# Convolutional Neural Networks

## Why convolutions?

deeplearning.ai

# Why convolutions

main advantages [ parameter sharing
               [ sparsity of connection
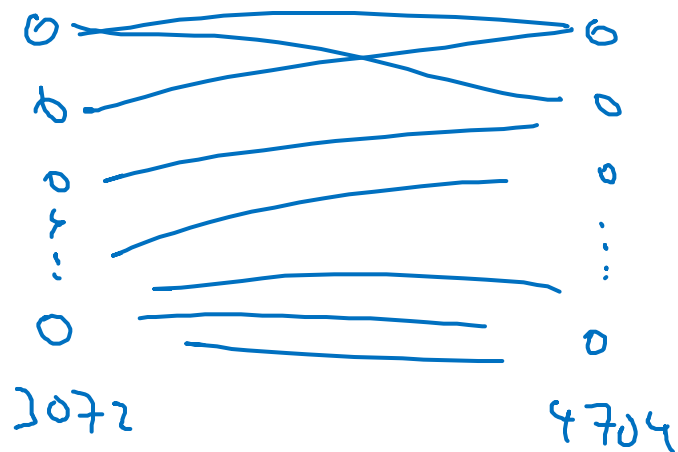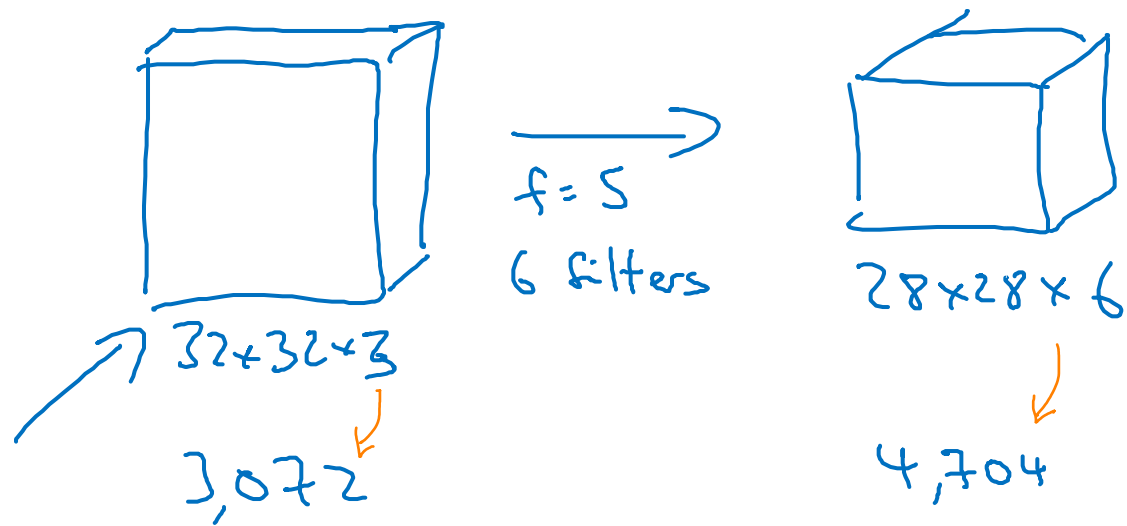


$5 \times 5 \quad - \quad 25$

$26$

$6 \times 26 = 156$ parameters

$32 \times 32 \times 3$

$3,072$

$f = 5$
6 filters

$28 \times 28 \times 6$

$4,704$

$3,072 \times 4,704 \approx 14M$ weight

a lot of parameters to train
when image is large

3072

4704

Andrew Ng

# Why convolutions

이 잘 동작하는가?

translation invariance 라는 좋은 특성을 가짐

| 10 | 10 | 10 | 0 | 0 | 0 |
|----|----|----|---|---|---|
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |
| 10 | 10 | 10 | 0 | 0 | 0 |

*

| 1 | 0 | -1 |
|---|---|----|
| 1 | 0 | -1 |
| 1 | 0 | -1 |
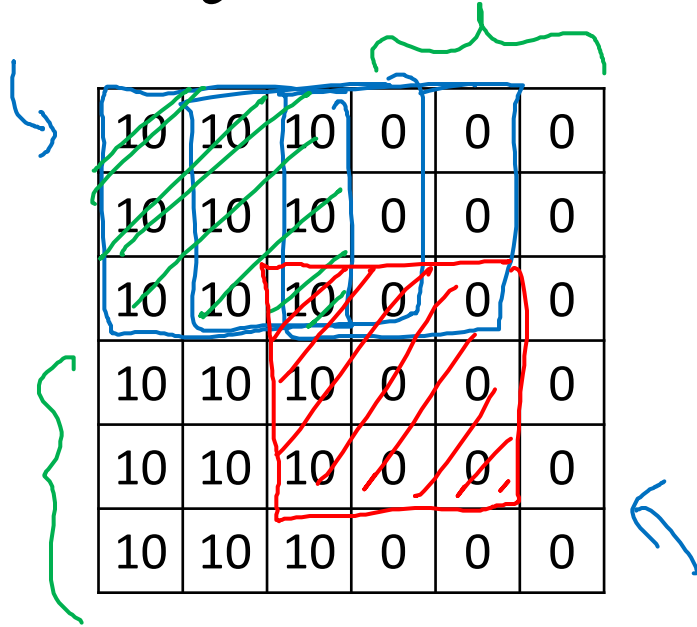
3×3

=

| 0 | 30 | 30 | 0 |
|---|----|----|---|
| 0 | 30 | 30 | 0 |
| 0 | 30 | 30 | 0 |
| 0 | 30 | 30 | 0 |

**Parameter sharing:** A feature detector (such as a vertical edge detector) that's useful in one part of the image is probably useful in another part of the image.
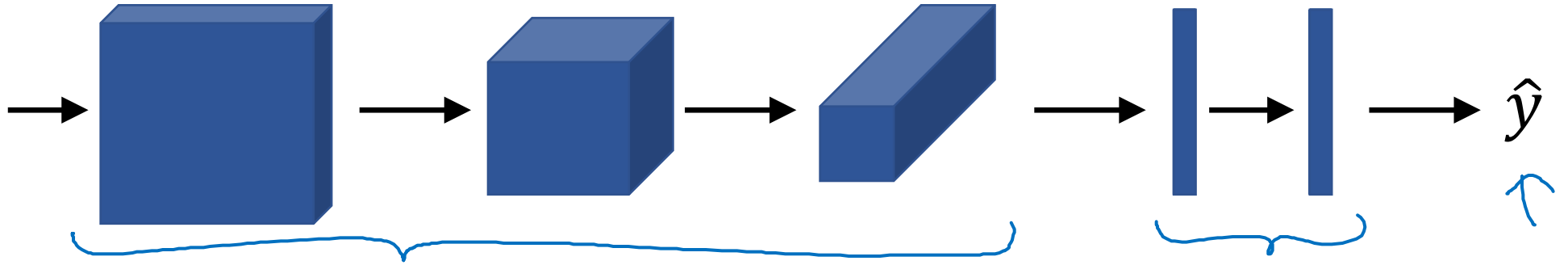
**Sparsity of connections:** In each layer, each output value depends only on a small number of inputs.

이 두 메커니즘 덕분에 네트워크는 아주 적은 파라미터를 갖게 되고, 이는 NN이 smaller training sets로 학습될 수 있게 해주며 오버피팅이 일어날 염려를 덜어준다.

Andrew Ng

# Putting it together

Training set $(x^{(1)}, y^{(1)}) \dots (x^{(m)}, y^{(m)})$.

$w, b$



$\hat{y}$

Cost $J = \dfrac{1}{m} \displaystyle\sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$

Use gradient descent to optimize parameters to reduce $J$