deeplearning.ai

# Convolutional Neural Networks

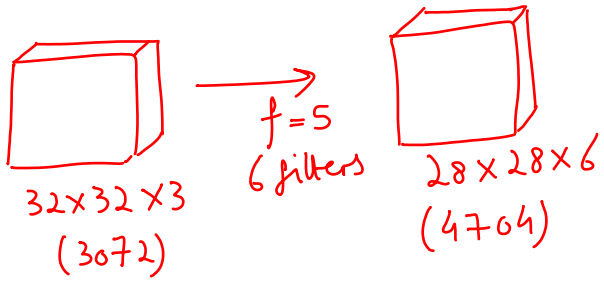
## Why convolutions?

# Why convolutions

2 Adv over FC layers

① parameter sharing

② Sparsity of connections




Actual params

5x5 → filter

# filters

25x6 = 150

+6 bias params

= 156 params

Imagine If these 2 layers were fully connected, then

# params = $n^{[l+1]} \times n^{[l]}$

= 4704 x 3072

≈ 12M (lot of params for a small Image)

# Why convolutions

| | | | | | |
|---|---|---|---|---|---|
| 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 |

=

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

Note
CNNs also are "translation Invariant" meaning If Image of cat shifts to the right, it still is able to detect a cat, why? Because the filters are all applied to each iteration of the stride & the shifted Images' "cat features" will be captured in a later/earlier stride's convolution

This came from a very small # of calculations ie, only 9 cells were involved ⇒ this cell is "derived from" or "connected to" only 9 (out of 36) cells ⇒ sparse

**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.

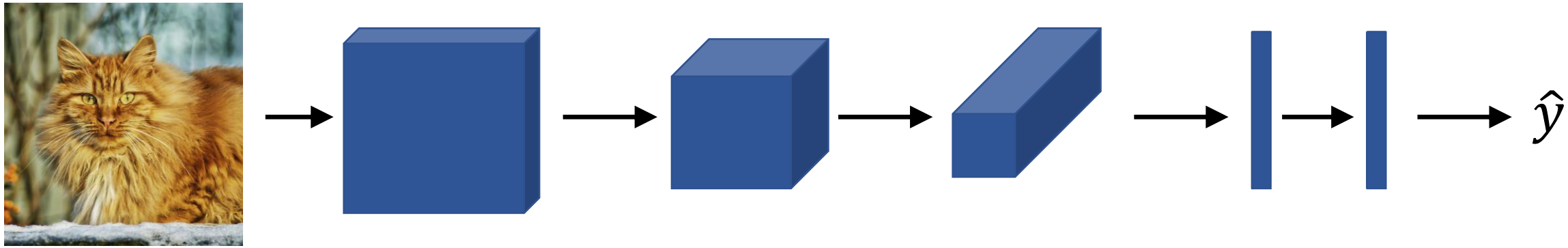
Same filter is used across all stride iterations of the Image to detect vertical edge

**Sparsity of connections:** In each layer, each output value depends only on a small number of inputs.

Andrew Ng

# Putting it together

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




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

Use gradient descent to optimize parameters to reduce $J$