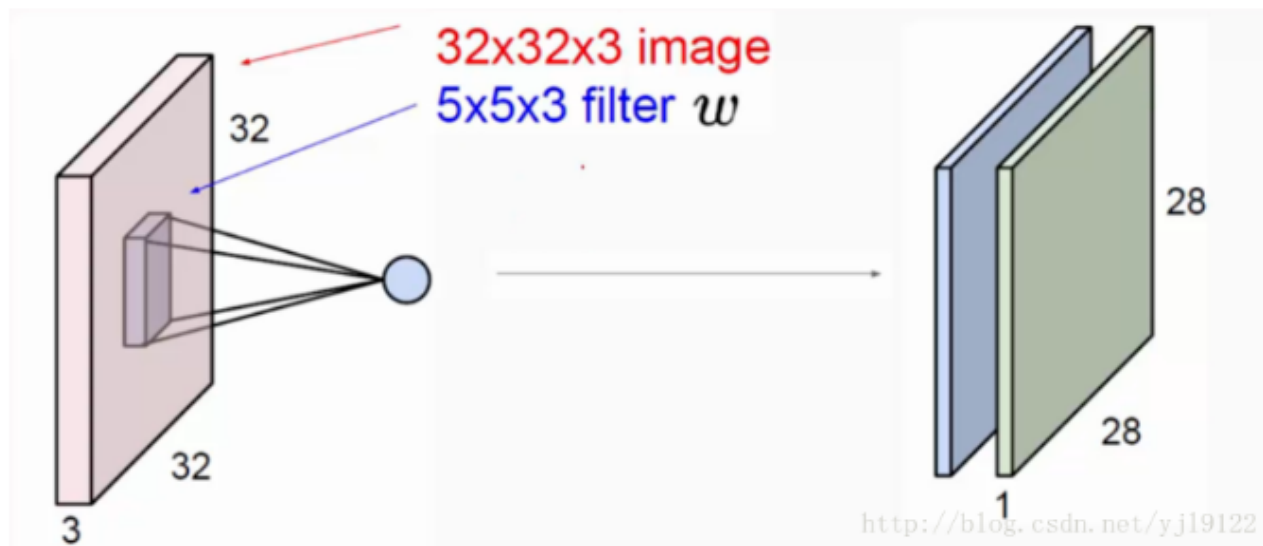


## 卷积神经网络CNN

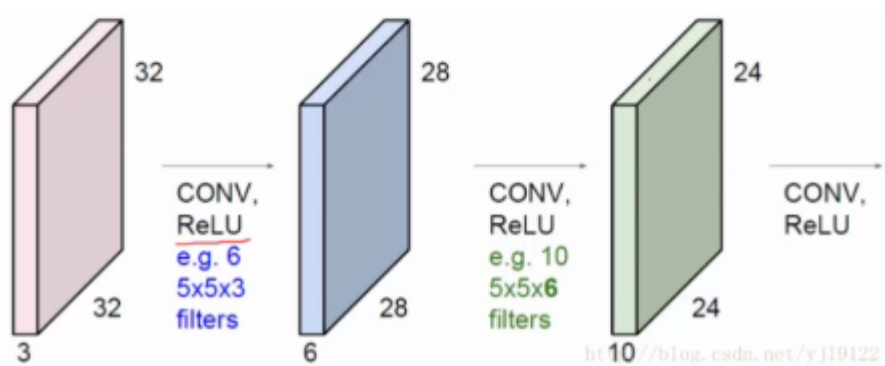
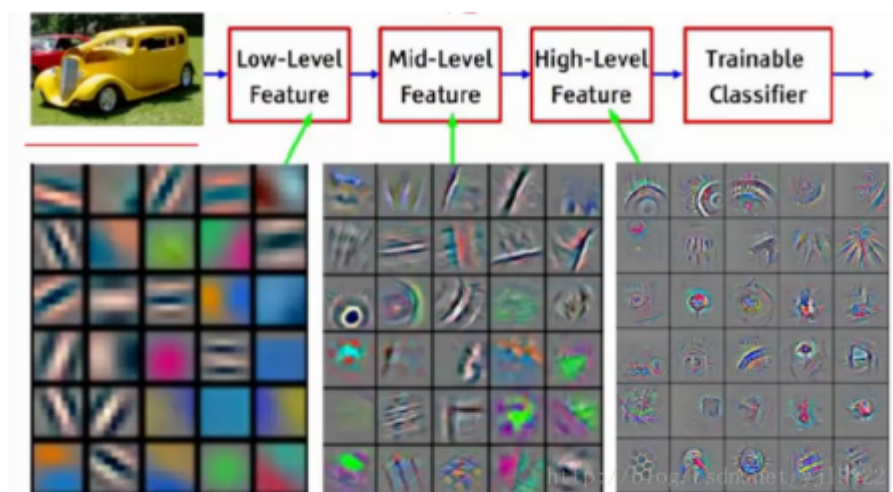
- 卷积神经网络(CNN)由输入层、卷积层、激活函数、池化层、全连接层、输出层组成。
- Input -> Conv -> ReLU -> Pool -> Fc -> Output

### 卷积层

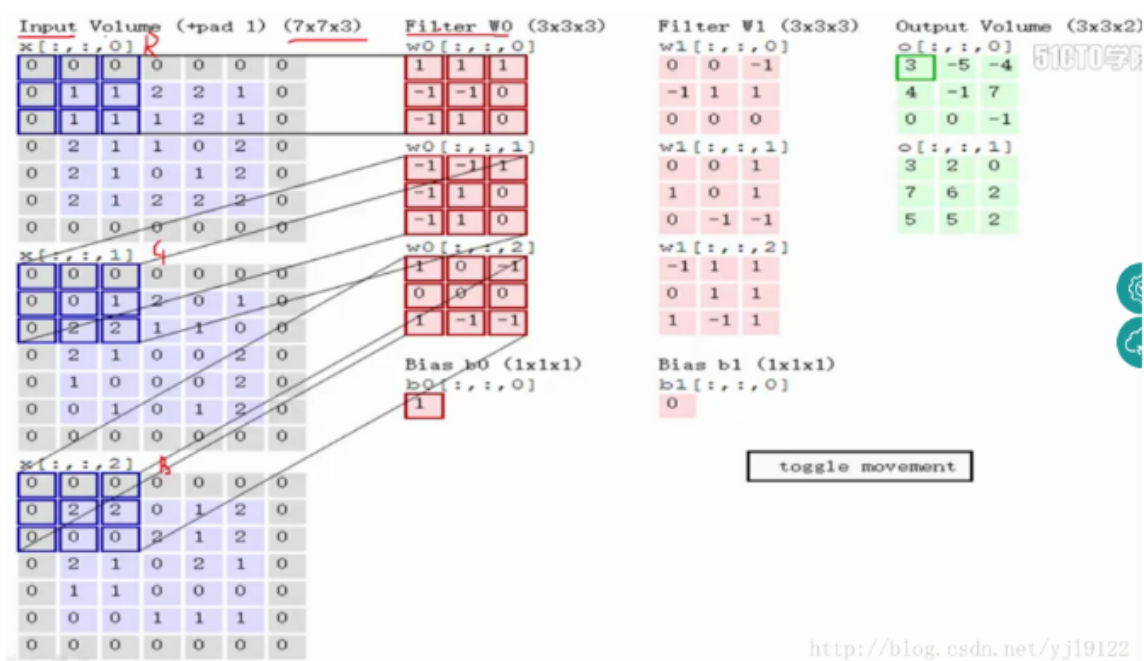
- 用来进行特征提取。



- 输入图像是  $32 \times 32 \times 3$ ，3是它的深度，卷积层是一个  $5 \times 5 \times 3$  的卷积核，卷积核的深度必须和输入图像的深度相同，通过一个卷积核与输入图像的卷积可以得到一个  $28 \times 28 \times 1$  的特征图。
- 使用多层卷积得到更深层次的特征图。



- 卷积的图解过程。



- 特性：权值共享。就是用相同的卷积核去扫描一张图片的每个位置。

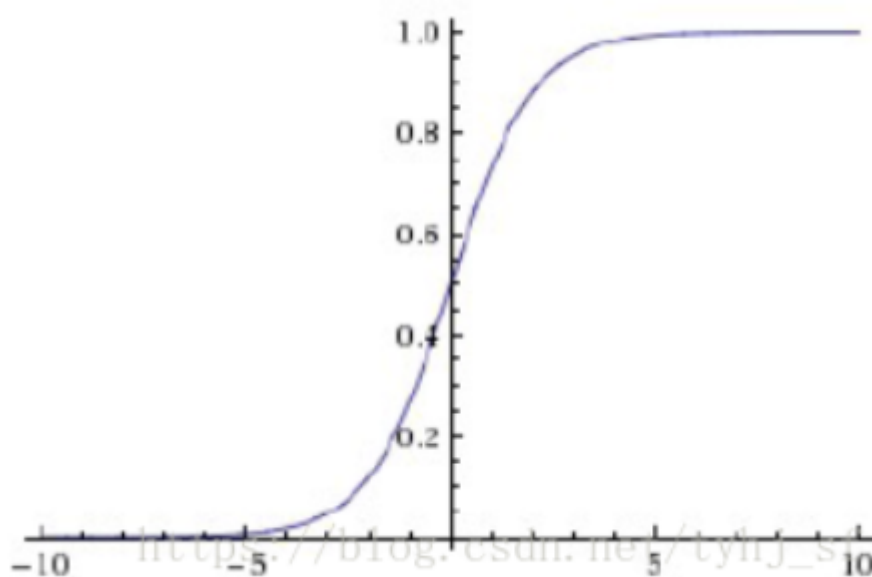
## 激活函数

- 问题：如果输入变化很小，导致输出结构发生截然不同的结果，使结果要是0-1之间的任何数。
- 激活函数是用来加入非线性因素，因为线性模型的表达力不够。
- Tanh在特征相差明显时的效果会很好，在循环过程中会不断扩大特征效果显示出来，但是，在特征相差比较复杂或是相差不是特别大时，需要更细微的分类判断的时候，sigmoid效果就好了。
- sigmoid 和 Tanh作为激活函数，一定要注意一定要对 input 进行归一化，否则激活后的值都会进入平坦区，使隐层的输出全部趋同，但是 ReLU 并不需要输入归一化来防止它们达到饱和。

### 常用的激活函数

- 激活函数应该具有的性质：
  - 非线性。线性激活层对于深层神经网络没有作用，因为其作用以后仍然是输入的各种线性变换。
  - 连续可微。梯度下降法的要求。
  - 范围最好不饱和。当有饱和的区间段时，若系统优化进入该段，梯度近似为0，网络的学习就会停止。
  - 单调性。当激活函数是单调时，单层神经网络的误差函数是凸的，好优化。
  - 在原点处近似线性。这样当权值初始化为接近0的随机值时，网络可以学习的较快，不用调节网络的初始值。
- Sigmoid函数

$$f(z) = \frac{1}{1 + e^{-z}}$$



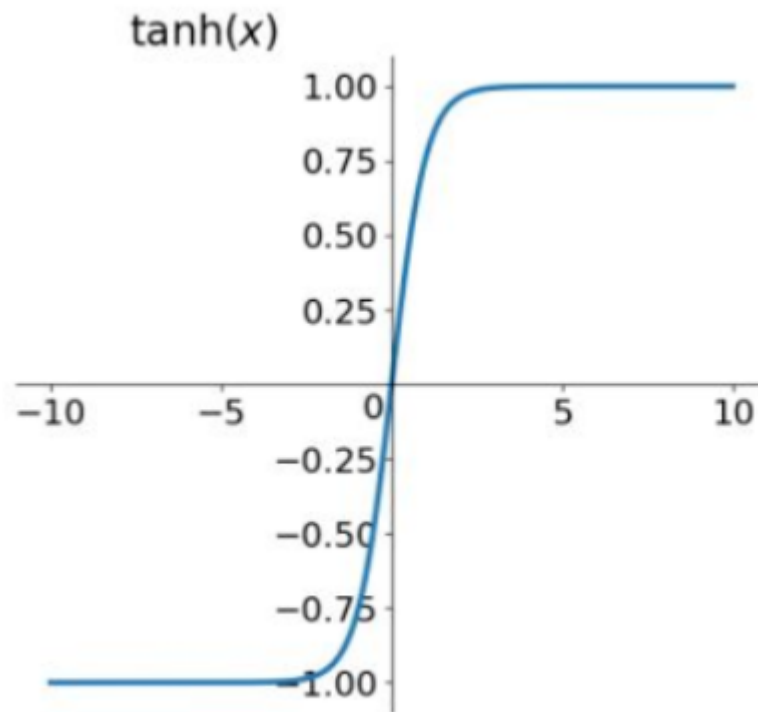
- 特点：它能够把输入的连续实值变换为0和1之间的输出，特别的，如果是非常大的负数，那么输出就是0；如果是非常大的正数，输出就是1。
- 目前已被淘汰
- 缺点：
  - **饱和时梯度值非常小。**由于BP算法反向传播时后层的梯度是以乘性方式传递到前层，因此当层数较多的时候，传到前层的梯度就会非常小，网络权值得不到有效的更新，即梯度耗散。如果该层的权

值初始化使得 $f(x)$ 处于饱和状态时，网络基本上权值无法更新。

- 输出值不是以0为中心值。

- Tanh函数

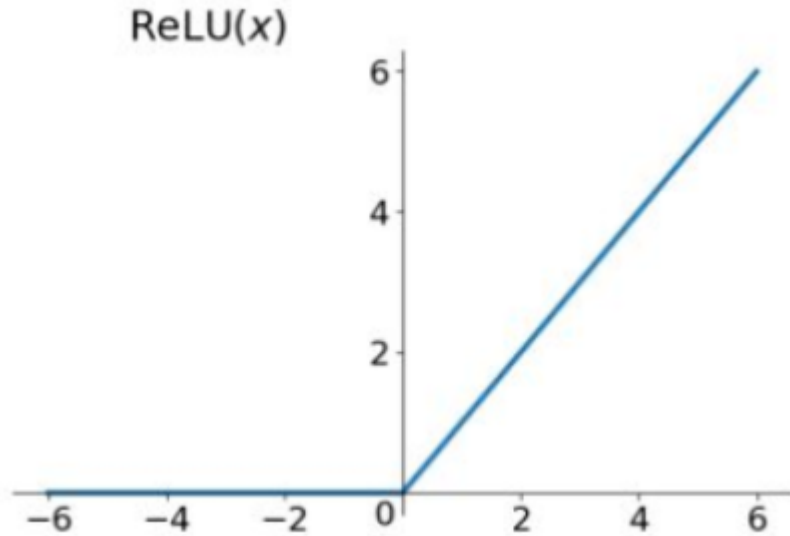
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



- 解决了不是以0为中心值得问题，但是仍然具有梯度消失问题。

- ReLU函数

$$Relu = \max(0, x)$$



○ 优点:

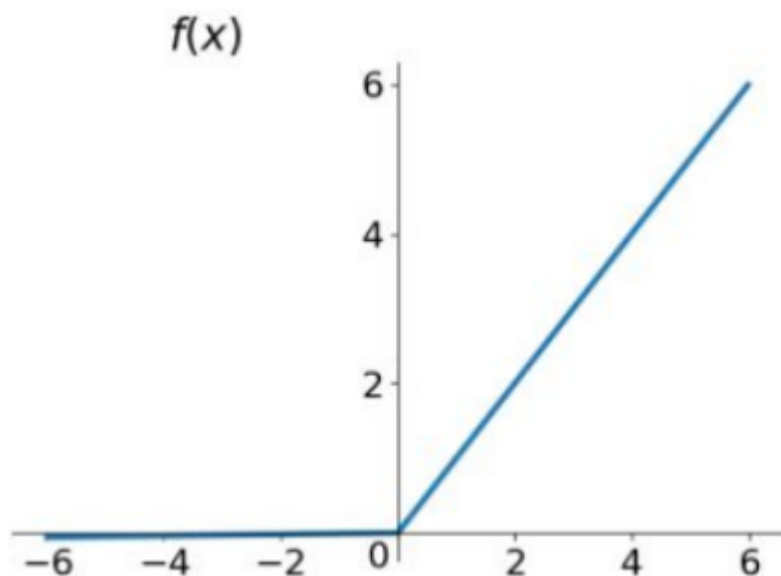
- $x > 0$ 时, 梯度恒为1, 无梯度消失问题, 收敛快。
- 增大了网络的稀疏性。当 $x < 0$ 时, 该层的输出为0, 训练完后为0的神经元越多, 稀疏性越大, 提取出来的特征更具有代表性, 泛化能力越强。
- 运算量很小。

○ 缺点:

- 如果后层的某一个梯度特别大, 导致W更新以后变得特别大导致该层的输入  $< 0$ , 输出为0, 这时该层就会死, 没有更新。当学习率比较大时, 可能会有40%的神经元都会在训练开始就死。

• Leaky ReLU函数

$$f(x) = \begin{cases} 1, (x < 0) \\ \alpha x + 1 (x \geq 0) \end{cases}$$



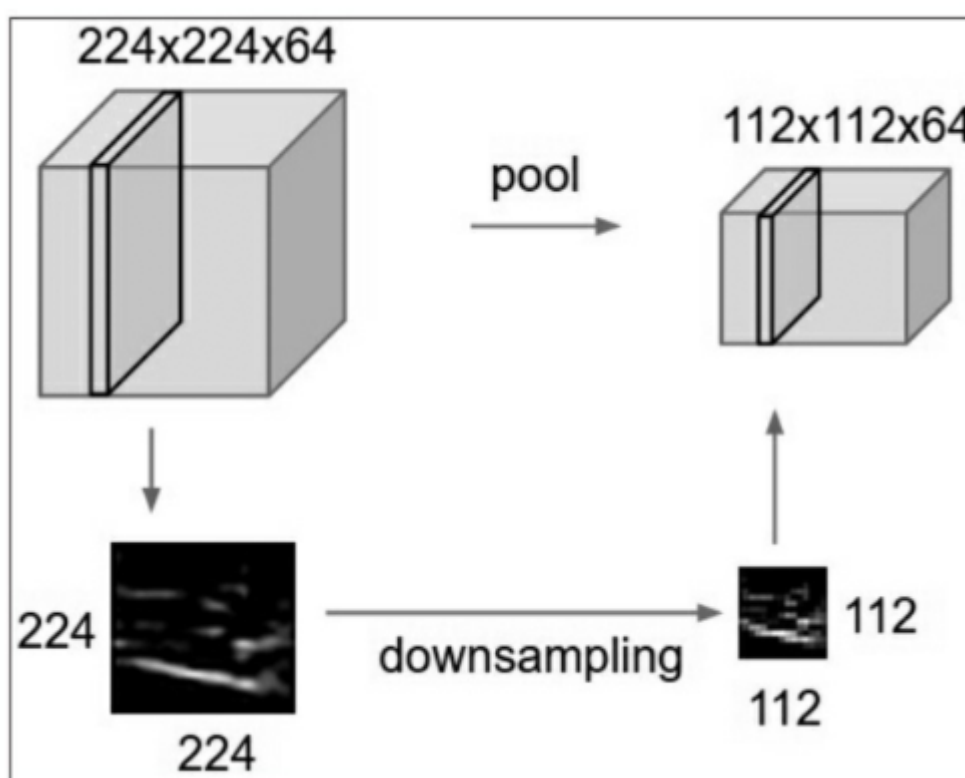
- 改善了ReLU的死亡特性，但是也损失了一部分稀疏性，且增加了一个超参数。
- Maxout函数

$$f(x) = \max(w_1^T x + b_1, w_2^T x + b_2)$$

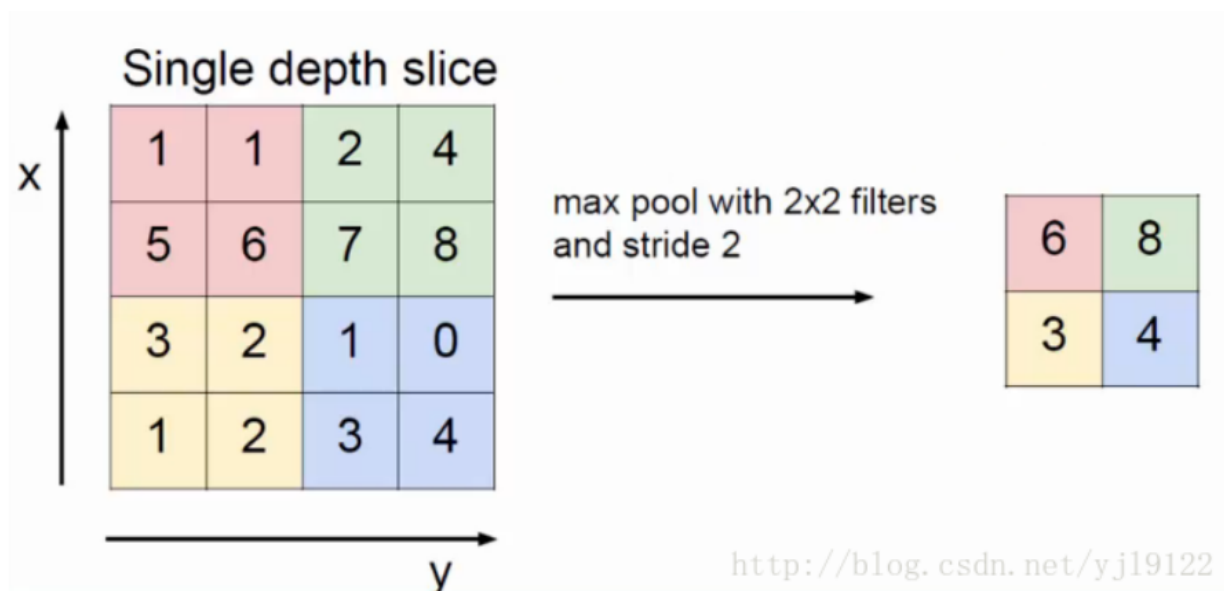
- 泛化了ReLU和Leaky ReLU，改善了死亡特性，但是损失了部分稀疏性，每个非线性函数增加了两倍的参数。

## 池化层

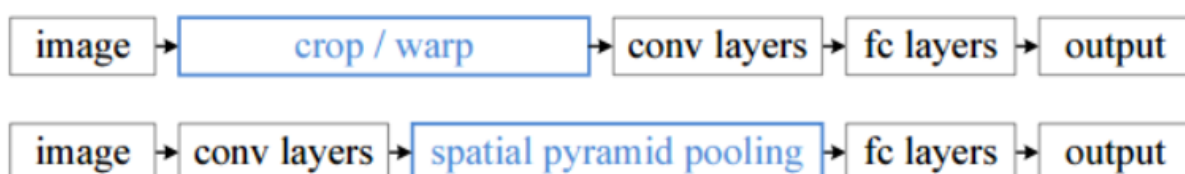
- 对于输入的特征图进行压缩，一方面使特征图变小，简化网络计算复杂度；另一方面进行特征压缩，提取主要特征。也可以避免过拟合现象。



- 一般池化有两种：作用于图像中不重合的区域，stride=sizeX。
- 最大池化。

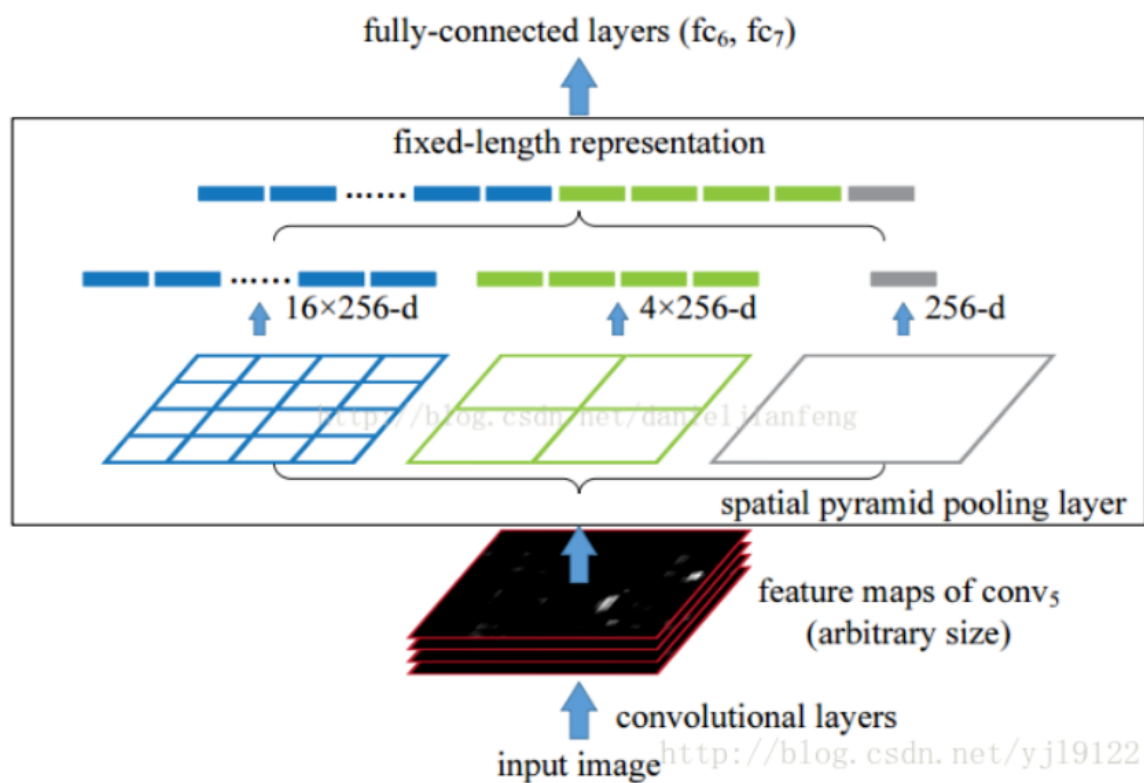


- 一个2\*2的filter，在每一个区域中寻找最大值。
- 平均池化。
  - 对每一个2\*2的区域元素求和，再除以4，得到主要特征。
- 重叠池化
  - 相邻池化窗口之间会有重叠区域，此时sizeX>stride。
- 空间金字塔池化
  - 空间金字塔池化可以把任何尺度的图像的卷积特征转化成相同维度，这不仅可以让CNN处理任意尺度的图像，还能避免cropping和warping操作，导致一些信息的丢失，具有非常重要的意义。
  - 一般的CNN都需要输入图像的大小是固定的，这是因为全连接层的输入需要固定输入维度，但在卷积操作是没有对图像尺度有限制，空间金字塔池化，先让图像进行卷积操作，然后转化成维度相同的特征输入到全连接层，这个可以把CNN扩展到任意大小的图像。



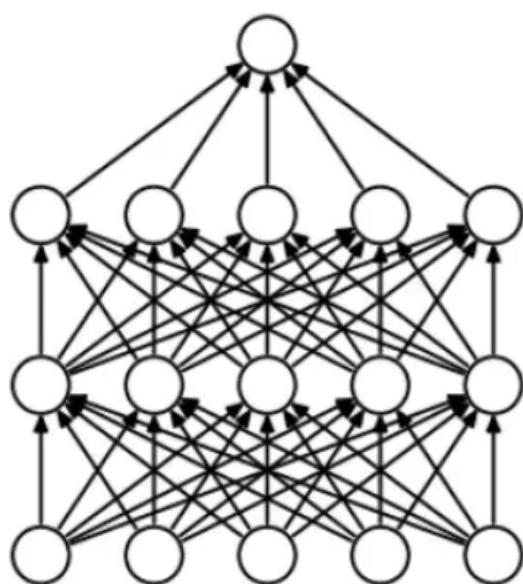
最下面的是空间金字塔池化。

- 它一个pooling变成了多个scale的pooling。用不同大小池化窗口作用于卷积特征，我们可以得到1X1,2X2,4X4的池化结果，由于conv5中共有256个过滤器，所以得到1个256维的特征，4个256个特征，以及16个256维的特征，然后把这21个256维特征链接起来输入全连接层，通过这种方式把不同大小的图像转化成相同维度的特征。

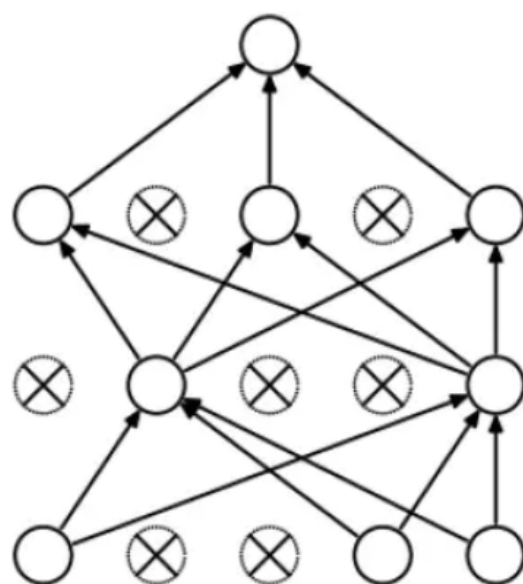


## 全连接层

- 连接所有的特征，将输出值送给分类器（如softmax分类器）。
- 将特征值转换为类别概率。
- 在进入全连接层之前，使用全局平均池化能够有效地降低过拟合。
- Dropout

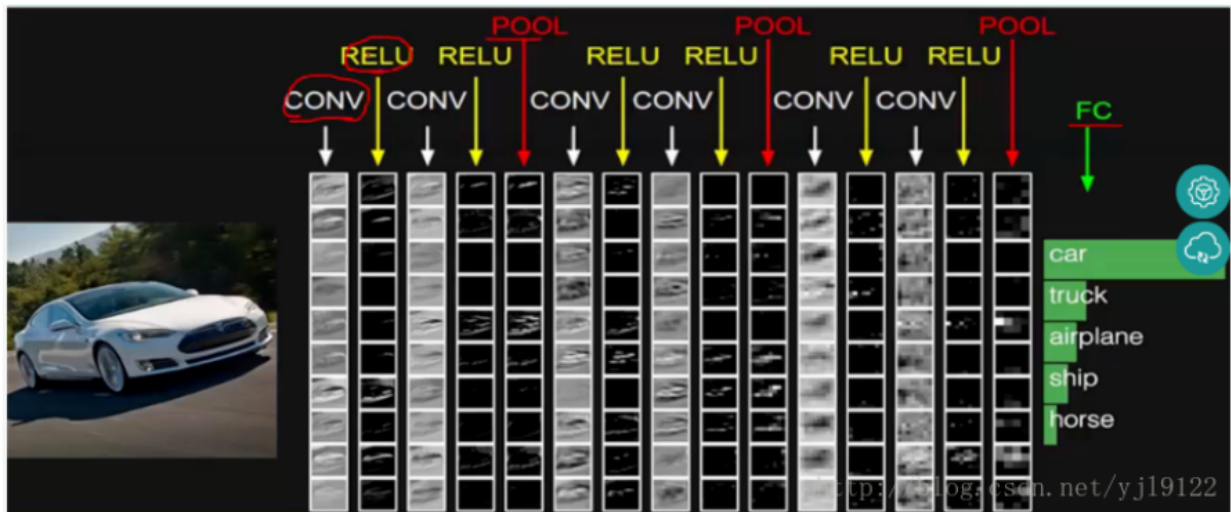


(a) Standard Neural Net



(b) After applying dropout.





### 防止过拟合的方法

- 提前终止（当验证集上的效果变差的时候）
- L1和L2正则化加权
- soft weight sharing
- dropout