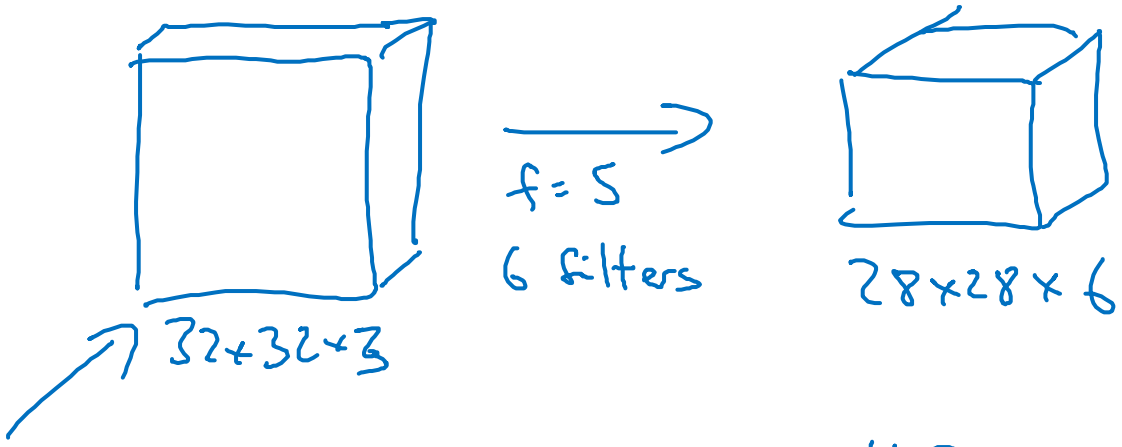
deeplearning.ai

Convolutional Neural Networks

Why convolutions?

Why convolutions

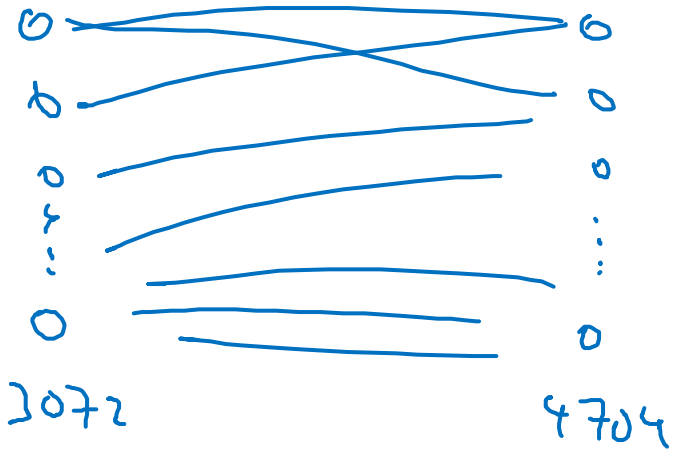
I think the two main advantages of convolutional layers over, just using fully connected layers, and the advantages are parameter sharing and sparsity of connections.



5×5 — 25

26

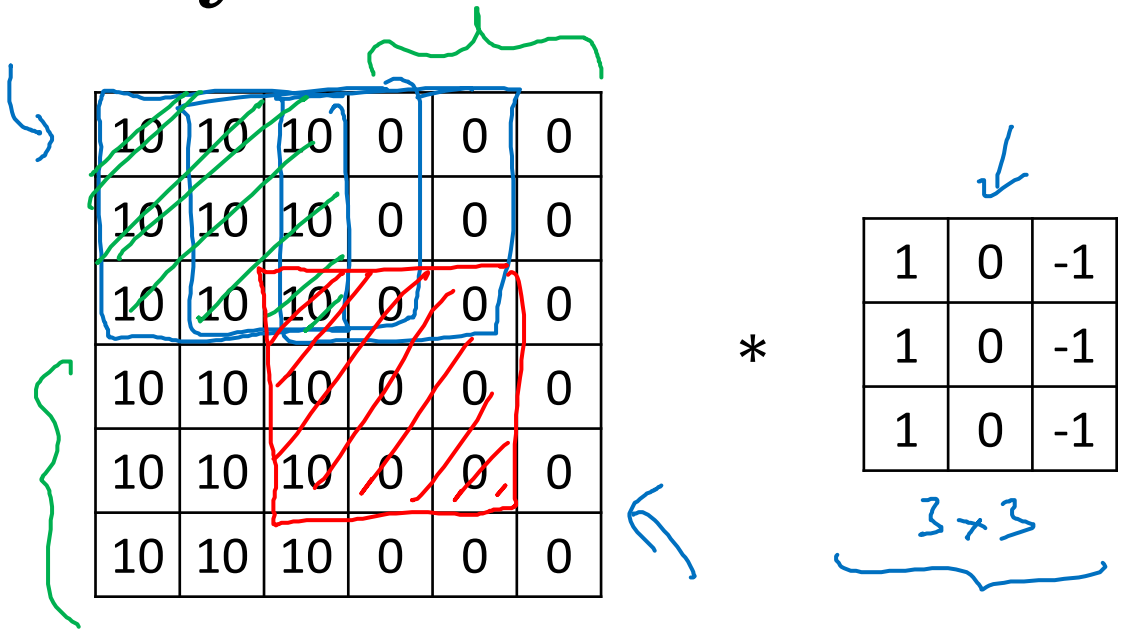
6 × 26 = 156 parameters



3,072 × 4,704 ≈ 14M

Andrew Ng

Why convolutions



translation invariance

And a convolutional structure helps the neural network encode the fact that an image shifted a few pixels should result in pretty similar features and should probably be assigned the same output label.

And th fact that you're applying the same filter yields all the position of the image, of both the early layers in the later layers. That hopes in your network automatically learn to be with a more robust, so to better capture this desirable property of translation invariance.

But being to share in this case the same nine parameters to compute all 16 of these outputs as one of the ways the number of parameters is reduced.

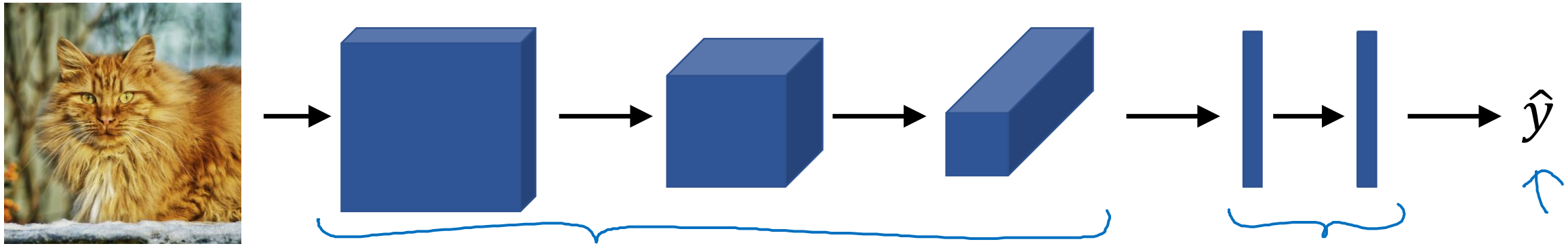
Parameter sharing: A feature detector (such as a vertical edge detector) that's useful in one part of the image is probably useful in another part of the image.

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

Andrew Ng

Putting it together

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



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