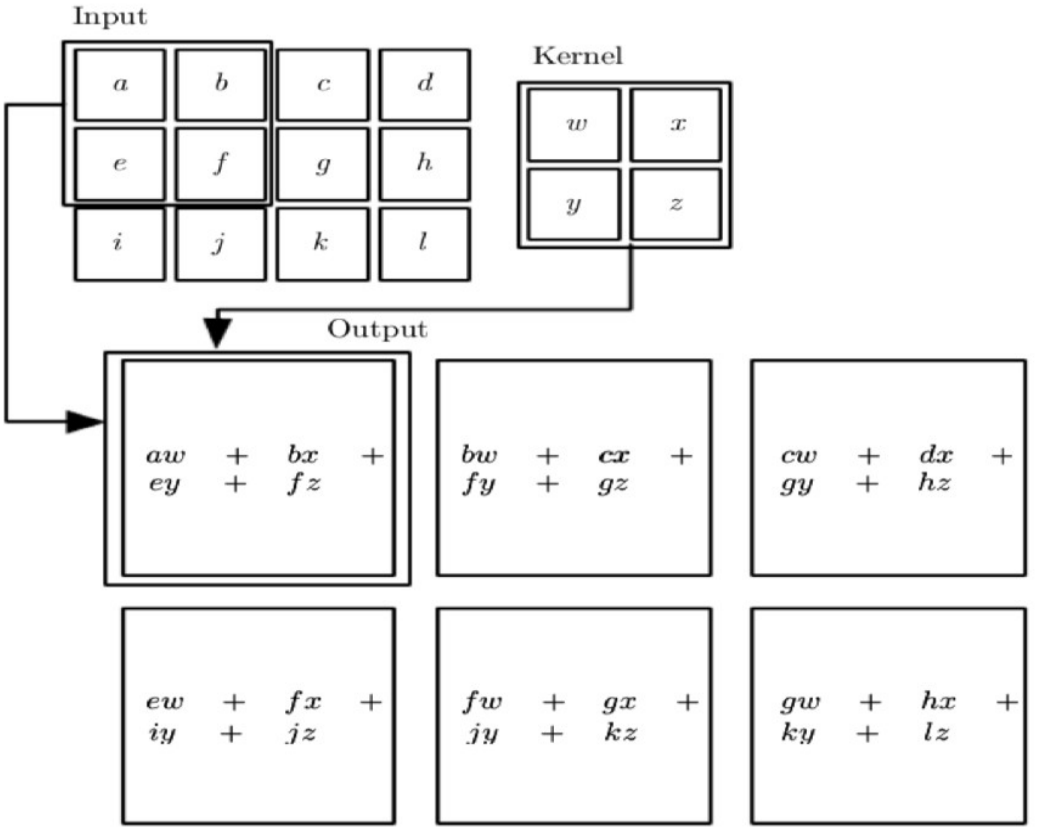
CNN Hands on

MNIST

Convolution

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

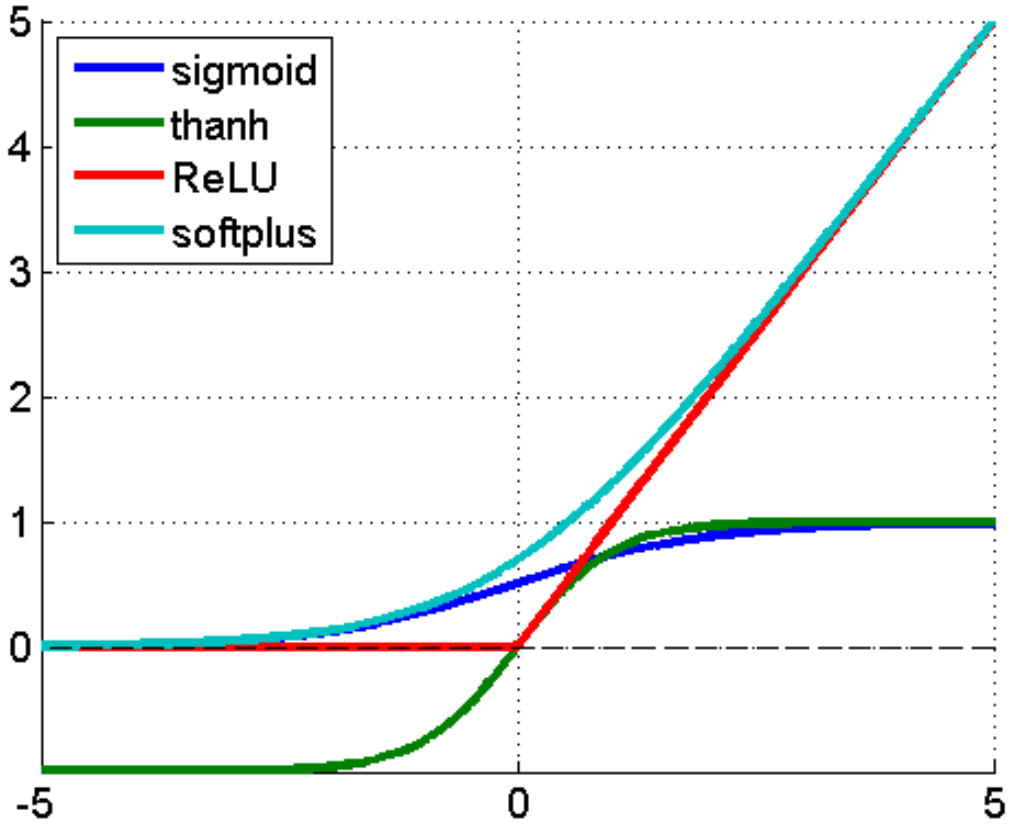
In Model:
“Valid”
“Same”



Activation Functions

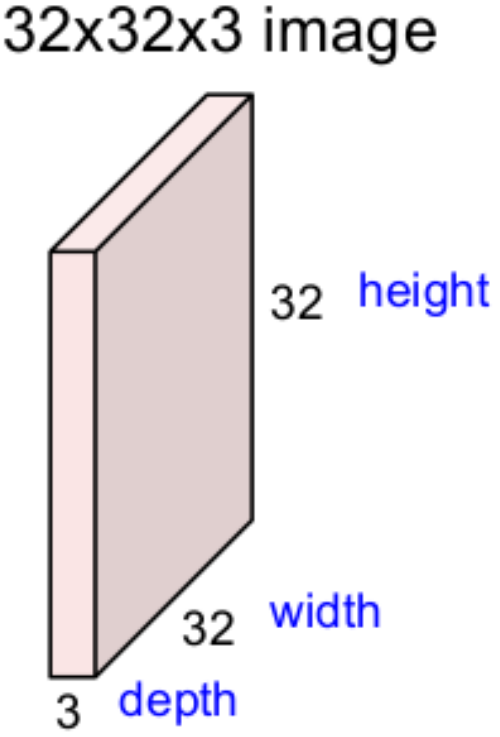
- Sigmoid: $f(x) = \frac{1}{1+exp(-x)}$
- Hyper-tengent $f(x) = tanh(x)$
- ReLU $f(x) = \max(0, x)$
- Soft-Plus $log(1 + e^x)$

Activation Functions



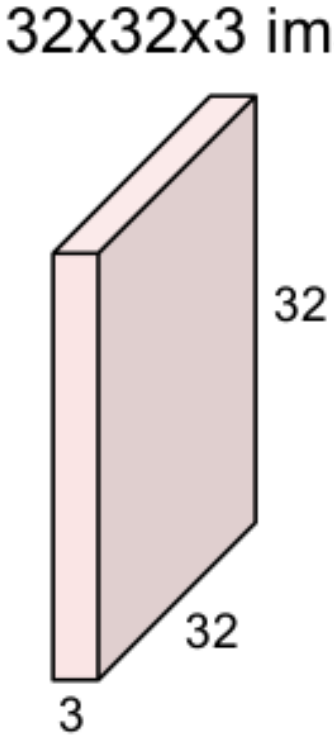
CNN Working Intuition

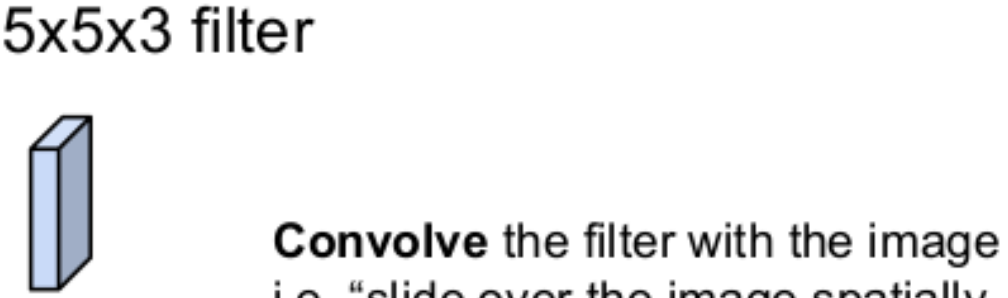
Convolution Layer



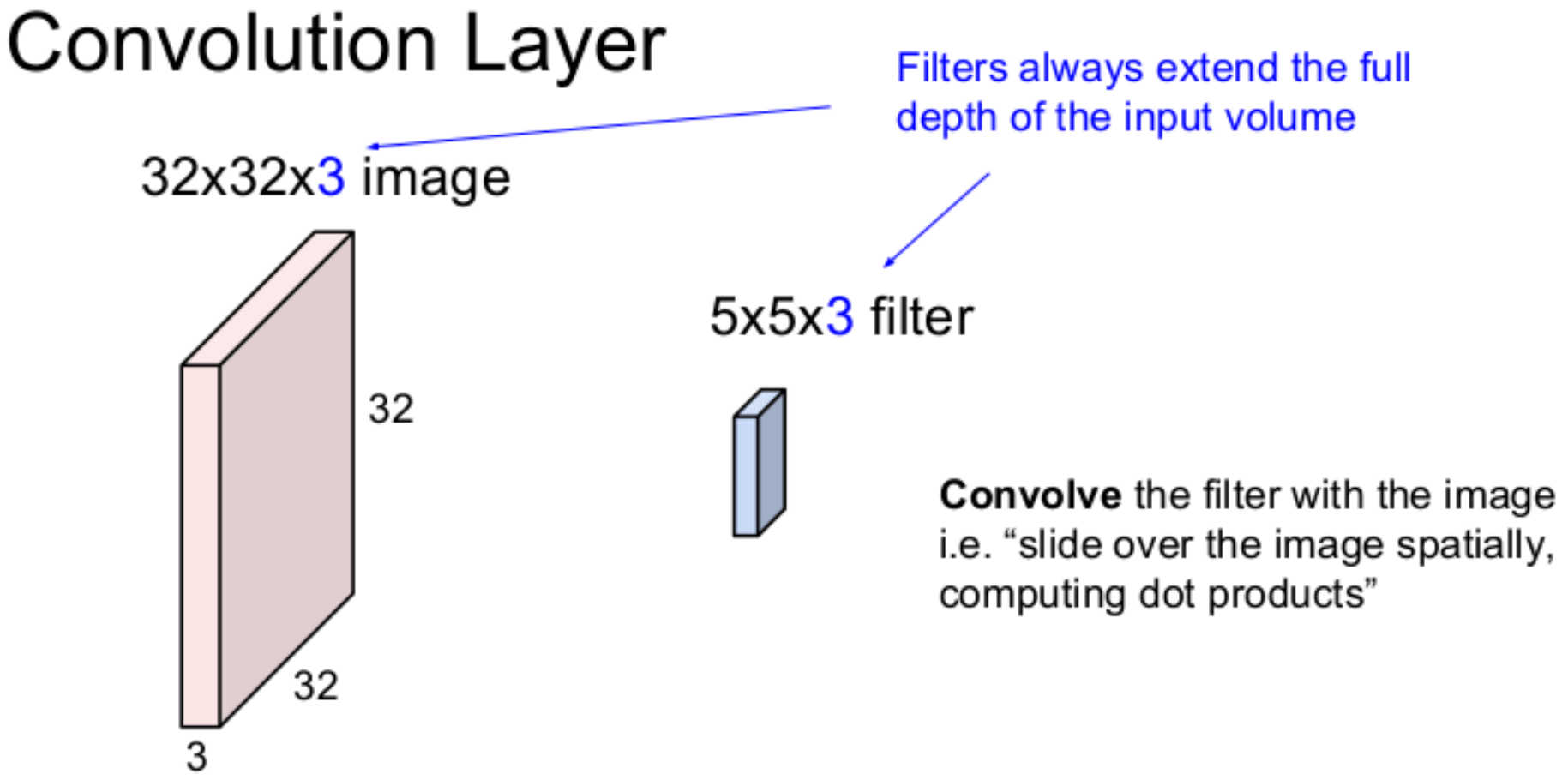
CNN Working Intuition

Convolution Layer



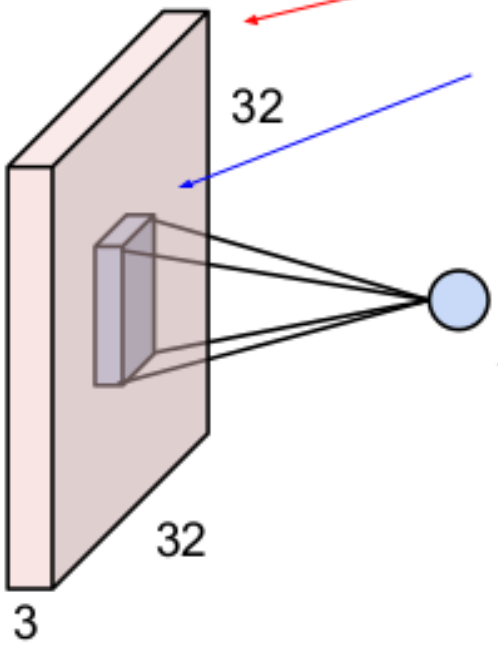


CNN Working Intuition



CNN Working Intuition

Convolution Layer



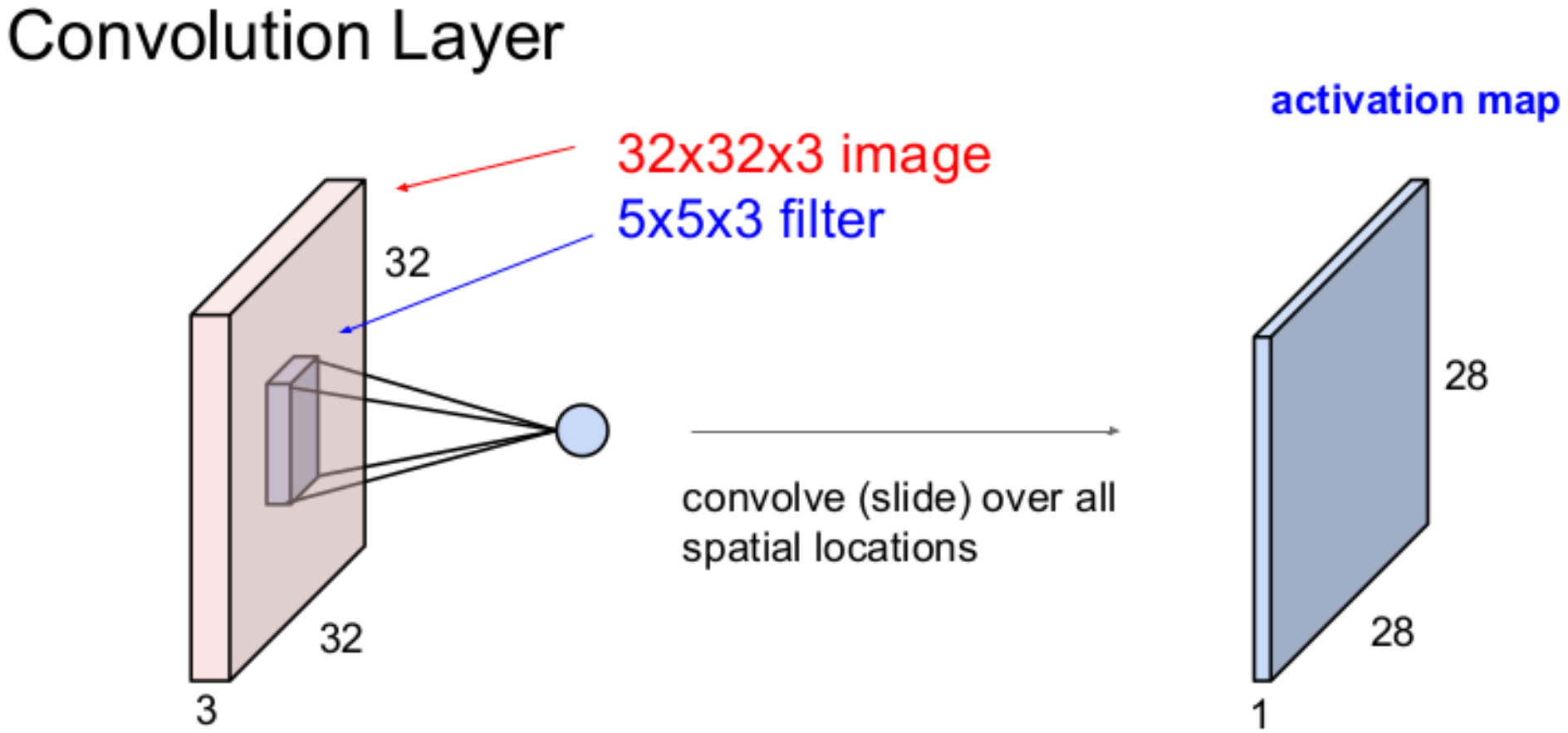
32x32x3 image

5x5x3 filter w

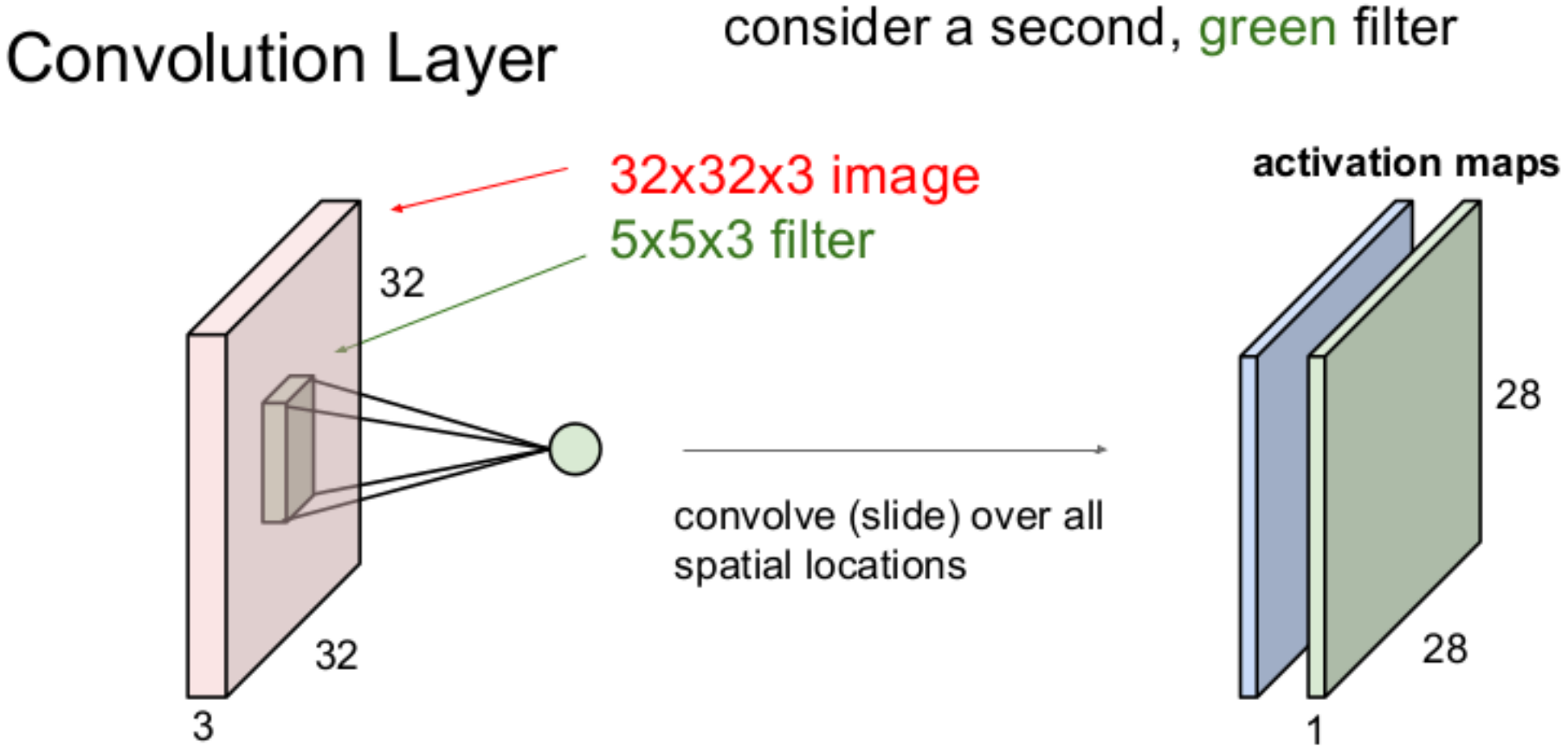
1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

CNN Working Intuition

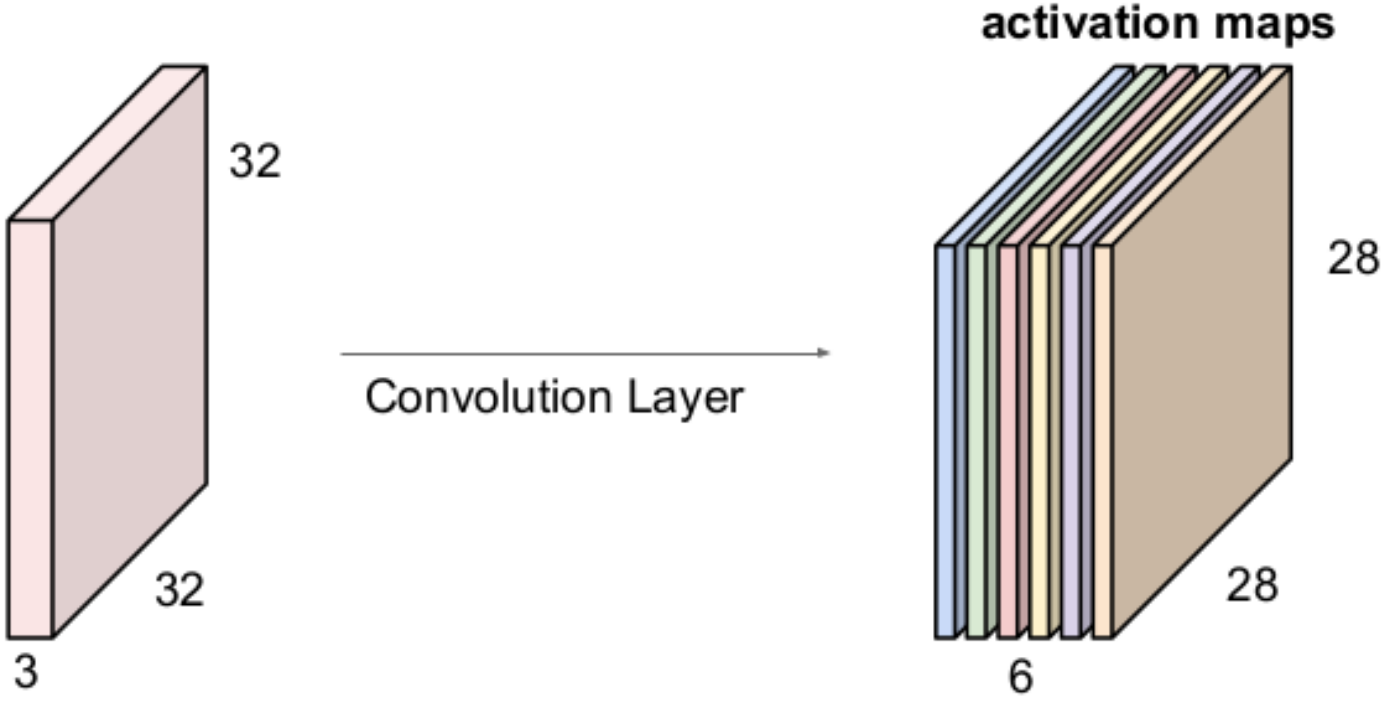


CNN Working Intuition



CNN Working Intuition

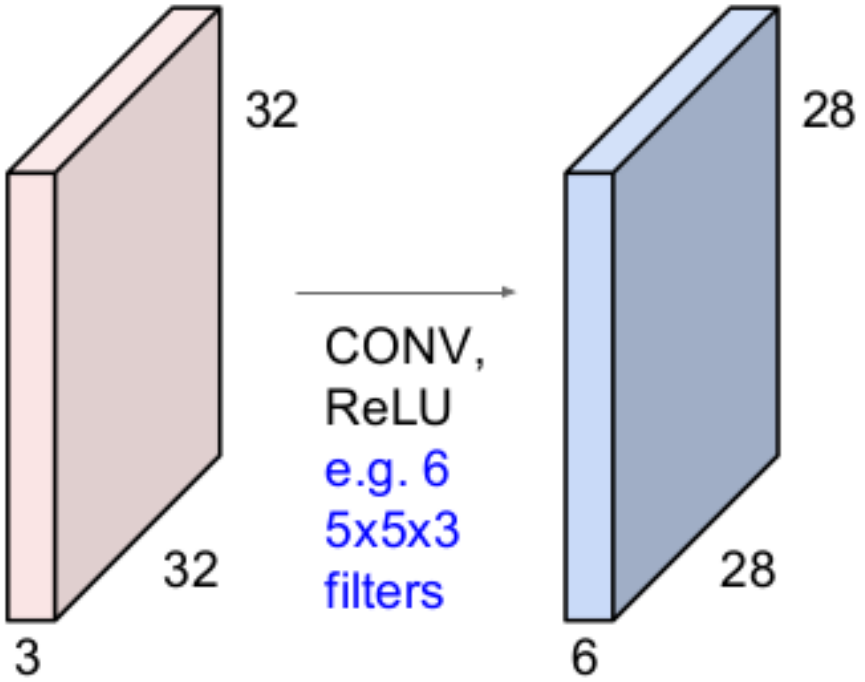
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



We stack these up to get a "new image" of size 28x28x6!

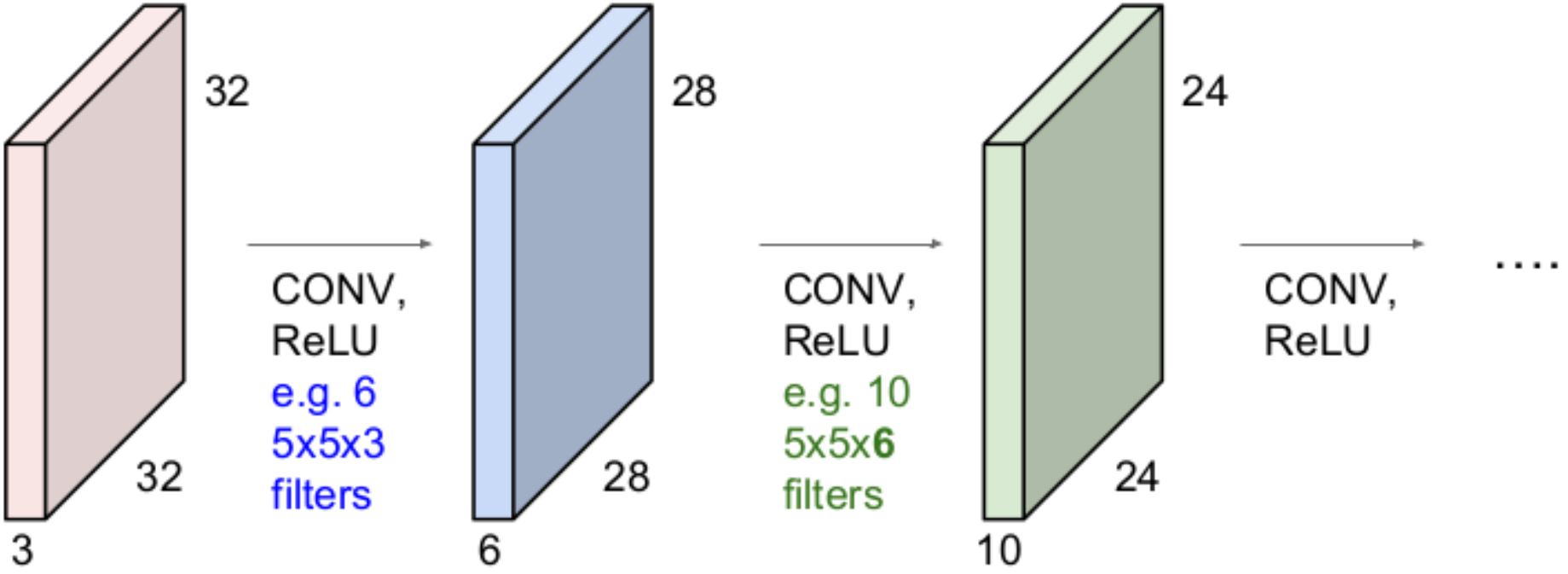
CNN Working Intuition

Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



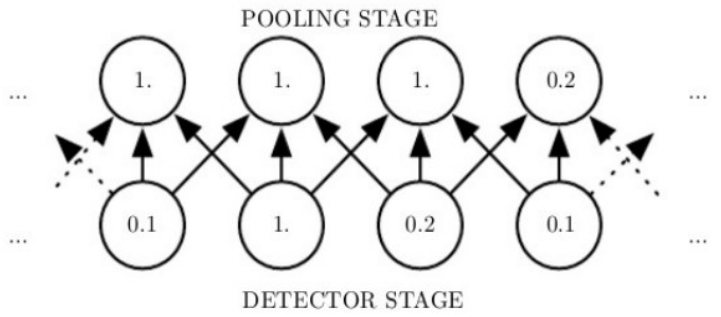
CNN Working Intuition

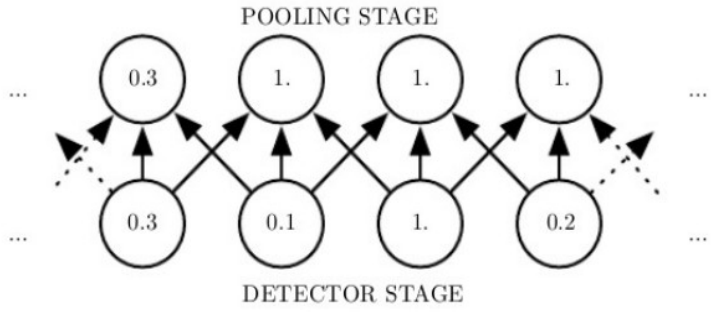
Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



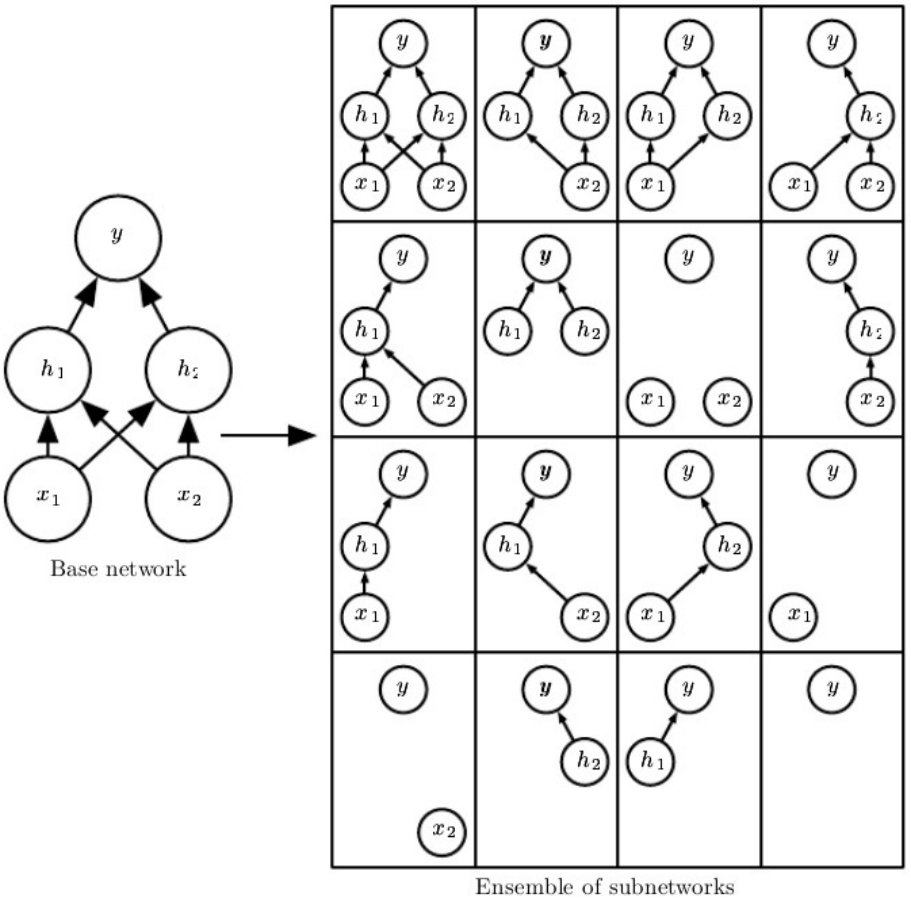
Building Translation Invariance via Pooling

- Pooling adds infinitely strong prior that the function the layer learns must be invariant to small translations.
- It improves the statistical efficiencey of the network
- It summarizes the responses over a whole neighborhood.
- It may handle inputs of varying size.





Dropout



Softmax

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

Objective Function

- mean_squared_error
- mean_absolute_error
- binary_crossentropy
- categorical_crossentropy

Categorical_Crossentropy

- Return the cross-entropy between an appriximating distribution and a true distribution.
- It measures the average numbr of bits needed to identify an event from a set of possibilities.
- p=true_dist and q=coding_dist
- Test Score: mean(loss)

$$H(p, q) = -\sum_x p(x) \log(q(x))$$

Thanks