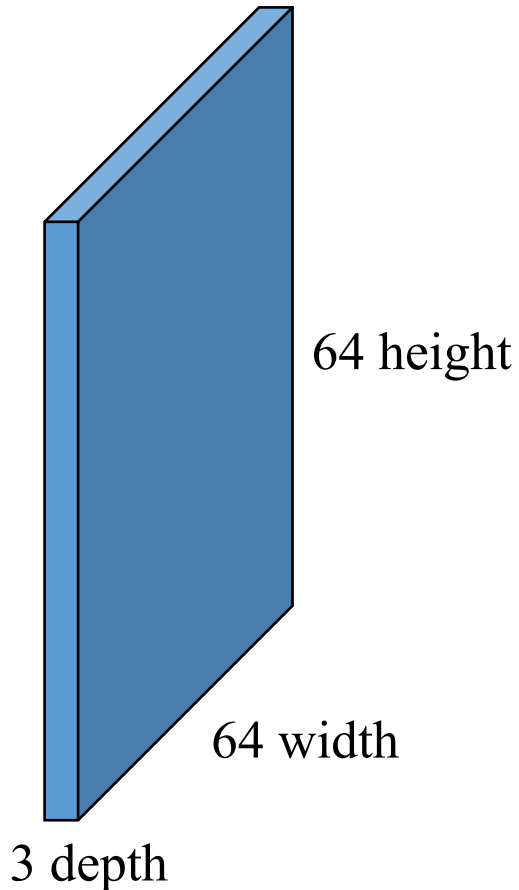
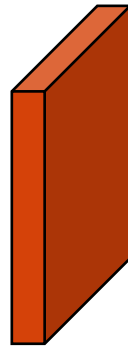


Convolution

64x64x3 image x



5x5x3 filter w

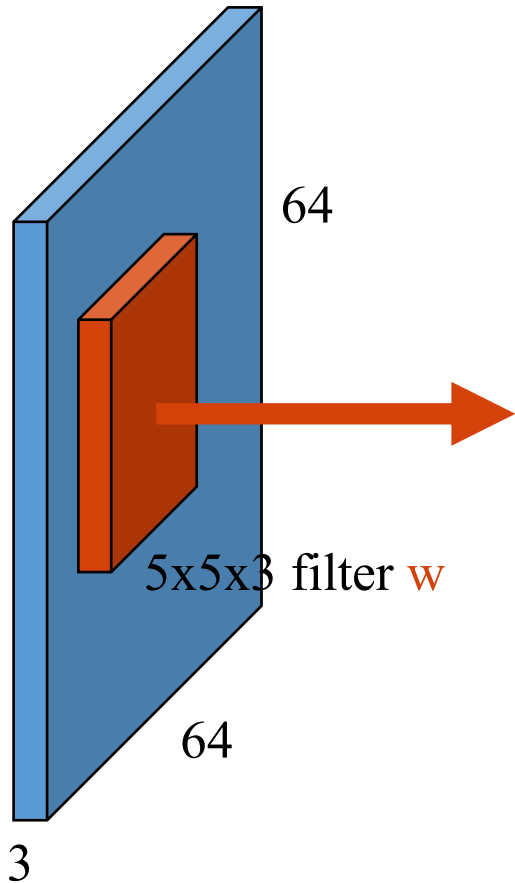


Convolve the filter with the image

“Slide over the image spatially,
computing dot products”

Convolution

64x64x3 image $\mathbf{x}$



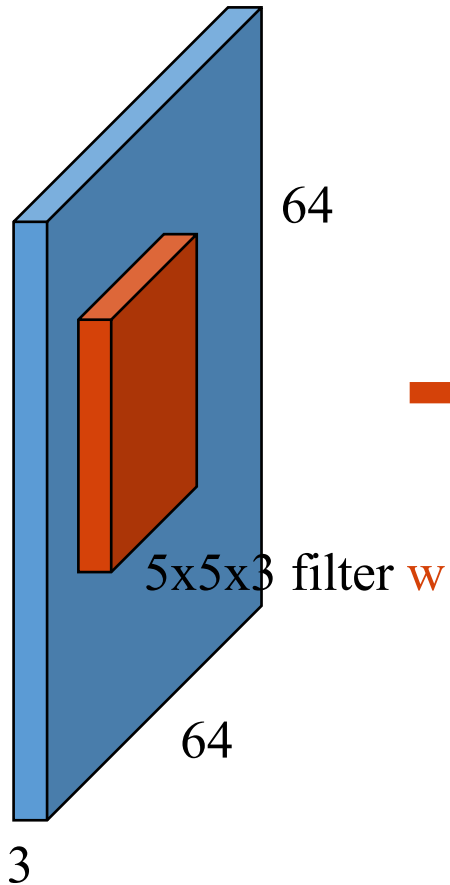
1 number:

the result of taking a dot product between the filter and a small 5x5x3 chunk of the image.

$$\mathbf{w}^T \mathbf{x} + \mathbf{b}$$

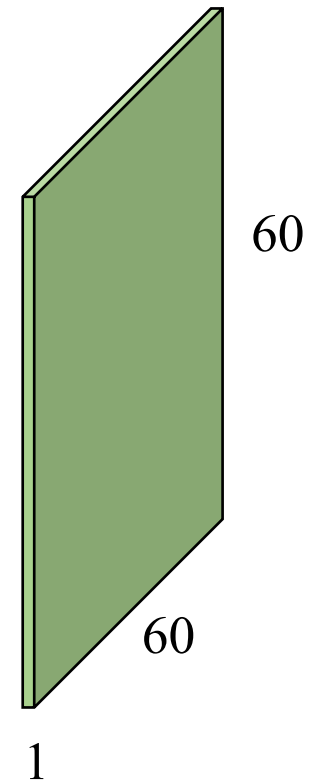
Convolution

64x64x3 image x



Convolve over all spatial locations

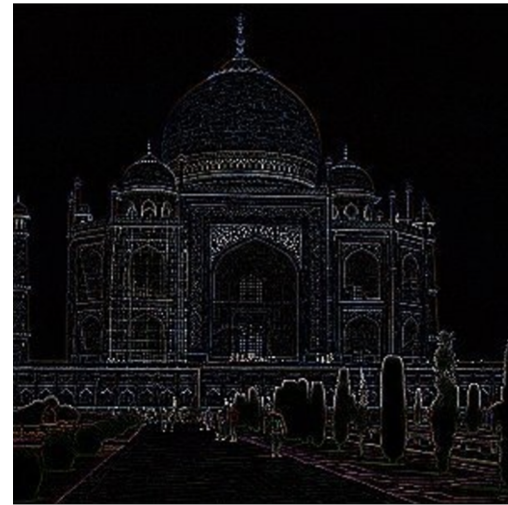
activation/feature map



Convolution - Intuition



	0	1	0	
	1	-4	1	
	0	1	0	

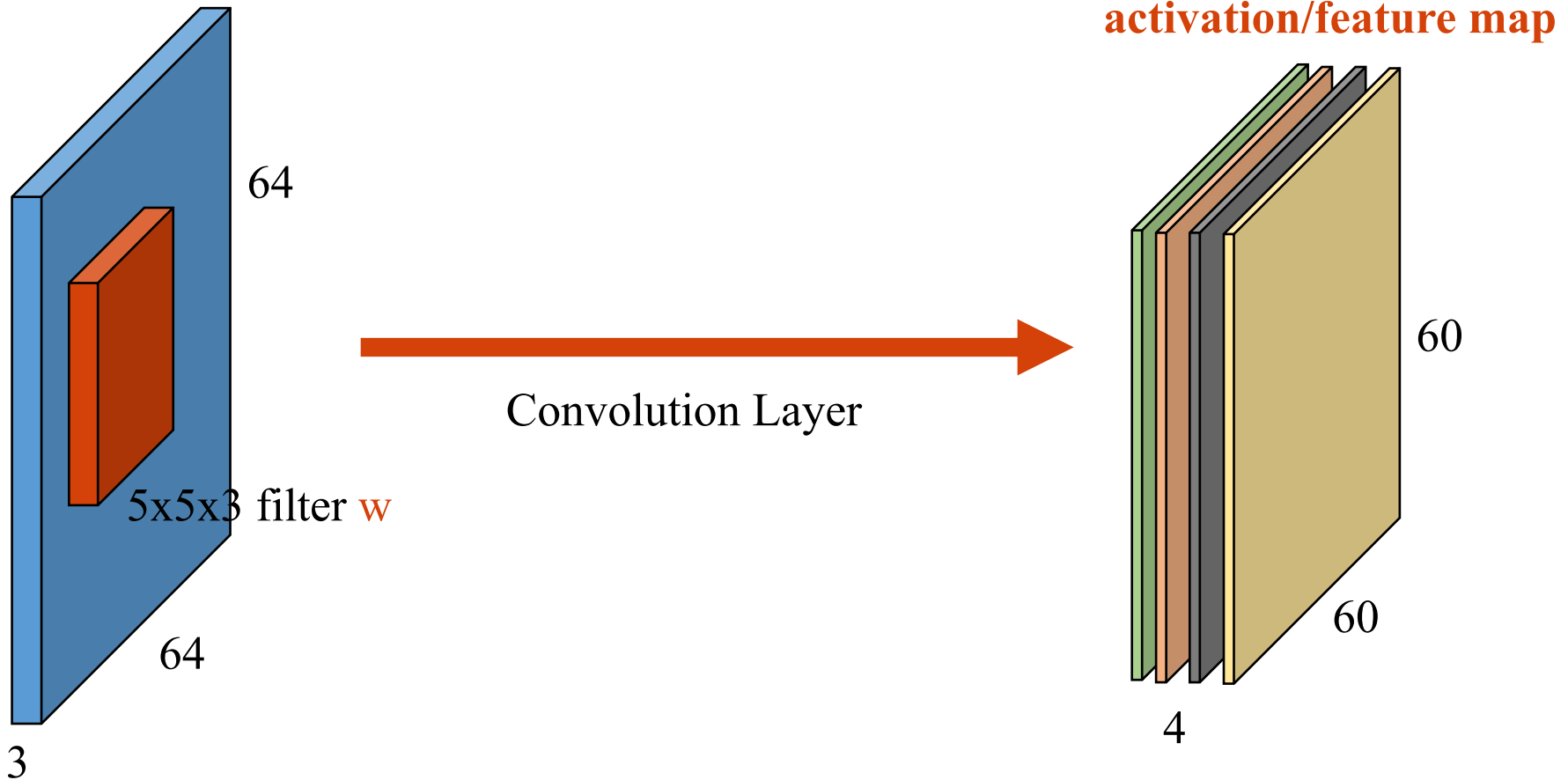


❑ Intuition of convolution

- ❑ **Similarity measurement:** image chunk and convolutional filter.
- ❑ Looking for **local patterns**.
- ❑ Patterns are not fixed and will be learned during training.

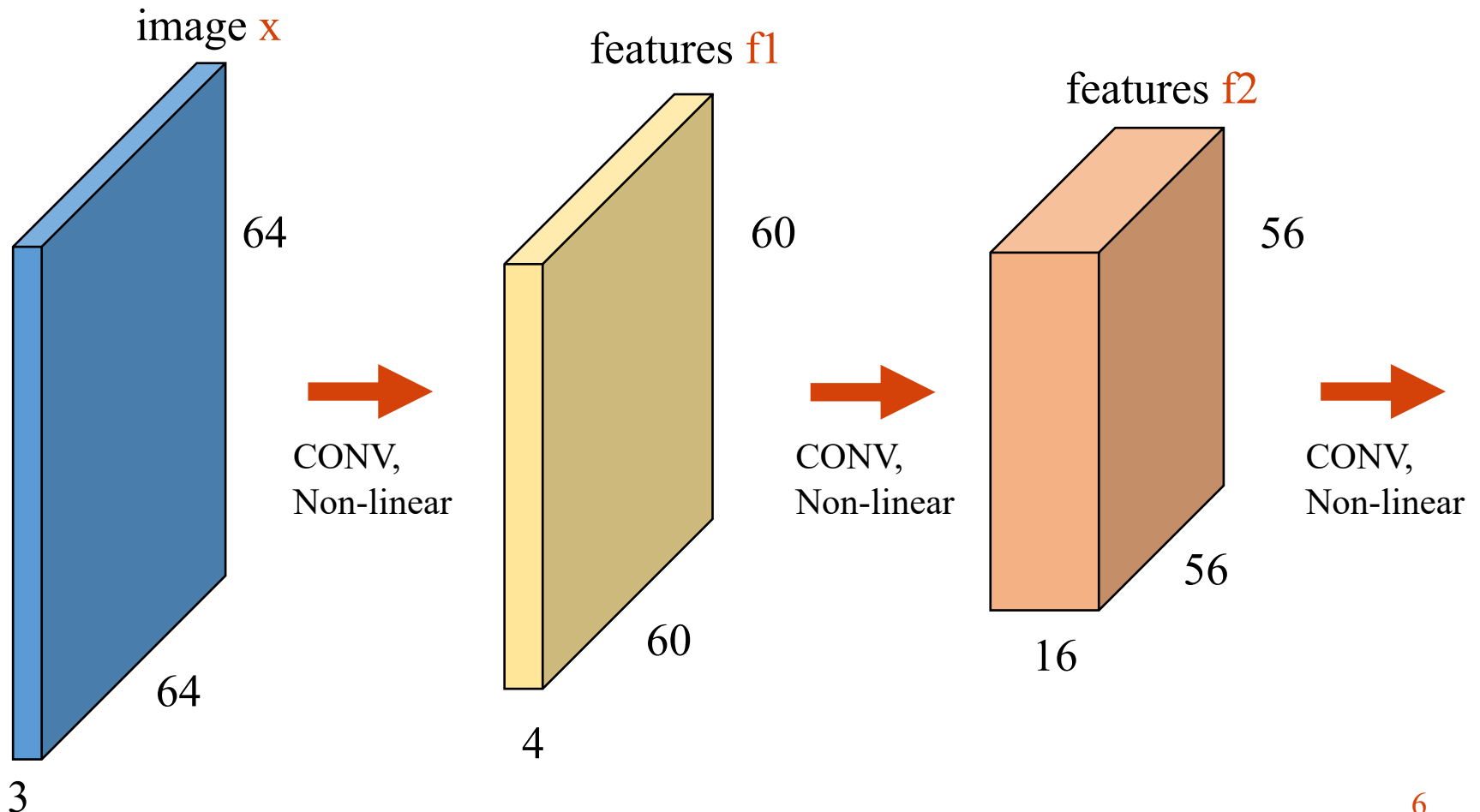
Convolutional Layer

- ❑ Stacking multiple convolutions in one layer



Convolutional Feature Maps

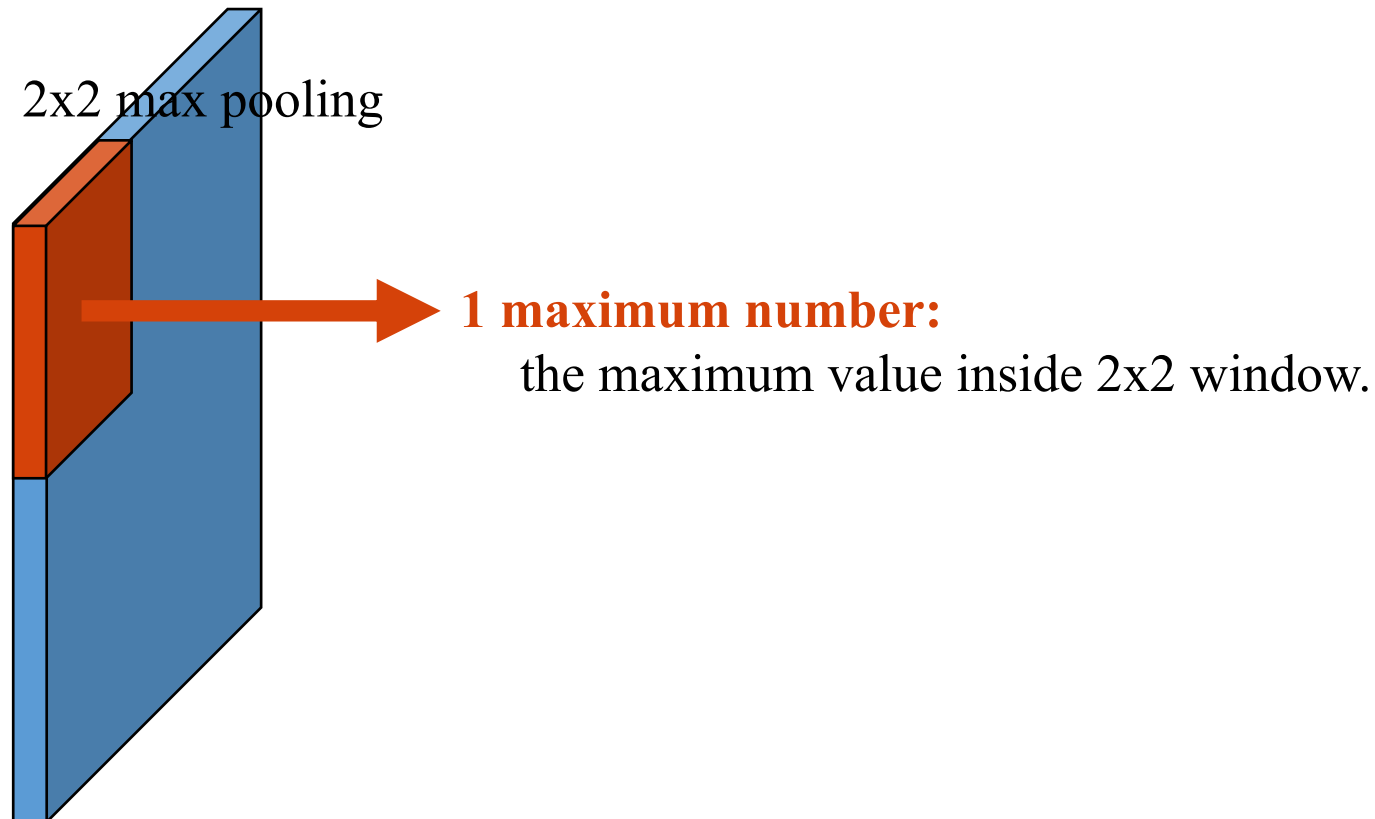
- A feature hierarchy by stacking convolutional layers



Max Pooling

- ❑ Max pooling operation

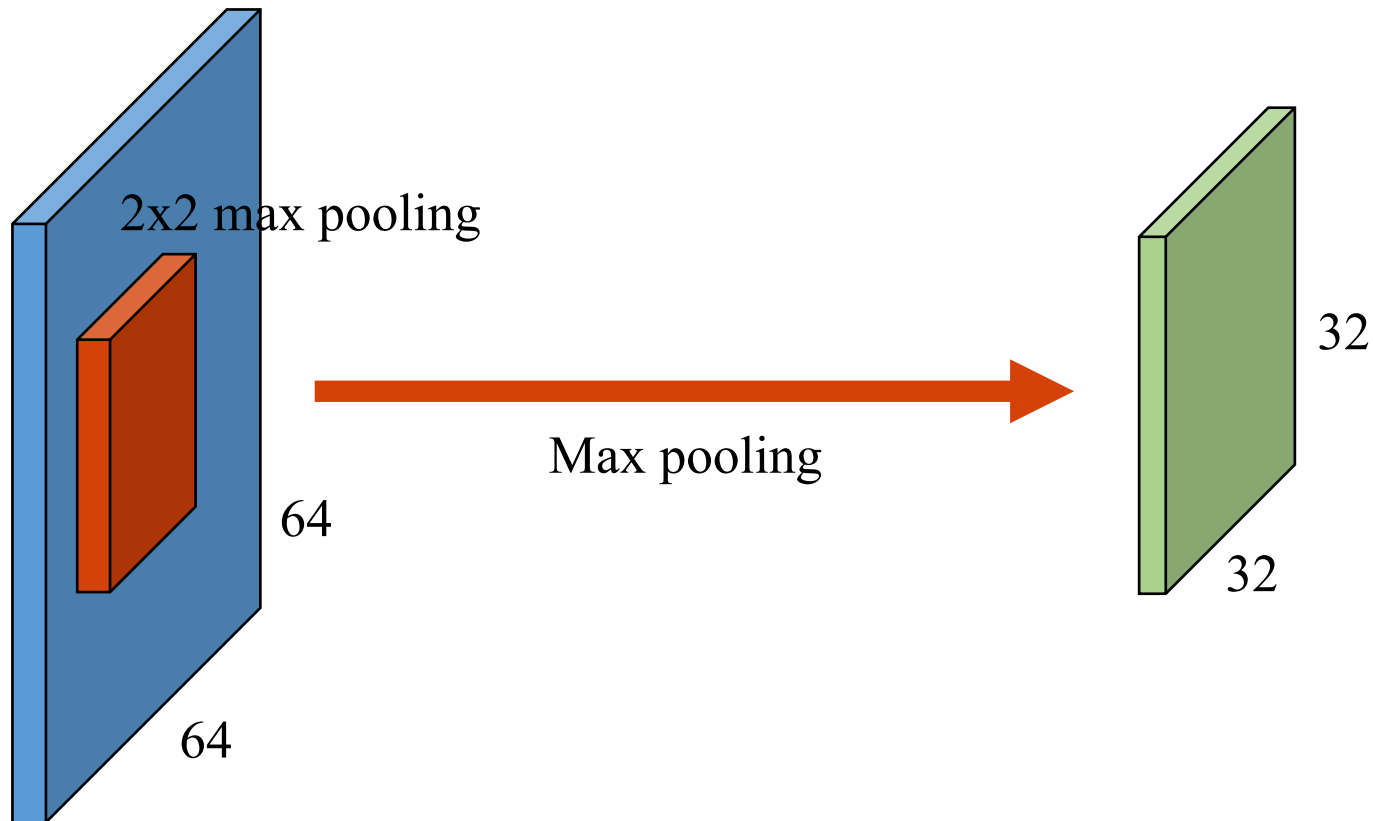
- ❑ Slides a small **non-overlapping** window.
 - ❑ Picks the **maximum value** inside each window.



Max Pooling

- ❑ Max pooling operation

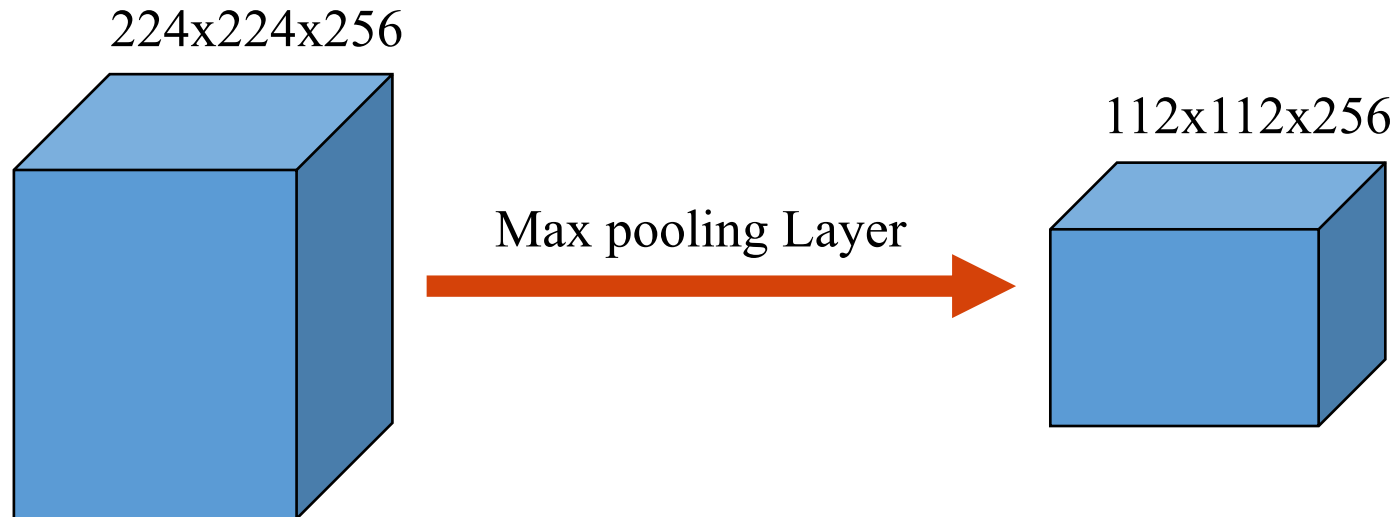
- ❑ Slides a small **non-overlapping** window.
 - ❑ Picks the **maximum value** inside each window.



Max Pooling Layer

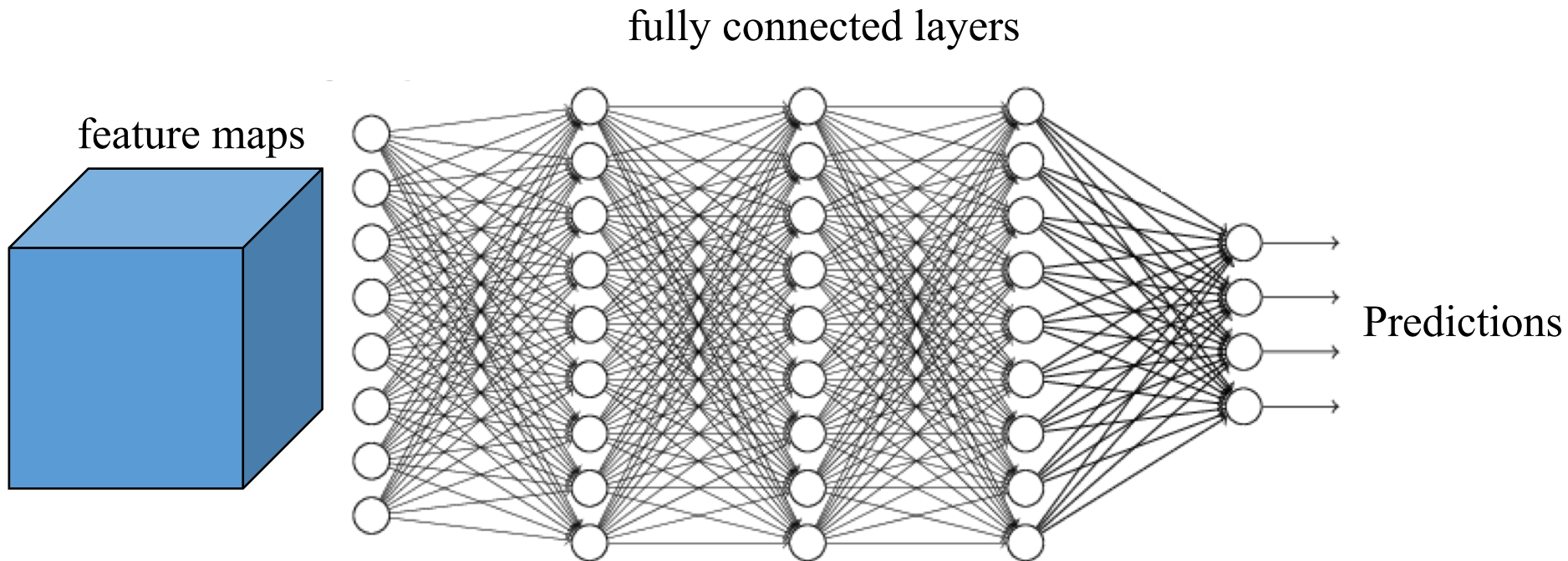
□ Advantages

- Reduces the redundancy of convolutions.
- Makes the representations smaller and more manageable.
- Reduces the number of parameters, controls overfitting.
- Invariant to small transformations, distortions and transitions.



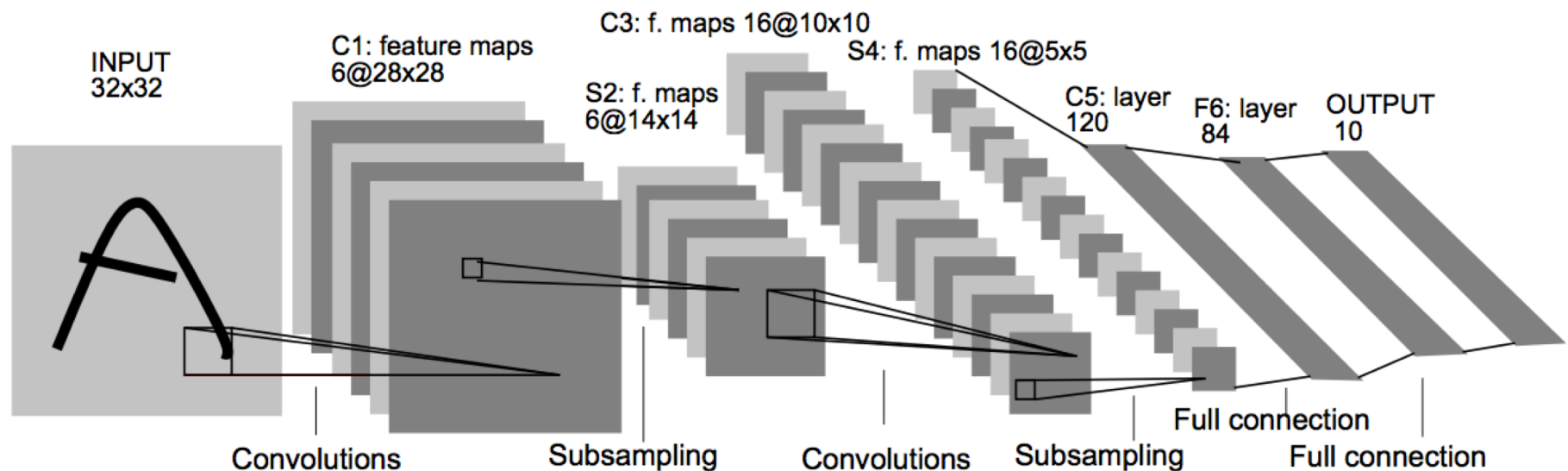
Fully Connected Layer

- ❑ A classifier on top of feature maps
- ❑ Maps high-dimensional matrix to predictions (1-D vector)



Convolutional Neural Network

- ❑ First CNN trained with backpropagation
 - ❑ Three different layers: convolution, subsampling, fully-connected
 - ❑ Training CNN: gradient-decent optimization (backpropagation)



Training Convolution Neural Network

□ Gradient Descent

- Randomly initialize weights at each layer.
- Compute a forward pass and calculate the cost function.
- Calculate the with respect to each weight during a backward pass.
- Update weights.

