

رسالة محمد



# مبانی بینایی کامپیوتر

مدرس: محمدرضا محمدی

بهار ۱۴۰۲

یادگیری ویژگی

Feature Learning

# شبکه‌های عصبی

- یک شبکه عصبی چندلایه شامل تعدادی لایه خطی و توابع فعال‌سازی غیرخطی است

- شبکه ۱ لایه:

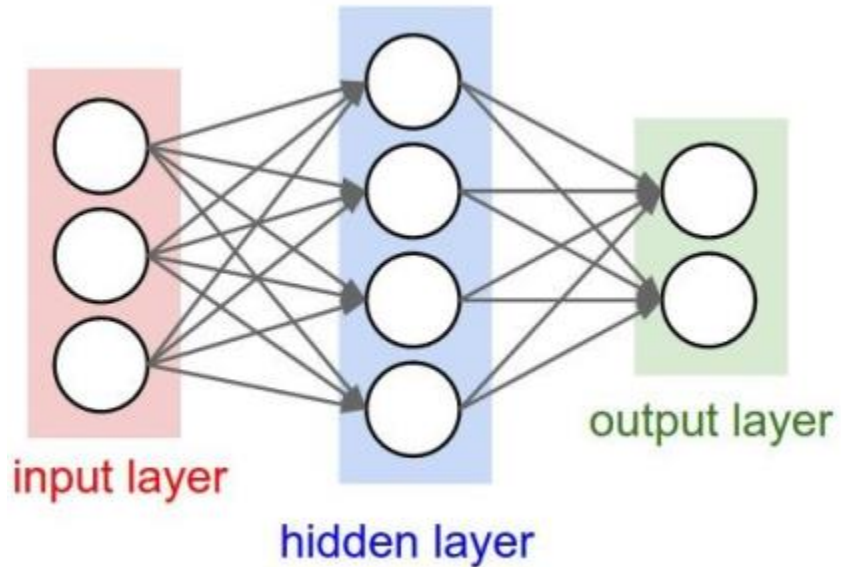
$$y = f(Wx)$$

- شبکه ۲ لایه:

$$y = f_2(W_2 f_1(W_1 x))$$

- شبکه ۳ لایه:

$$y = f_3(W_3 f_2(W_2 f_1(W_1 x)))$$

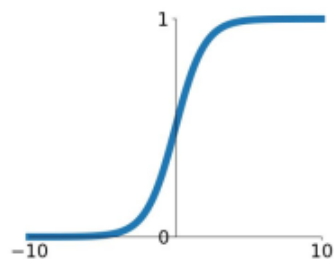


# توابع فعال سازی

- به دلیل خطی بودن ضرب داخلی، وجود توابع فعال سازی غیرخطی ضروری است

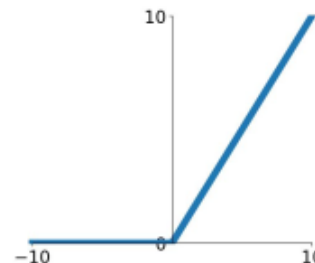
## Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



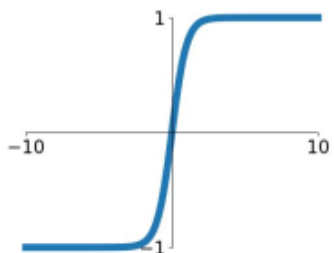
## ReLU

$$\max(0, x)$$



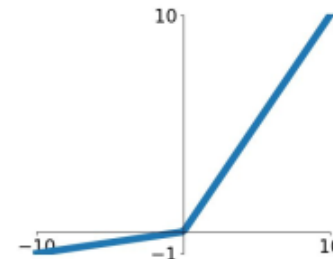
## tanh

$$\tanh(x)$$

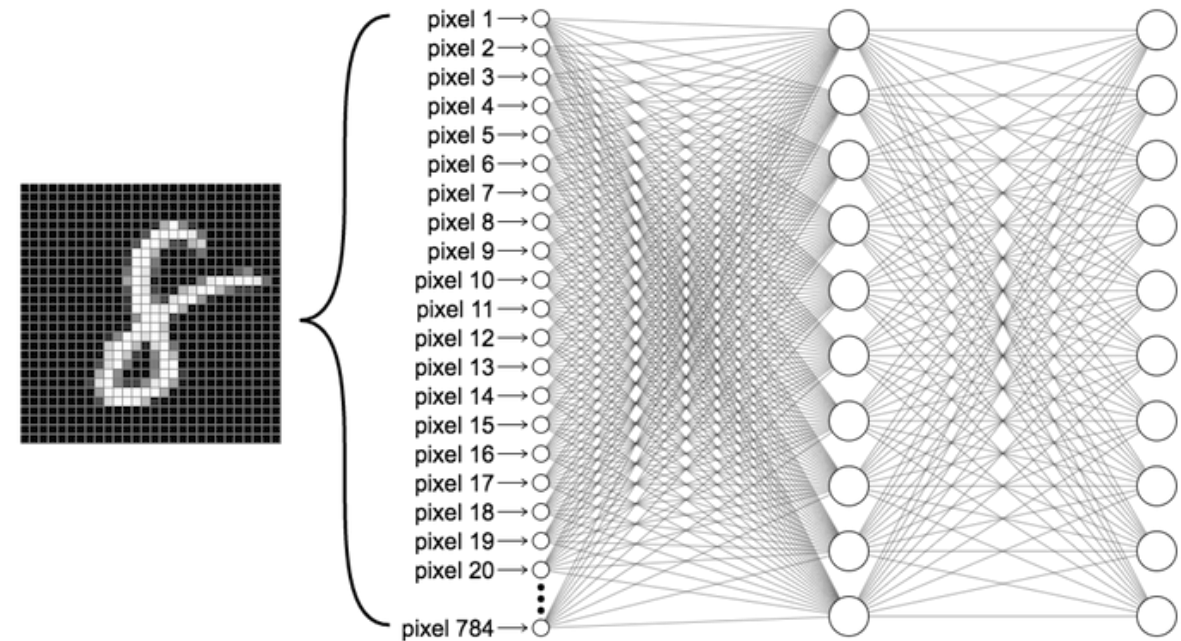
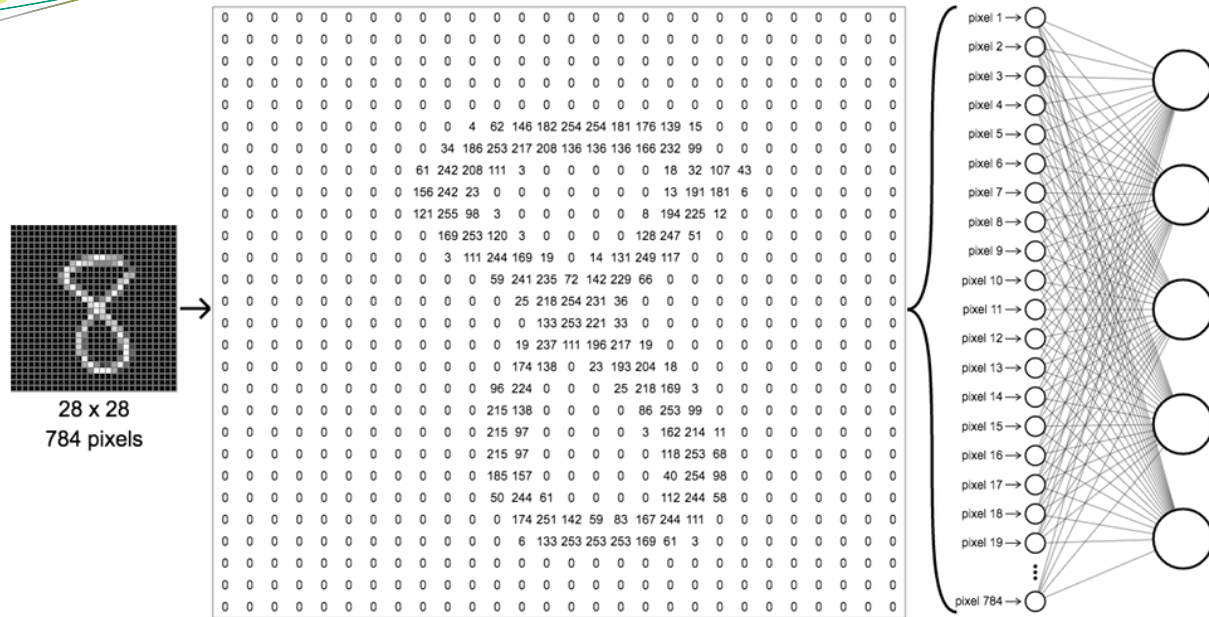


## Leaky ReLU

$$\max(0.1x, x)$$



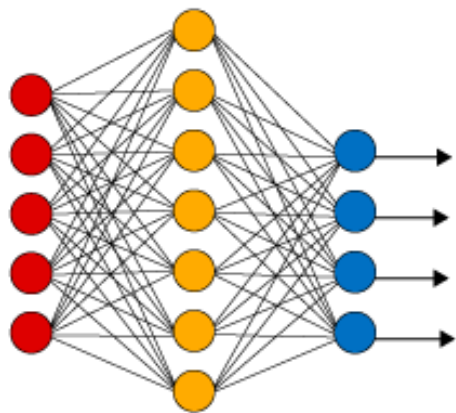
# شبکه‌های عصبی



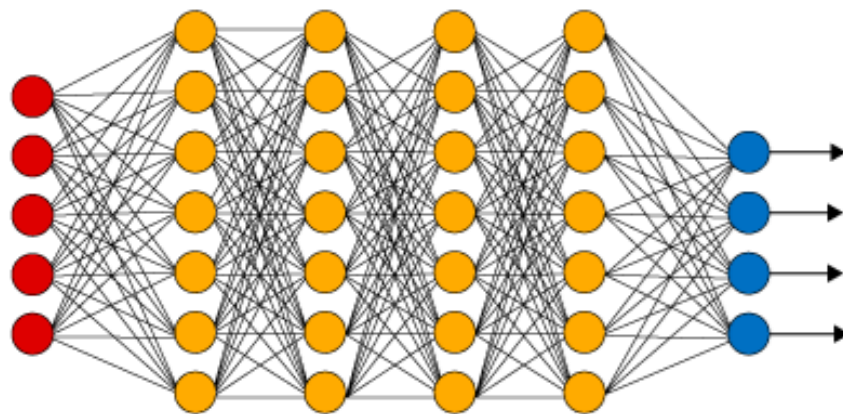
# شبکه‌های عصبی عمیق

- آیا یک شبکه دارای لایه‌های زیاد می‌تواند منجر به بهبود طبقه‌بندی تصویر شود؟
- مهمترین ایراد این ساختار در پردازش تصویر آن است که اطلاعات همسایگی را لحاظ نمی‌کند
- به عبارت دیگر، دانش بدست آمده را میان پیکسل‌های تصویر به اشتراک نمی‌گذارد
- ایده اصلی در پیشرفت یادگیری عمیق در حوزه بینایی کامپیوتر استفاده از لایه‌های کانولوشنی است

Simple Neural Network



Deep Neural Network



● Input Layer

● Hidden Layer

● Output Layer

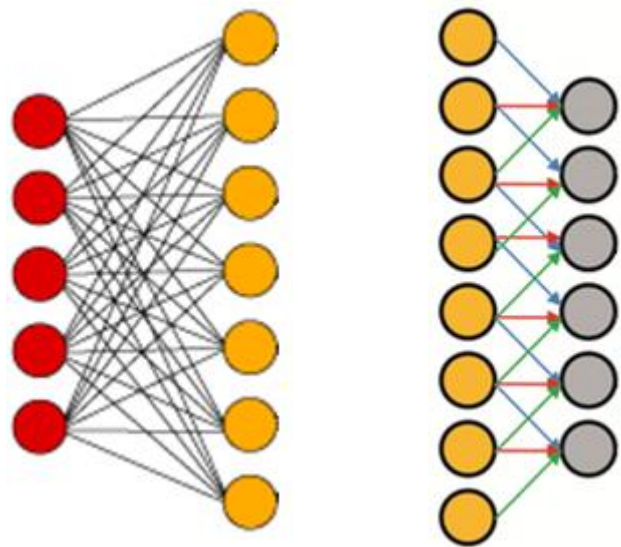
# شبکه‌های عصبی کانولوشنی

Convolutional Neural Networks



# کانولوشن

- در لایه‌های کاملاً متصل، مقدار هر نورون در لایه خروجی وابسته به تمام نورون‌ها در لایه قبل است
- کانولوشن یک‌بعدی مشابه با لایه کاملاً متصل است اما هر نورون خروجی تنها به بخشی از نورون‌های لایه ورودی متصل است
- در پردازش تصاویر از کانولوشن دوبعدی استفاده می‌شود



$3_0$	$3_1$	$2_2$	1	0
$0_2$	$0_2$	$1_0$	3	1
$3_0$	$1_1$	$2_2$	2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

# مثال: عملگر Sobel

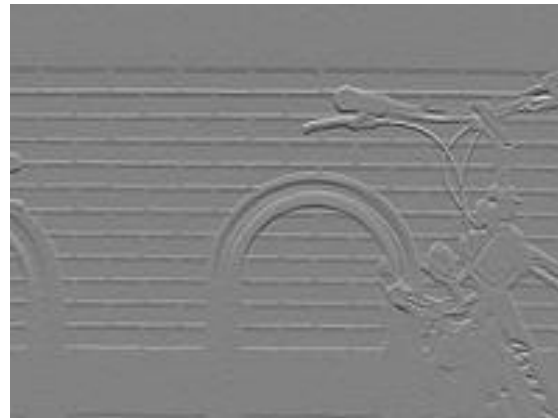


$G_y$

+1	0	-1
+2	0	-2
+1	0	-1

$G_x$

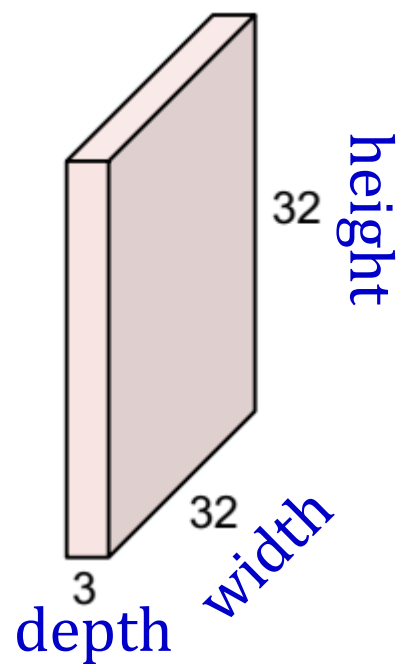
+1	+2	+1
0	0	0
-1	-2	-1



# لایه کانولوشنی

خروجی لایه کانولوشنی حاصل فیلتر کردن  
ماتریس ورودی با فیلتر مربوطه است که به  
صورت مکانی بر روی آن لغزانده می شود

ورودی یک ماتریس  
۳ بعدی است



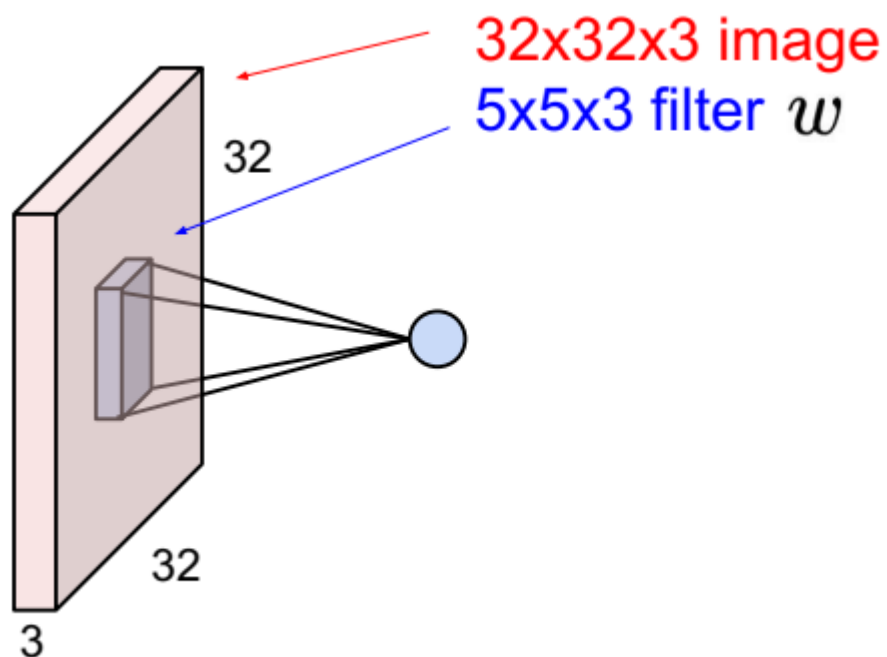
5x5x3 filter



# لایه کانولوشنی

خروجی لایه کانولوشنی حاصل فیلتر کردن ماتریس ورودی با فیلتر مربوطه است که به صورت مکانی بر روی آن لغزانده می شود

ورودی یک ماتریس ۳ بعدی است



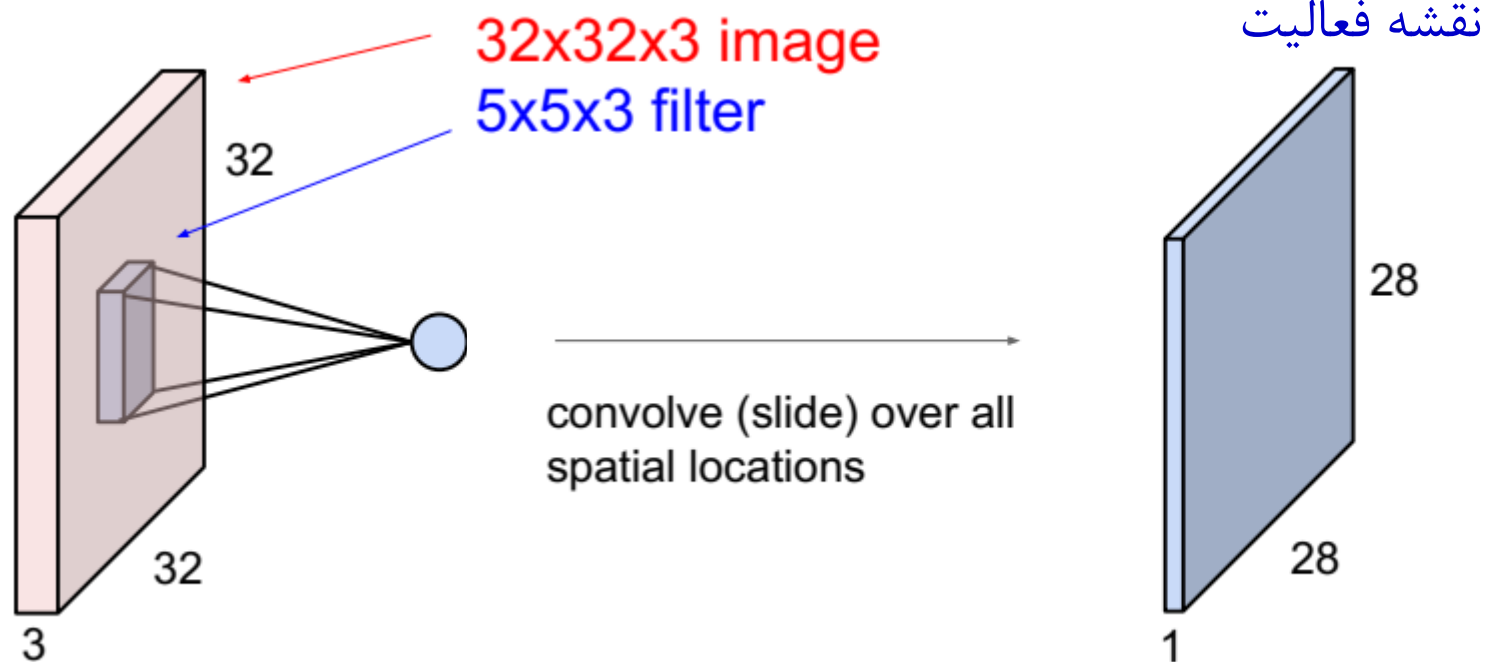
خروجی برابر با ضرب داخلی بین فیلتر و همسایگی مربوطه برای هر پیکسل است که معادل با ۷۵ ضرب و جمع است

$$w^T x + b$$

# لایه کانولوشنی

خروجی لایه کانولوشنی حاصل فیلتر کردن  
ماتریس ورودی با فیلتر مربوطه است که به  
صورت مکانی بر روی آن لغزانده می شود

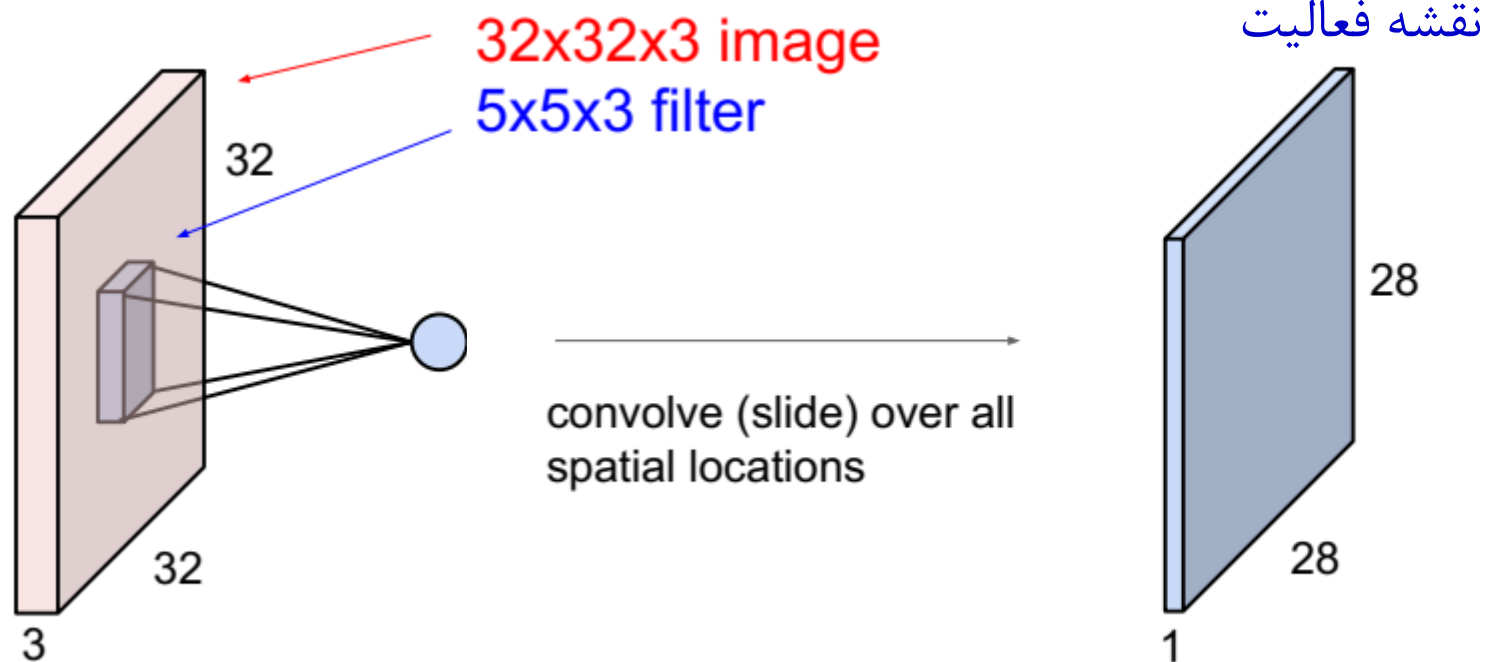
ورودی یک ماتریس  
۳ بعدی است



# لایه کانولوشنی

- البته یک فیلتر می‌تواند تنها یک مشخصه از تصویر را استخراج نماید

ورودی یک ماتریس  
۳ بعدی است

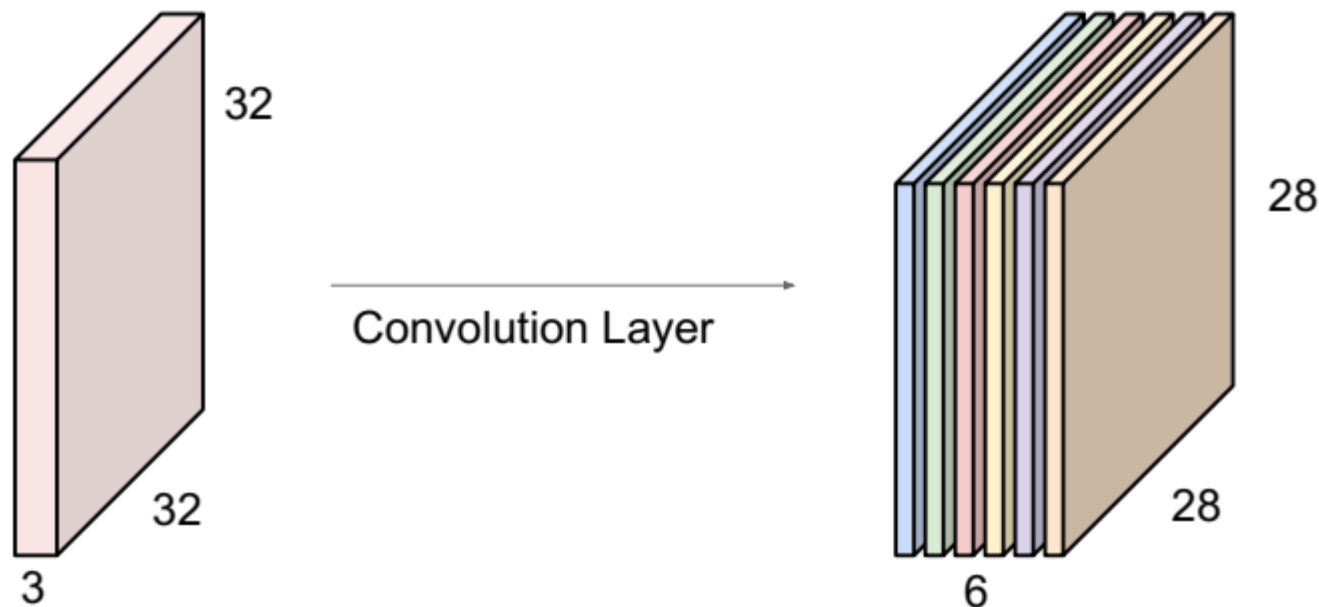


# لایه کانولوشنی

- البته یک فیلتر می تواند تنها یک مشخصه از تصویر را استخراج نماید

ورودی یک ماتریس  
۳ بعدی است

نقشه های فعالیت



# لایه کانولوشنی در Keras

```
keras.layers.Conv2D(filters, kernel_size, strides=(1, 1), padding='valid',  
                    data_format=None, dilation_rate=(1, 1), activation=None,  
                    use_bias=True, kernel_initializer='glorot_uniform',  
                    bias_initializer='zeros', kernel_regularizer=None,  
                    bias_regularizer=None, activity_regularizer=None,  
                    kernel_constraint=None, bias_constraint=None)
```

**filters:** Integer, the dimensionality of the output space

**kernel\_size:** Specifying the height and width of the 2D convolution window

**activation:** Activation function to use. If you don't specify anything, no activation is applied (see `keras.activations`)