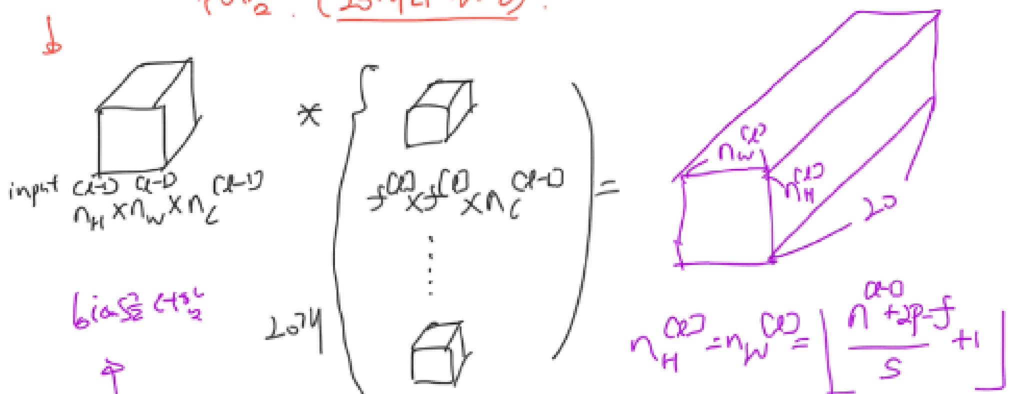
* CNN (Convolution Neural Network) → conv층 [l] layer에서
- $p^{[l]}$: l층 패딩
- $s^{[l]}$: l층 stride

input 이미지 : $n_W^{[l-1]} \times n_H^{[l-1]} \times n_C^{[l-1]}$ (이전층의 채널수)

filter는?
$f^{[l]}$: filter size.
filter의 채널수는 동일 $= n_C^{[l-1]}$

* filter의 개수는 [l]층의 값에서 주어짐. (20개라 가정).

output ⇒



bias를 더함

$$n_H^{[l]} = n_W^{[l]} = \left\lfloor \frac{n^{[l-1]} + 2p - f}{s} + 1 \right\rfloor$$

parameter수는? [l-1]층
$$\Rightarrow (f^{[l-1]} \times f^{[l-1]} \times n_C^{[l-2]} + 1) \times (\text{filter의 개수})$$

[l]층
$$= (f^{[l]} \times f^{[l]} \times n_C^{[l-1]} + 1) \times (\text{filter 개수})$$

↓ parameter수는 신경망층이 깊어질수록 증가.
이미지의 크기는 감소)

※ Pooling (풀링층)

* Pooling층 기능 : 표현의 크기를 줄임으로써 계산속도를 높임.
→ 특징 검출이 쉽다.

* Max pooling

1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

4×4

→

2×2

9	2
6	3

⇒ filter 2×2를 사용하고 stride=2 로 준 결과나옴일

$$\left\lfloor \frac{4-2}{2} + 1 \right\rfloor = 2$$

Max pooling을 통해 각 칸의 최대 크기의 수만 뽑아낸다.

maxpooling의 ← (f=2
s=1

Maxpooling 이유

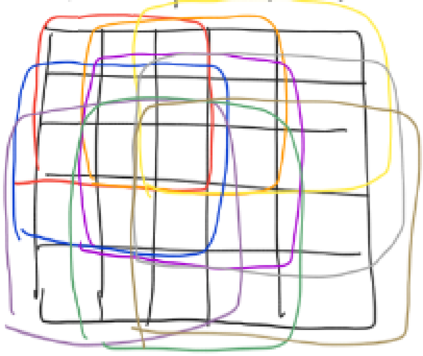
하이퍼파라미터

max pooling의

* f와 S는 상수(고정값)이라서 Gradient Descent 등의 학습을 시키지 어렵다.

⇒ pooling의 각 범위에서의 최대 값이 그 범위에서 가장 feature를 잘 나타내는 특성인 것.

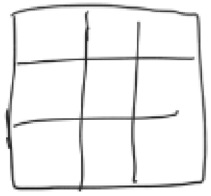
ex) Input: 5X5



Max pooling →

$f=3$
$S=1$

$\left\lfloor \frac{5-3}{1}+1 \right\rfloor = 3$

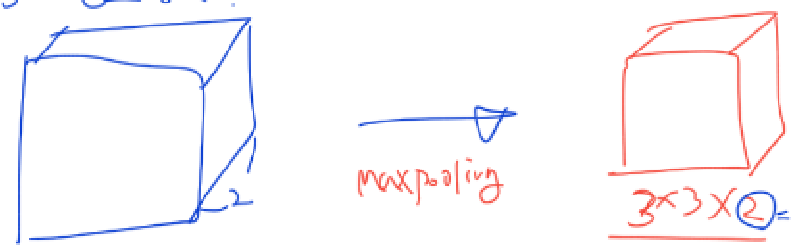


3x3

5X5 input에서 3x3 칸씩 잡아서 가장 큰 수를 뽑고 한칸씩 이동

2차원 Max pooling

3차원도 동일



$5 \times 5 \times 2$
$= n_c$

maxpooling

$3 \times 3 \times 2 = n_c$

채널의 수가 같다.
max pooling은 각 채널별로 적용.

* pooling 특징
input과 output의 n_c (채널수가 같다.)

* Average pooling은 각 filter 칸의 평균을 output에 저장

* Summary of pooling ⇒ 하이퍼파라미터들
f: filter size
S: stride
Max or average pooling

출력 계산식은 Conv와 동일

$\left\lfloor \frac{n+2p-f}{S}+1 \right\rfloor$

또한 pooling의 in/out img의 n_c는 동일..

* 대부분의 Max pooling에서는 padding을 사용하지 않는다.

$n_H \times n_W \times n_C \xrightarrow{\text{pooling}} \boxed{\left\lfloor \frac{n_H - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_W - f}{s} + 1 \right\rfloor \times n_C}$

∴ $p = 0$

※ 풀링의 유의할점은 학습하는 변수가 없다.

No parameters to learn

(※조절가능)

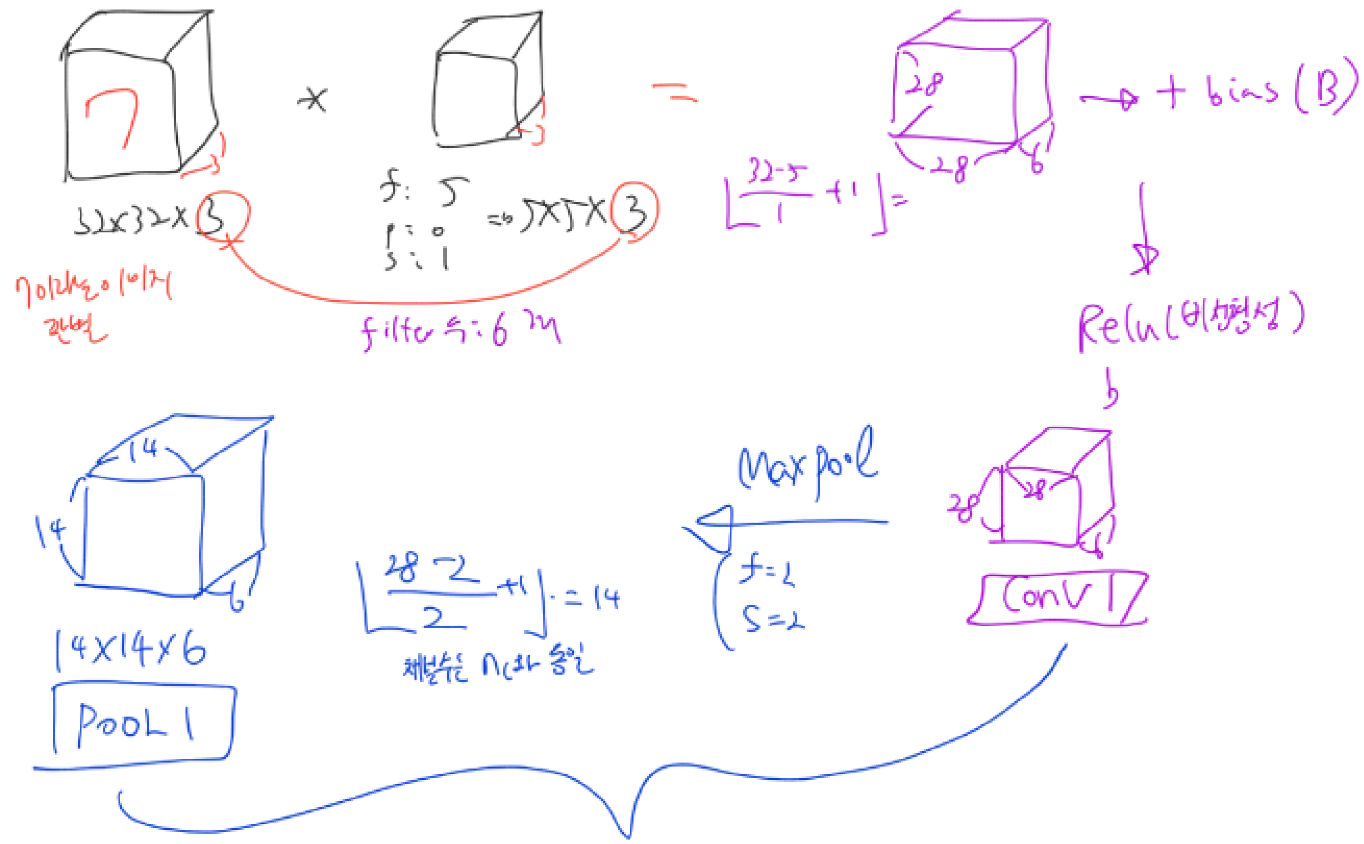
ㄴ> because f, S가 고정변수인 Hyper parameters이기 때문

* maxpooling 특징은

이미지 특징이 필터 한 부분에서 검출이 되면 높은수를 남기고,

그렇지 않으면 다른 최대값에 비해 상대적으로 작아짐 → 특징을 더 잘 남긴다.

※ CNN Example (예제 (Conv와 Pooling 층까지의 이해).



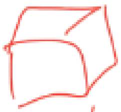
= Layer 1 (신경망에서

일반적인 신경망층을 셀 때는 W와 변수를 가지는 층만 계산

Pool층은 W가 없고 Hyper 변수만 가지므로

(Conv1, Pool)을 Layer 1으로 둔다.

Layer 1



14x14x6

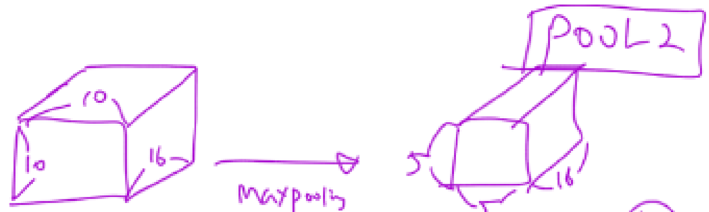
$f=5$

$s=1$

$p=0$

filter수=16

$\lfloor \frac{14-5}{1}+1 \rfloor = 10$



10x10x16

Conv2

$f=2$

$s=2$

$\lfloor \frac{10-2}{2}+1 \rfloor = 5$

5x5x16

n_c는 동일

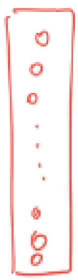
Conv2, POOL2 = Layer 2

→ 이제 이 5X5X16=400인 POOL2를 한줄로 펼쳐서

Layer2의

5X5X16=

400



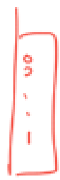
400X1의 벡터로 만든다.

(여러개 뉴런처럼)

→ 이 400개 뉴런으로 120개 뉴런을 가진 층을 만들어보자

400개 뉴런이

120개 뉴런과 fully connect



120

FC3

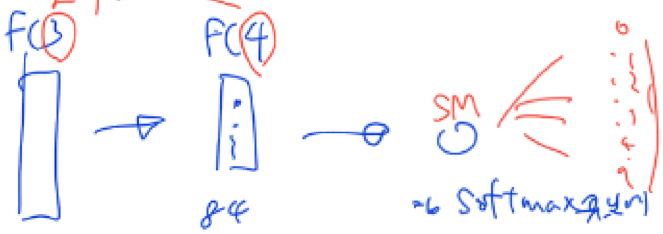
fully connected

$W^{[3]}$ (120, 400)의 크기를 가지고 완전연결

400개 뉴런이 각각 120개 뉴런과 연결되어있다.

$b^{[3]} = (120)$

Layer



10개의 출력이 나오게 된다.

120

84

→ softmax 유닛에

적용가능한 84개의 열벡터가 생성

120개를 84개로 fully connected

7이라는 숫자를 0 ~ 9 중에 찾기위함이므로

Layer more deep

↓

n_H, n_W는 decrease ↓.

n_c 는 increase ↑.

일반적 CNN (신경망)패턴 : CONN → POOL → CONV → POOL → FC → FC → Softmax

detail :

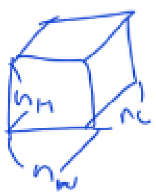
	Activation shape	Activ size	parameter 개수
input	(32, 32, 3)	$a^{[0]}$ 32×32×3 = 3072	0
CONV1 (f=5, s=1)	(28, 28, 8)	$a^{[1]}$ 6272	208 (5×5×8 + 1×8)
POOL1	(14, 14, 8)	$a^{[1]}$ 1568	0
CONV2 (f=5, s=1)	(10, 10, 16)	$a^{[2]}$ 1600	400 = (5×5×16 + 1×16)
POOL2	(5, 5, 16)	400	0
FC3	(120, 1)	120	120×400 + 1 (bias) = 48001
FC4	(84, 1)	84	84×120 + 1 = 10081
Softmax	(10, 1)	10	84×10 + 1 = 841

$\underbrace{(10, 1)}_{0 \sim 9 : (숫자범위이므로}$

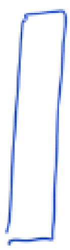
* POOL층은 parameter가 따로 없다 (f, s 등의 하이퍼 parameter는 존재)
대부분의 parameter들은 FC층에 존재
활성값 size도 층 깊어질수록 감소.

신경망 : (CONV, POOL, FC) 후에 softmax.

fully-Connected층으로 갈 때



$n_H \times n_w \times n_c$의 act size를 한줄로펴서



$[n_H \times n_w \times n_c, 1]$의

벡터로 변환

W역시 행렬

ex) (400,1)에 W(120,400)

↓ FC3

(120,1)로 줄이고

W(84,120)

84로 더줄여서

Soft Max

output

10

마지막 수정: 오후 7:25