pooling Layer(Sampling)

: resizing Conv layer



Max Pooling

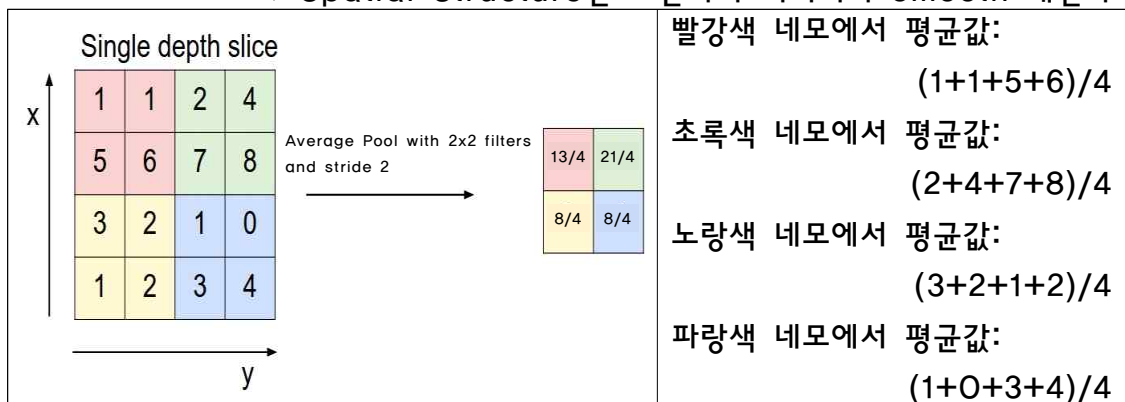
Filter 내에서 가장 큰 값 선택 \Rightarrow 더 강한 특징만 남는다



Average Pooling

Filter 내에서 평균값 선택

\Rightarrow Spatial Structure만 보존하여 이미지가 smooth 해진다



- (2x2 filter의 경우) 전체 데이터의 75%를 버리고 25%만 선택
 \rightarrow Computational Complexity 감소한다
- Depth를 줄이지 않고 Spatially하게만 줄인다(Height & Width)
 $32 \times 32 \times 3 \rightarrow 16 \times 16 \times 3$
- Q) Stride와 Pooling 모두 down-sampling인데 어느 것 사용?
 A) 최근 CNN 아키텍처는 stride를 사용하는 경우가 많다(stride 추천)

3. FC layer(Fully Connected Layer)

마지막 Pooling layer를 통과한 데이터가 1x1024 feature를 갖는다면
FC layer with $W=1024 \times 5$ 를 통과하여 1x5의 Output이 나온다

