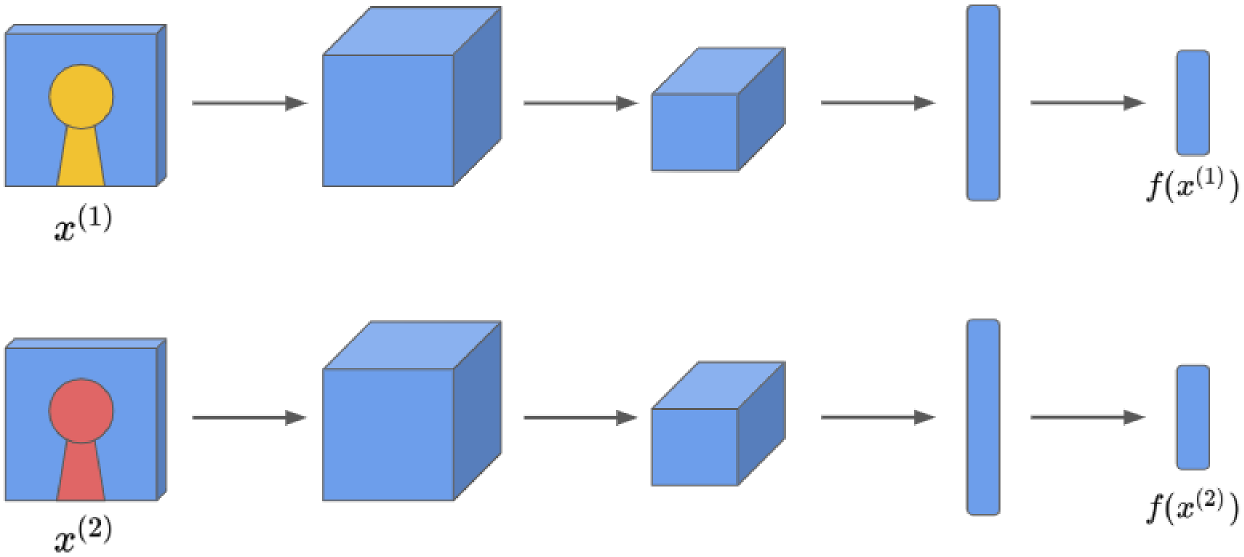
딥러닝 스터디
C4 W4 L04 ~ L11

이승목

2020.10.20.

목차

얼굴 인식 – 샴 네트워크



$$d(x^{(1)}, x^{(2)}) = ||f(x^{(1)}) - f(x^{(2)})||_2^2$$

얼굴인식 – Triplet loss

Anchor

A

Positive

P

Anchor

A

Negative

N

$d(A,P) = 0.5 \rightarrow 0.2$ $\qquad$ $d(A,N) = \cancel{0.51}\ 0.7$

Want: $\underbrace{\| f(A) - f(P) \|^2}_{d(A,P)} + \alpha \le \underbrace{\| f(A) - f(N) \|^2}_{d(A,N)}$

$\underbrace{\| f(A) - f(P) \|^2}_{0} - \underbrace{\| f(A) - f(N) \|^2}_{0} + \underbrace{\alpha}_{\text{margin}} \le 0$ $\qquad$ $f(img) = \vec{0}$

[Schroff et al.,2015, FaceNet: A unified embedding for face recognition and clustering]

Andrew Ng

얼굴인식 – Triplet loss

Given 3 images A, P, N:

$$\mathcal{L}(A,P,N) = \max\left(\boxed{\|f(A)-f(P)\|^2 - \|f(A)-f(N)\|^2 + \alpha}, 0\right)$$

$\leq 0 \quad > 0$

$$J = \sum_{i=1}^{m} \mathcal{L}(A^{(i)}, P^{(i)}, N^{(i)})$$

A, P

Training set: 10k pictures of 1k persons

[Schroff et al.,2015, FaceNet: A unified embedding for face recognition and clustering]

얼굴인식 – Triplet loss choosing data

During training, if A,P,N are chosen randomly,
$d(A, P) + \alpha \leq d(A, N)$ is easily satisfied.

$\|f(A)-f(P)\|^2 + \alpha \leq \|f(A)-f(N)\|^2$

Choose triplets that're "hard" to train on.

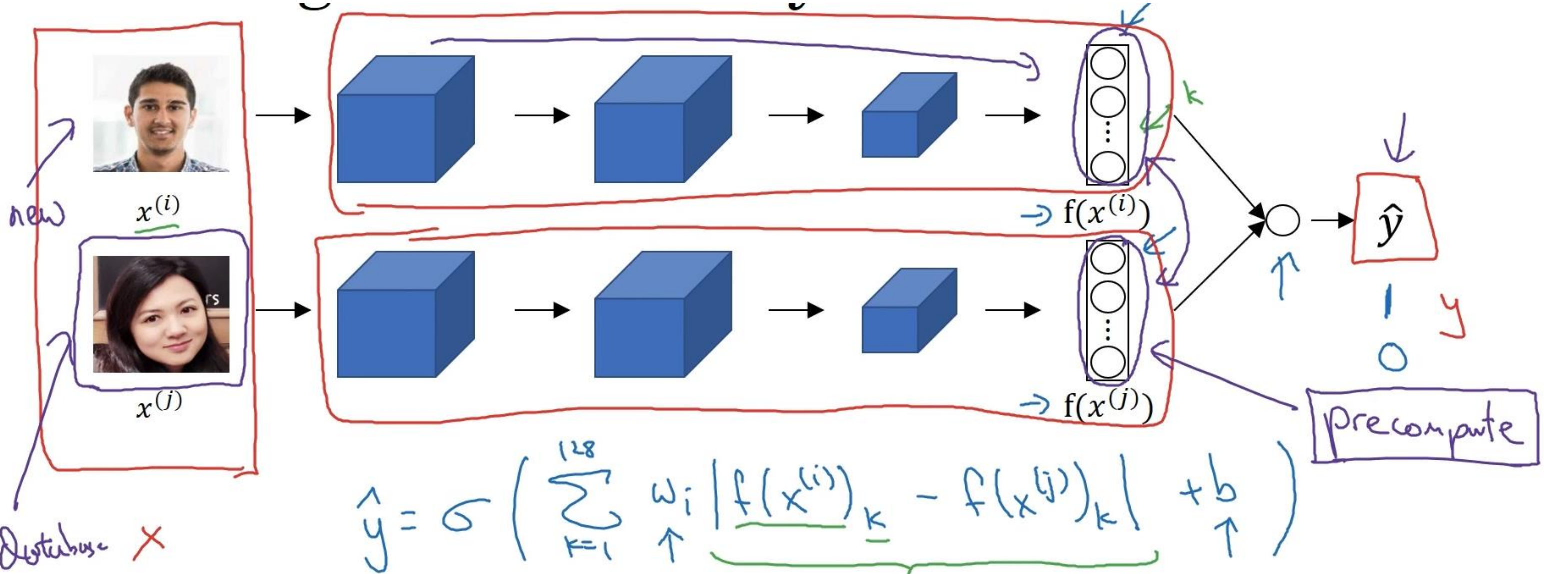
$d(A,P) + \alpha \leq d(A,N)$

$\underline{d(A,P)} \approx \underline{d(A,N)}$

$\downarrow \qquad \uparrow$

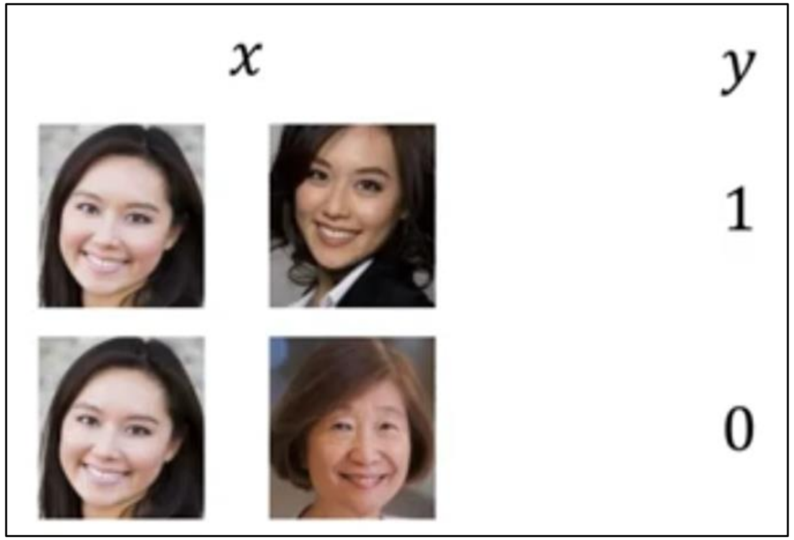
[Schroff et al.,2015, FaceNet: A unified embedding for face recognition and clustering]

얼굴인식 – 지도학습으로 얼굴인식



$$\hat{y} = \sigma\left(\sum_{k=1}^{128} w_i \left| f(x^{(i)})_k - f(x^{(j)})_k \right| + b\right)$$

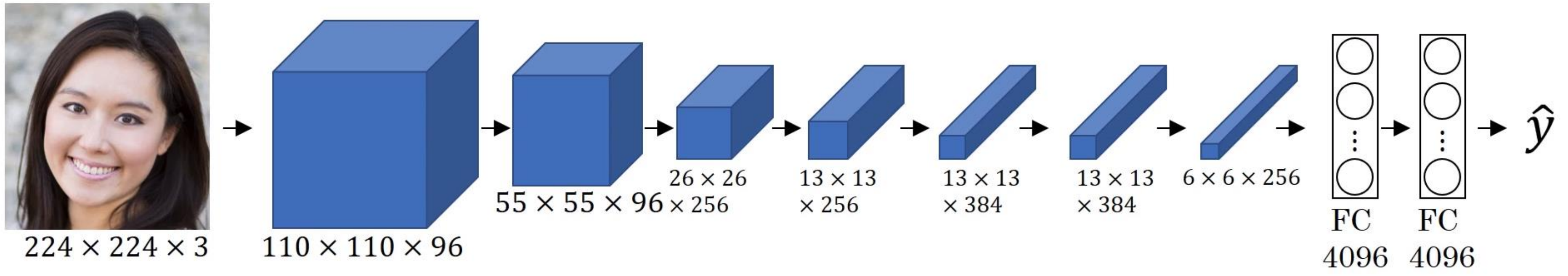
$$\frac{\left(f(x^{(i)})_k - f(x^{(j)})_k\right)^2}{f(x^{(i)})_k + f(x^{(j)})_k} \qquad \chi^2$$



[Taigman et. al., 2014. DeepFace closing the gap to human level performance]

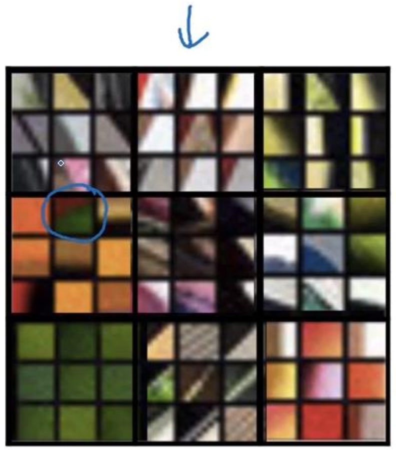
Andrew

CNN이 하는 일 – 시각화



Pick a unit in layer 1. Find the nine image patches that maximize the unit's activation.

Repeat for other units.



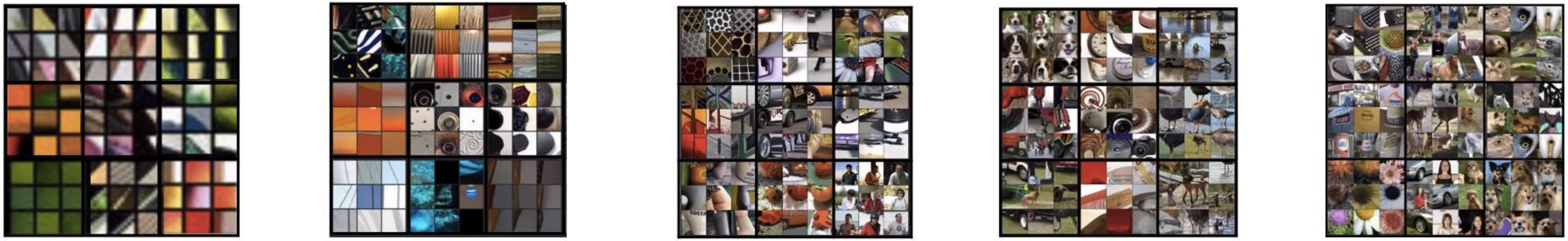
Visualizing and understanding convolutional networks라는

[Zeiler and Fergus., 2013, Visualizing and understanding convolutional networks]

Andrew Ng

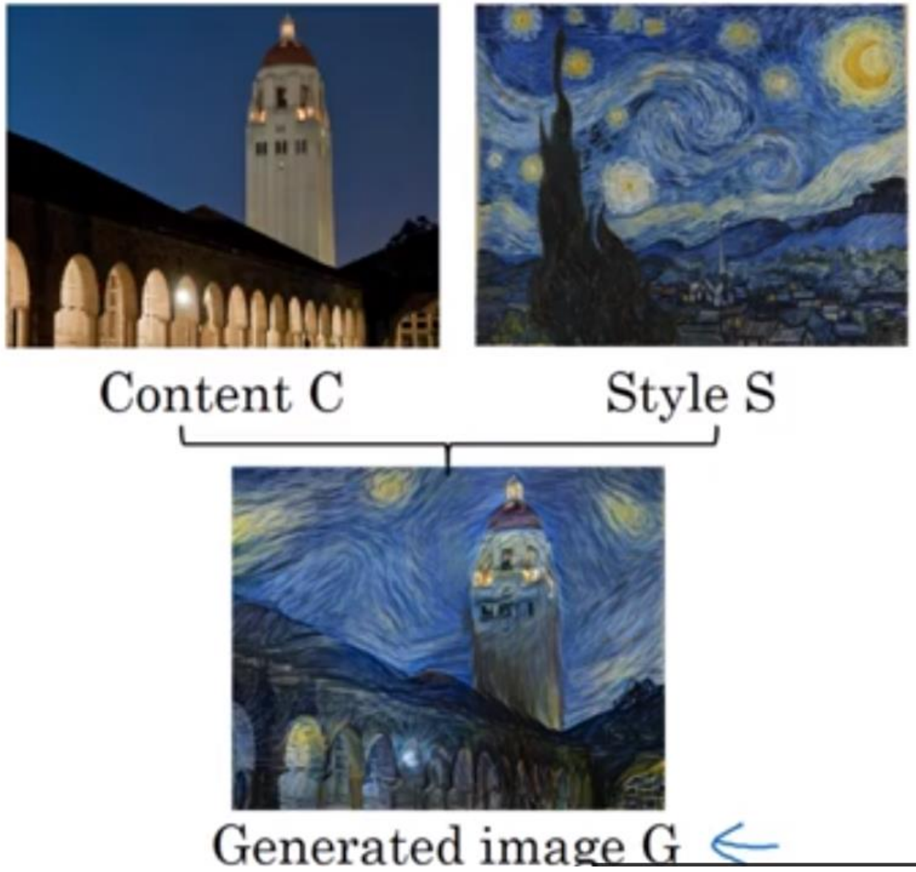
CNN이 하는 일 – 시각화

- 한 유닛은 비슷한 종류의 이미지에 대해 잘 활성화 된다.
- 높은 층일 수록 더 복잡한 feature를 잡아낸다.



Layer 1 Layer 2 Layer 3 Layer 4 Layer 5

스타일 전이



$$J(G) = \alpha J_{content}(C, G) + \beta J_{style}(S, G)$$

스타일 전이

1. Initiate G randomly

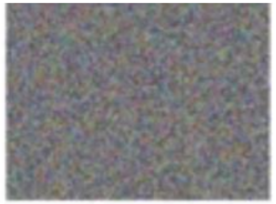
   G: 100 × 100 × 3

   RGB

2. Use gradient descent to minimize $J(G)$

$$G := G - \frac{\partial}{\partial G} J(G)$$


Initial



First iter.

Second iter.

Final output

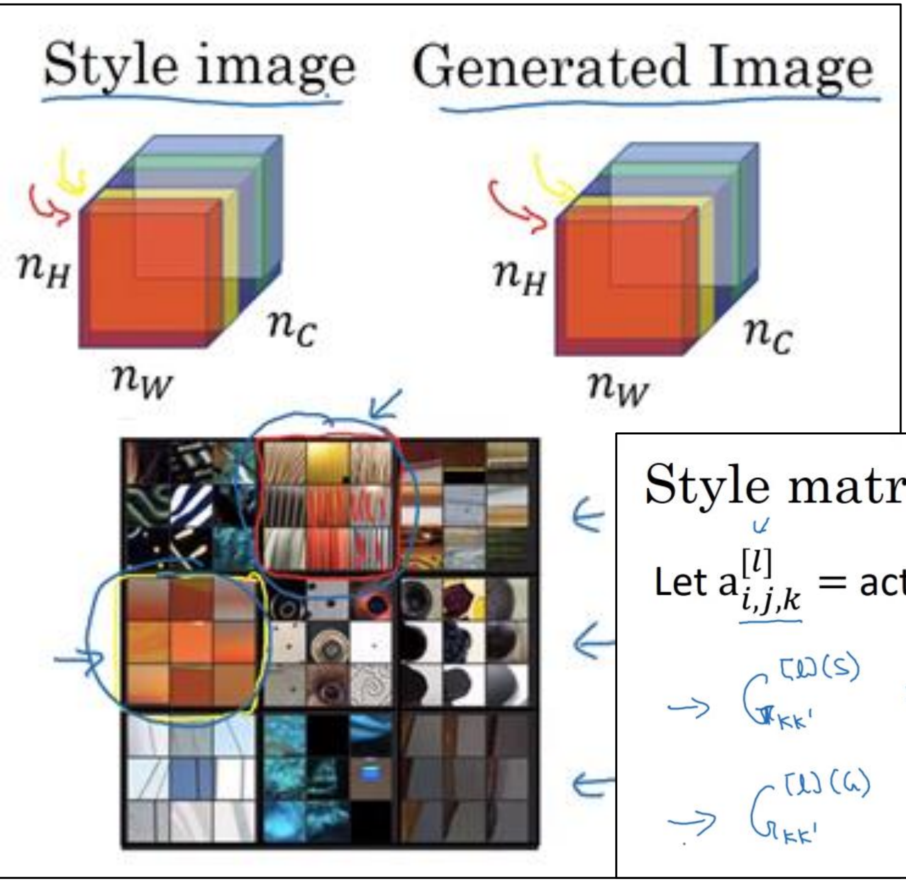
스타일 전이 – content cost

$$J(G) = \alpha\, J_{content}(C, G) + \beta\, J_{style}(S, G)$$

- Say you use hidden layer l to compute content cost.
- Use pre-trained ConvNet. (E.g., VGG network)
- Let $a^{[l](C)}$ and $a^{[l](G)}$ be the activation of layer l on the images
- If $a^{[l](C)}$ and $a^{[l](G)}$ are similar, both images have similar content

$$J_{content}(C, G) = \frac{1}{2}\underbrace{\| a^{[l](C)} - a^{[l](G)} \|^2}$$

스타일 전이 – style cost



$$J^{[l]}_{style}(S,G) = \frac{1}{b}||G^{[l](S)} - G^{[l](G)}||^2_F$$
$$= \frac{1}{(2n_H^{[l]}n_W^{[l]}n_C^{[l]})^2}\sum_k\sum_{k'}(G^{[l](S)}_{kk'} - G^{[l](G)}_{kk'})^2$$

$$J_{style}(S,G) = \sum_l \lambda^{[l]} J^{[l]}_{style}(S,G)$$

Style matrix

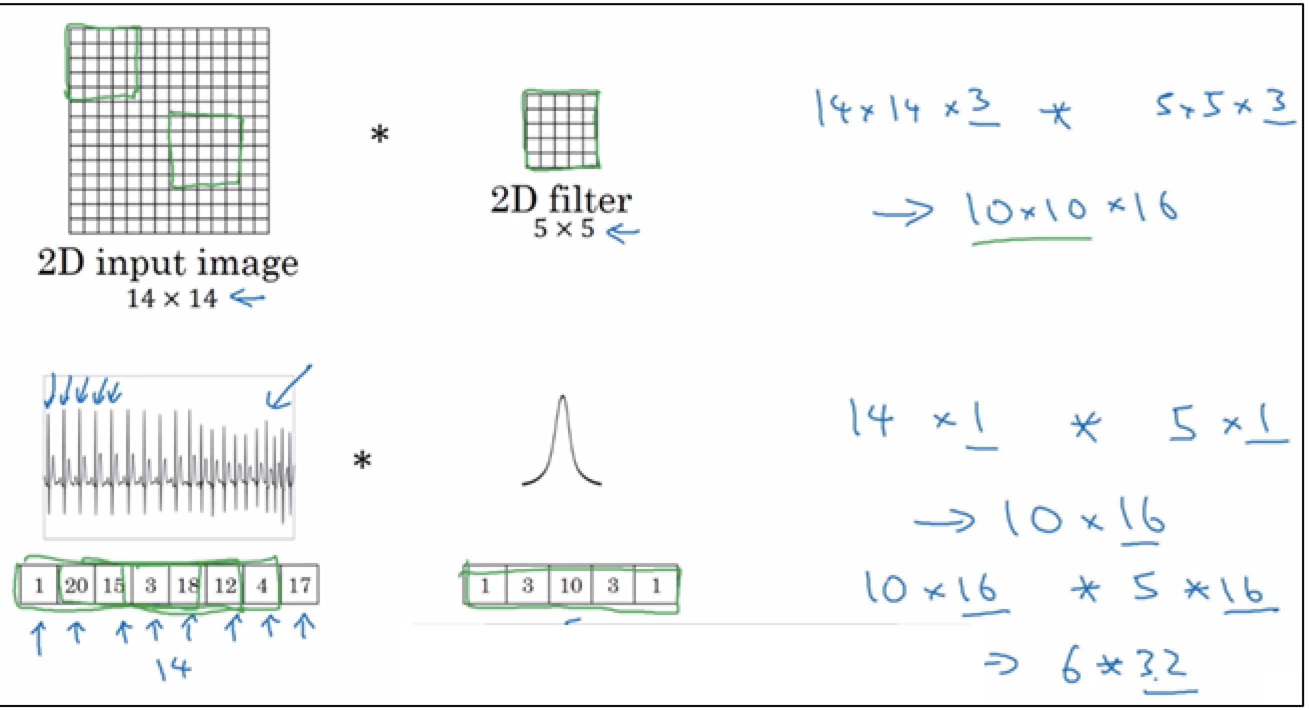
H W C

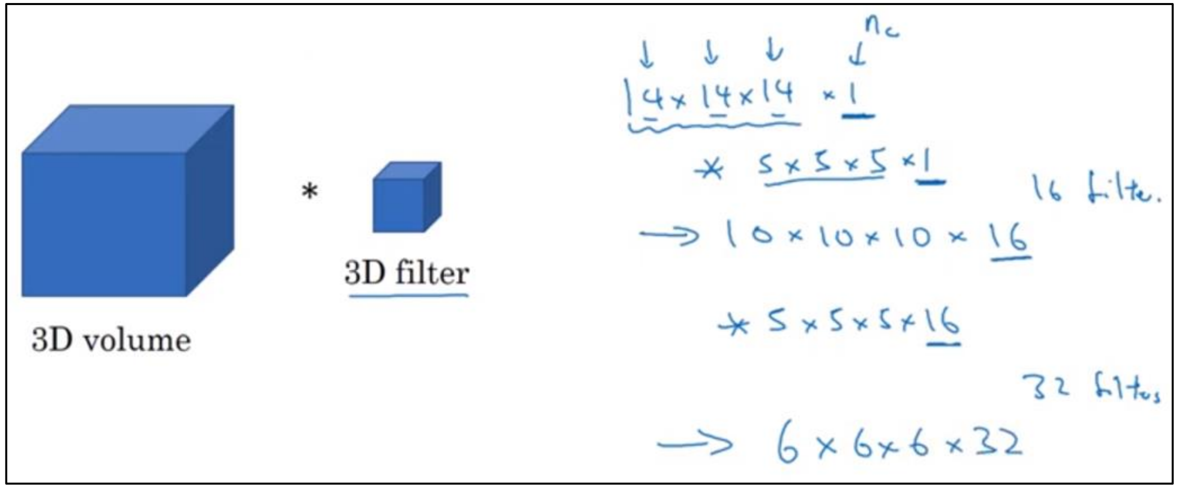
Let $a^{[l]}_{i,j,k}$ = activation at (i,j,k). $G^{[l](s)}$ is $n_c^{[l]} \times n_c^{[l]}$

$$G^{[l](S)}_{kk'} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a^{[l](S)}_{ijk}\, a^{[l](S)}_{ijk'}$$

$$G^{[l](G)}_{kk'} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a^{[l](G)}_{ijk}\, a^{[l](G)}_{ijk'}$$

1D와 3D에서 CNN


2D input image
14 × 14
2D filter
5 × 5
14×14×3 * 5×5×3
→ 10×10×16
14×1 * 5×1
→ 10×16
10×16 * 5×16
→ 6×32
1 20 15 3 18 12 4 17
14
1 3 10 3 1


3D volume
3D filter
n_c
14×14×14×1
* 5×5×5×1
16 filters
→ 10×10×10×16
* 5×5×5×16
32 filters
→ 6×6×6×32