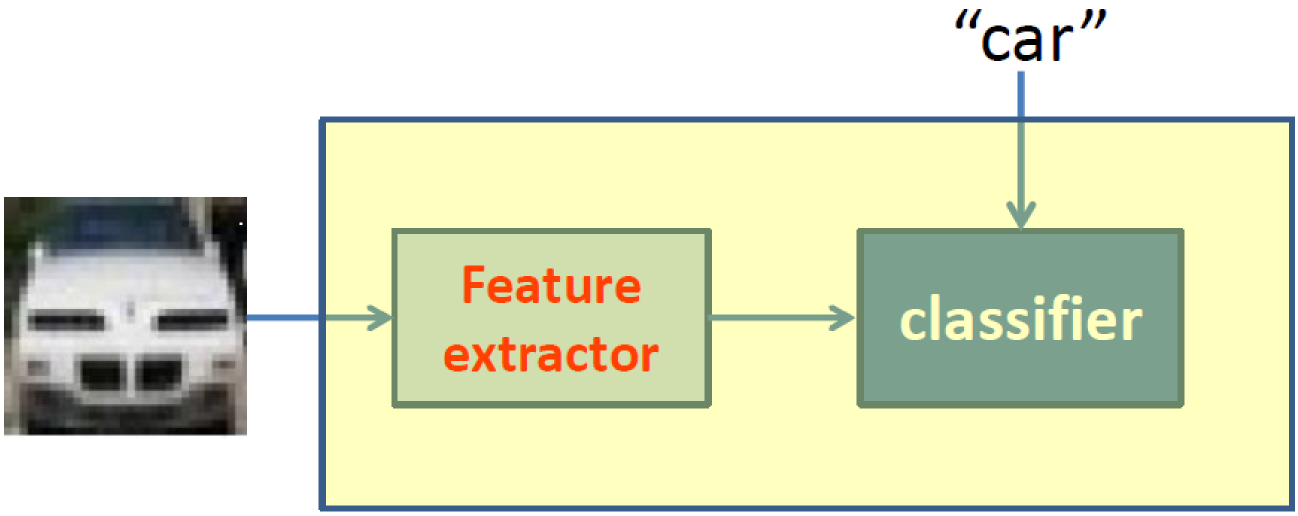
INT301 W8

Key Ideas of Deep Learning:

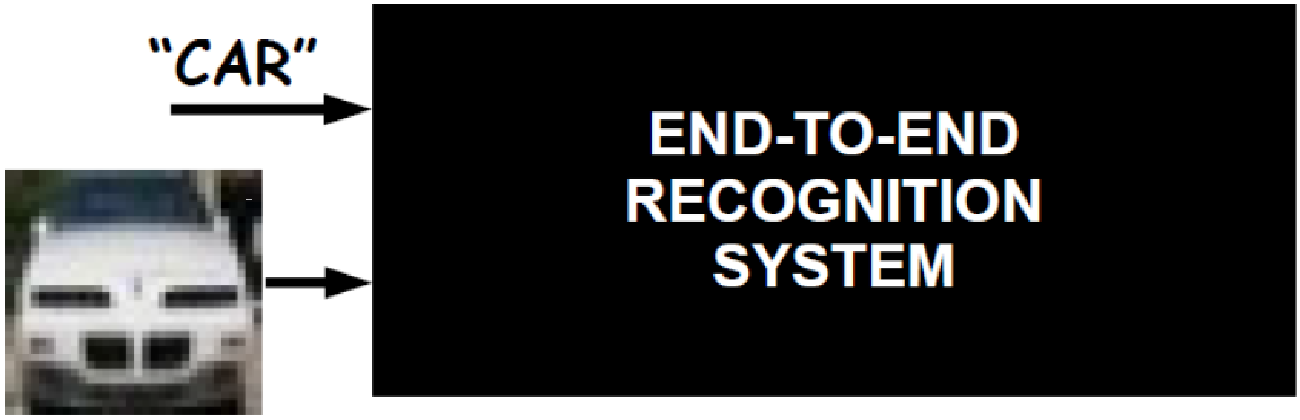
- Deal with non linear system
- Learn feature from data (or big data)
- Build feature hierarchies (function composition)
- End to end learning

End-to-end Object Recognition



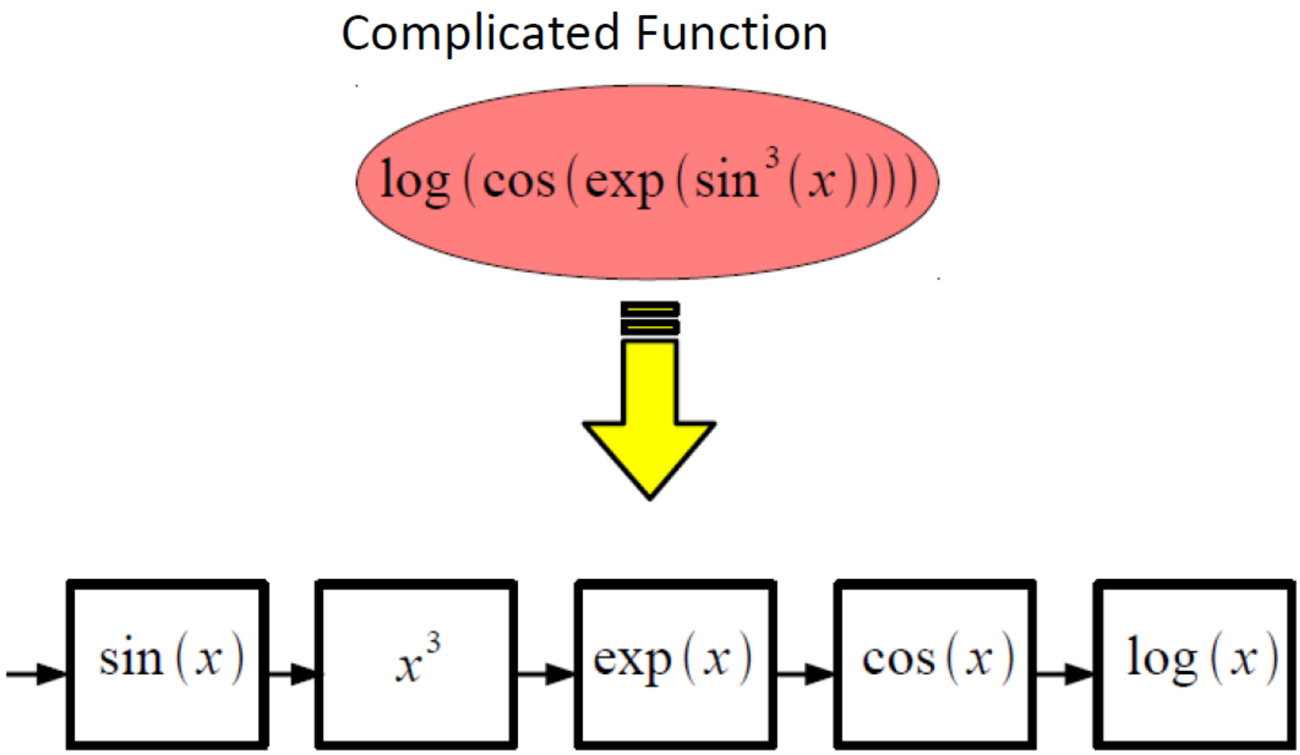
How to use data to optimize features for the given task?

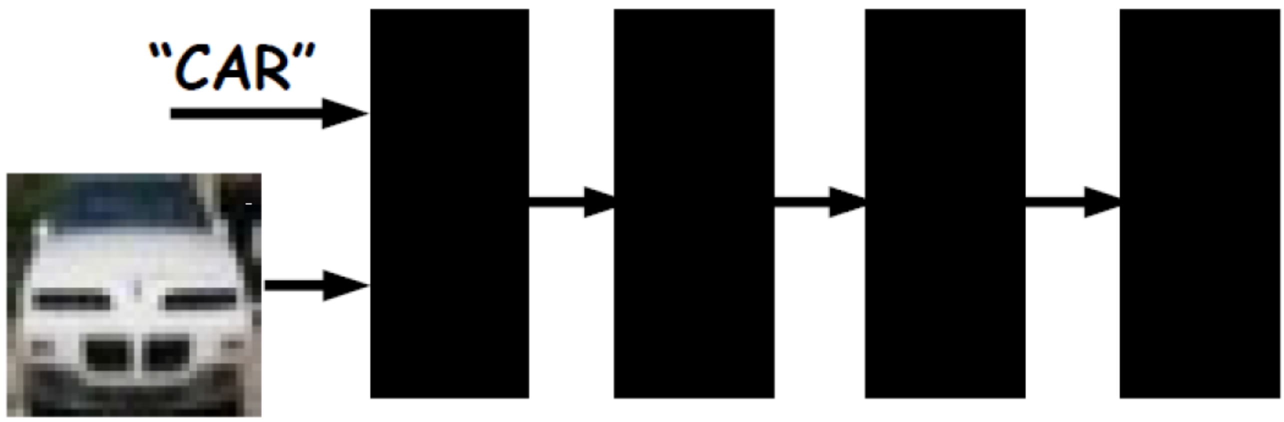
- Everything becomes adaptive.
- No distinction between feature extractor and classifier.
- Big non-linear system trained from raw pixels to labels



By combining simple building blocks, we can make more and more complex systems (highly non-linear system).

Building A Complicated Function





Each black box can have trainable parameters.

Their composition makes a highly non-linear system.

System produces a hierarchy of features

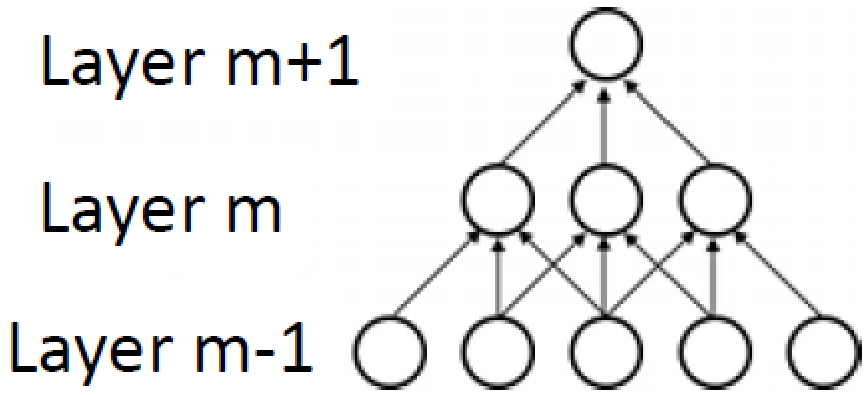
Convolutional NN

Convolutional Neural Networks is extension of traditional Multi layer Perceptron , based on 3 ideas:

- Local receive fields (局部感受野)
- Shared weights (权值共享就是说，给一张输入图片，用一个卷积核去扫这张图，卷积核里面的数就叫权重，这张图每个位置是被同样的卷积核扫的，所以权重是一样的，也就是共享)
- Spatial / temporal sub-sampling (下采样)

Sparse Connectivity

稀疏连接，指对一个视觉区域内极小的一部分敏感，而对其他部分则可以视而不见。



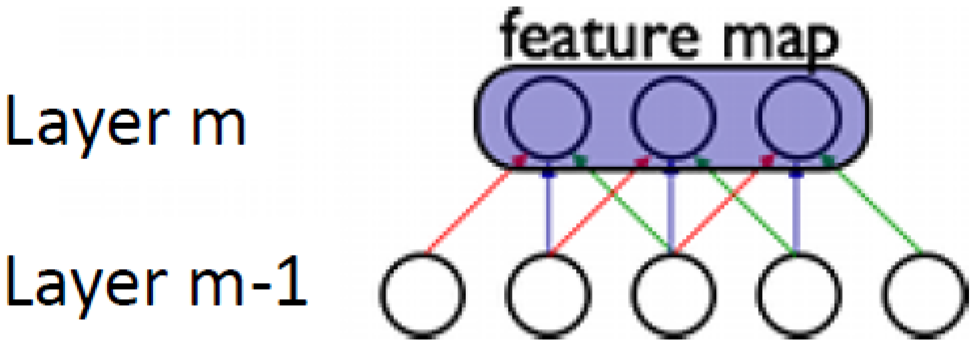
The inputs of hidden units in layer m are from a subset of units in layer m-1 , units that have spatially contiguous receptive fields.

假如 m-1 层是视网膜 (input retina)，上图中 layer m 的每个 unit 只接受 3 个 input retina 的 unit (只有 3 个箭头指入)。但 input retina 有 5 个 units，那么其他不相连 units 的变化就和 layer m 的 unit 不相关。这就是稀疏连接。

该架构确保学习的 'filters' 对 spatially local input pattern 产生最强的响应。

Shared Weights

在 CNN 中，每个 filter 在整个视野中复制 (卷积操作)，这些 replicated units 都共享相同的权重 (weight and bias)，形成一个 feature map (对图像进行卷积，出来的结果就是 feature map，有多少个 filter 就有多少个 feature map)。



Replicating units 以这种方式检测特征，无论它们在视野中的位置如何。此外，权重共享通过大大减少学习的自由参数的数量来提高学习效率。对模型的约束使 CNN 能够更好地泛化视觉问题。

Convolution

If we denote the k-th feature map at a given layer as h^k, whose filters are determined by the weights w^k and bias b^k, then the feature map is obtained using convolution as follows (for tanh non-linearities):

$$h_{ij}^{k} = \tanh\left(\left(W^{k} * x\right)_{ij} + b_{k}\right)$$

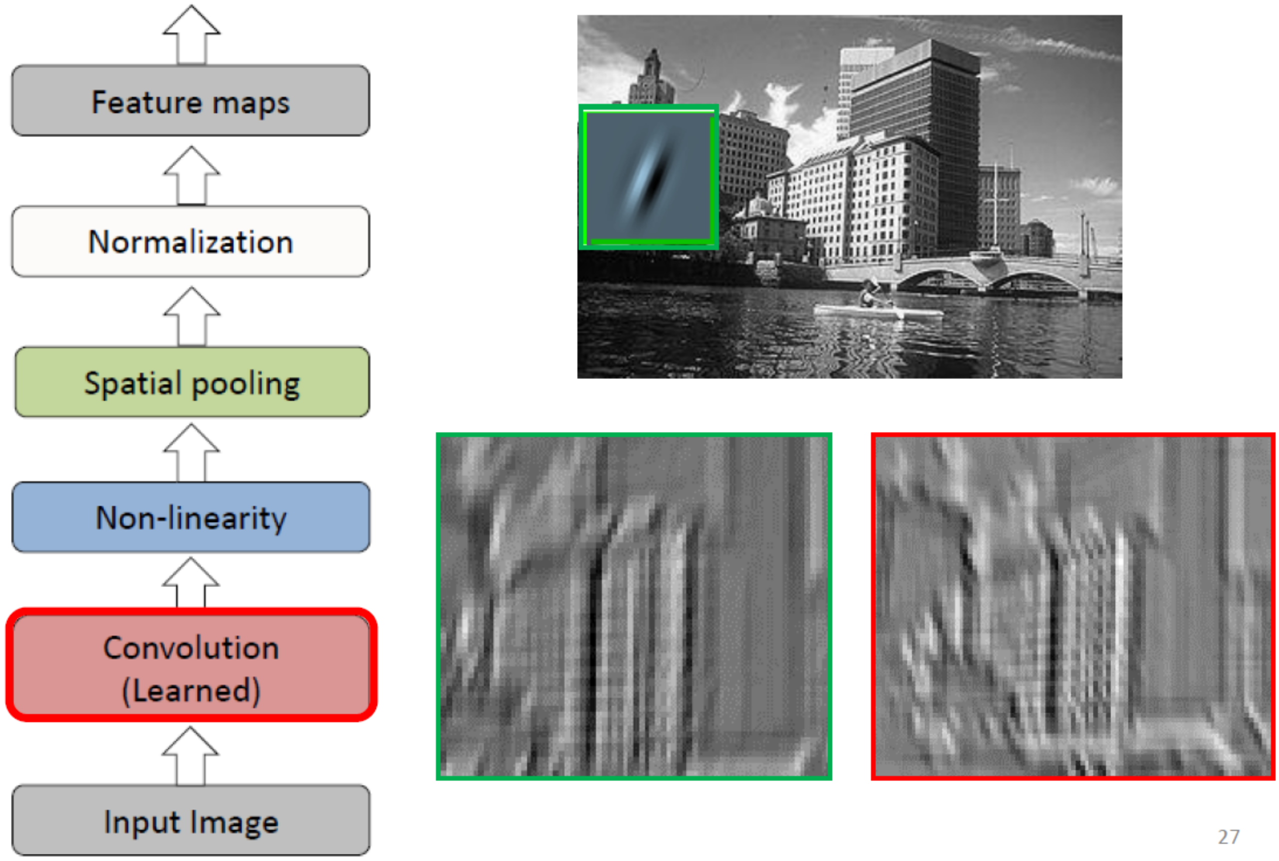
Suppose that we have some N × N square neuron layer which is followed by our convolutional layer. If we use an m × m filter ω, our convolutional layer output will be of size (N-m+1) × (N-m+1) (如果有步长和padding，就是 (N-m+2×padding)/步长+1).

In order to compute the pre-nonlinearity input to some unit x_{ij}^{ℓ} in our layer, we need to sum up the contributions (weighted by the filter components) from the previous layer cells:

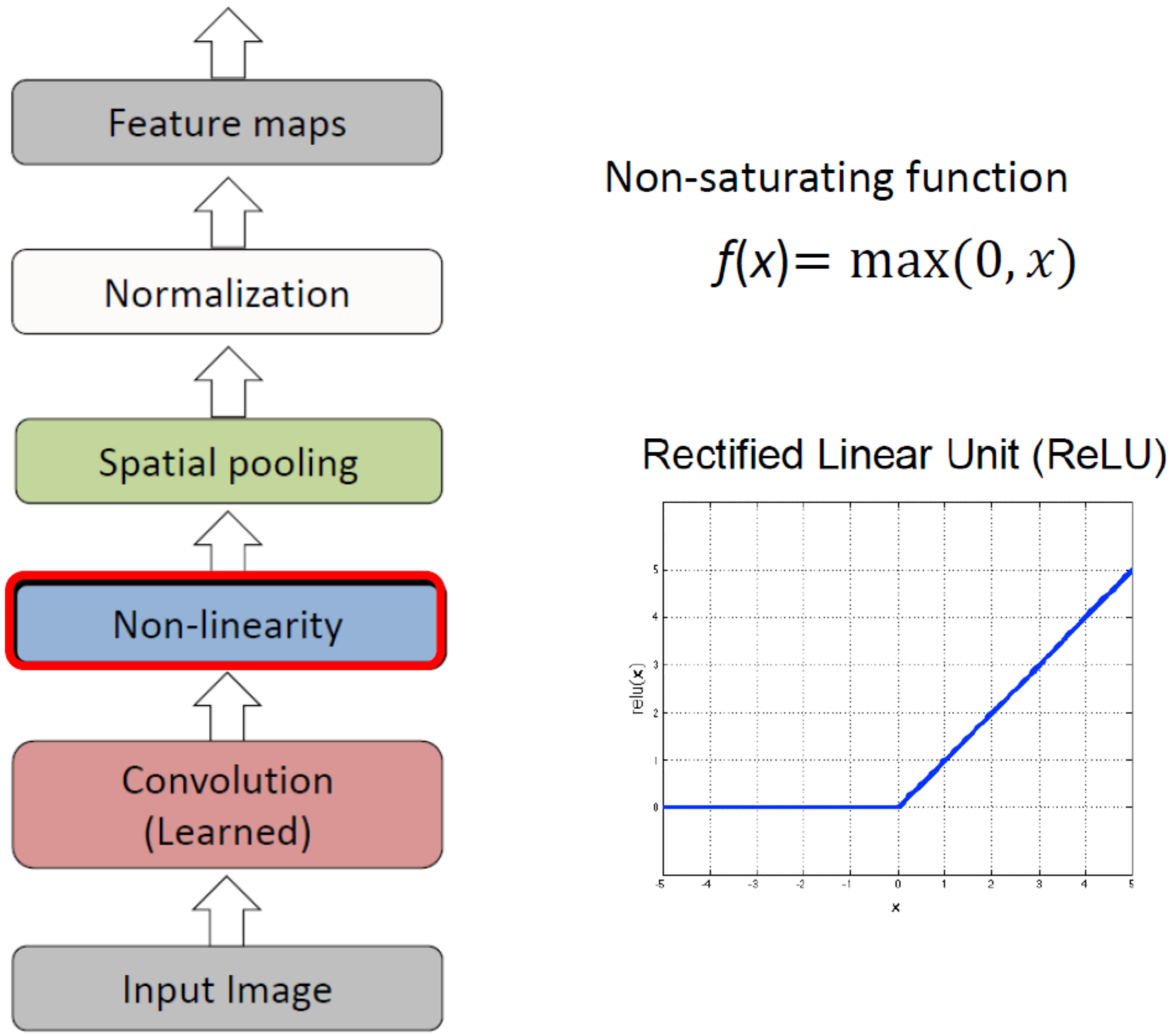
$$x_{ij}^{\ell} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega_{ab} y_{(i+a)(j+b)}^{\ell-1}$$

注：上式是卷积操作的公式，先相乘，再累加。

假如 input 为 3 x 3，filter 为 2 x 2，卷积后得到 2 x 2 的 feature map。上式中，l 代表 feature map 这层，l-1 代表 input那层。i 和 j 代表 feature map 中某一像素的坐标，假如这里我们取 2 x 2 feature map 坐上那点，则 i，j 都为 0。a 和 b 代表 filter 的坐标，y 是 input 的值。



Non-linearity



使用 ReLU 做激活函数的好处：

- 通过 ReLU 的向前和向后传递都只是一个简单的 if 语句
- 计算量少，不像 sigmoid 要算指数
- 在处理具有许多神经元的大型网络时，这种优势是巨大的，并且可以显着减少训练和评估时间

Sigmoid 激活函数容易饱和 (saturate)：

- sigmoid 导数不为零的区间相对狭窄
- 一旦 sigmoid 到达左侧或右侧区域，反向传播通过它几乎毫无意义，因为导数非常接近 0

ReLU 仅在输入小于 0 时饱和：

- 甚至可以使用 leaky ReLU 消除这种饱和

Pooling

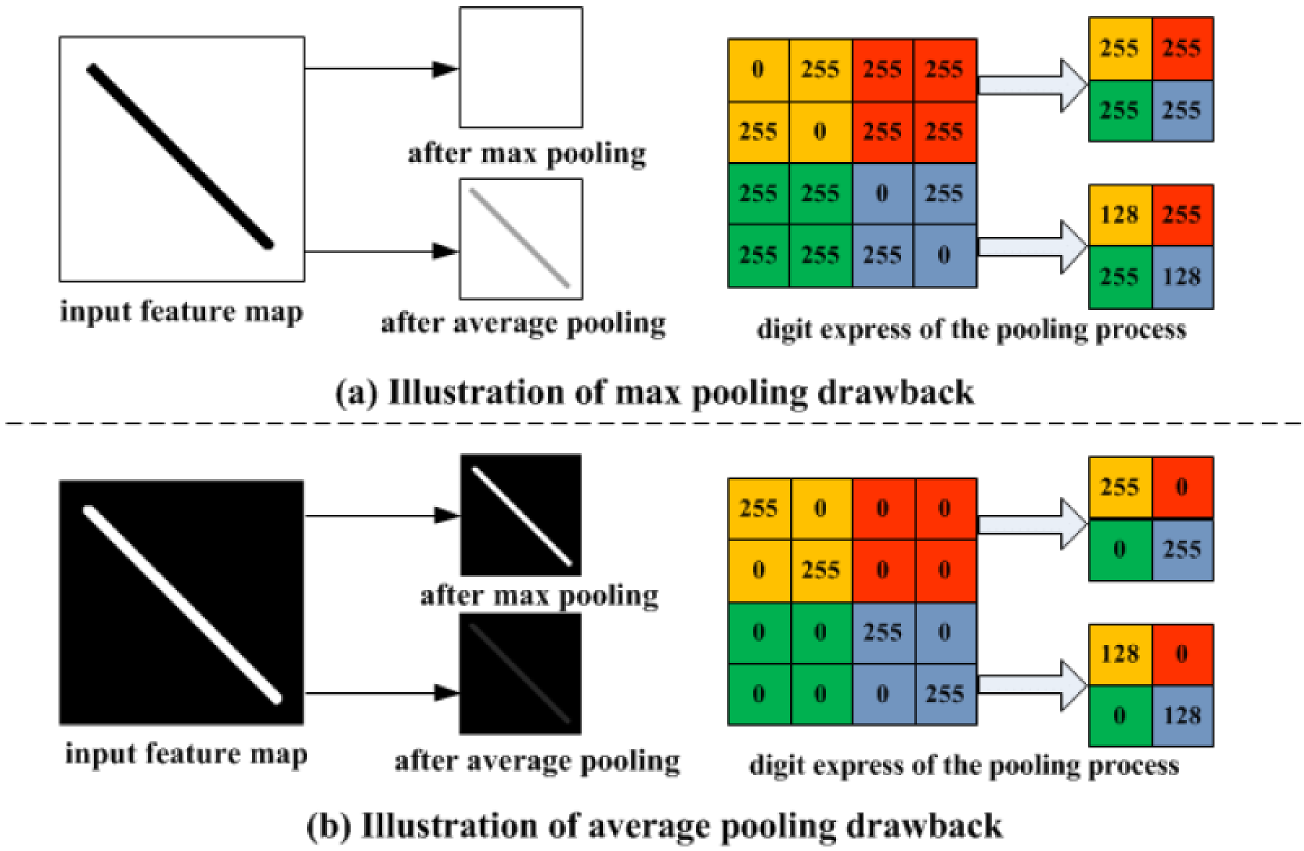
为了减少方差，池化层会计算图像某个区域上特定 feature 的最大值或平均值。 这将确保获得相同的结果，即使图像特征具有较小的平移。

Subsampling (pooling) Mechanism：

- Reduce spatial resolution - Reduce sensitivity to shift and distortion (降低空间分辨率-降低对偏移和失真的敏感度)

一般而言，池化的目标是将联合特征表示转换为新的、更可用的特征表示，以保留重要信息，同时丢弃不相关的细节。

实现对位置或照明条件变化的不变性、对杂波的鲁棒性 (robustness to clutter) 和表示的紧凑性 (compactness of representation)，都是池化的常见目标。



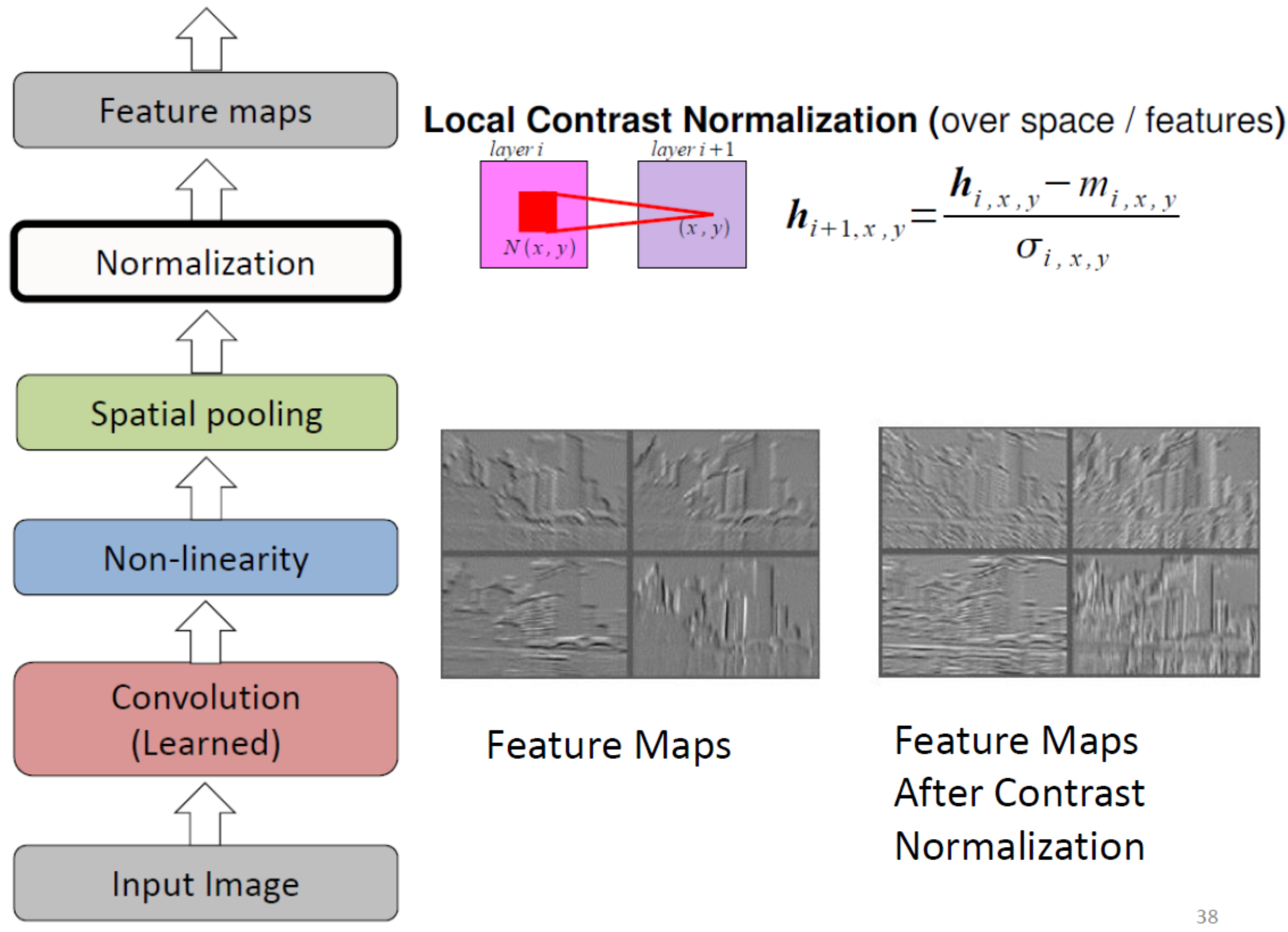
(a) Illustration of max pooling drawback

(b) Illustration of average pooling drawback

最大池化 (max pooling) 在视觉中很有用，原因有两个：

- 通过消除非最大值，它减少了上层的计算 (降低了维数)
- 它提供了一种平移不变性 (translation invariance) 的形式 (平移不会有太大区别)

Normalization

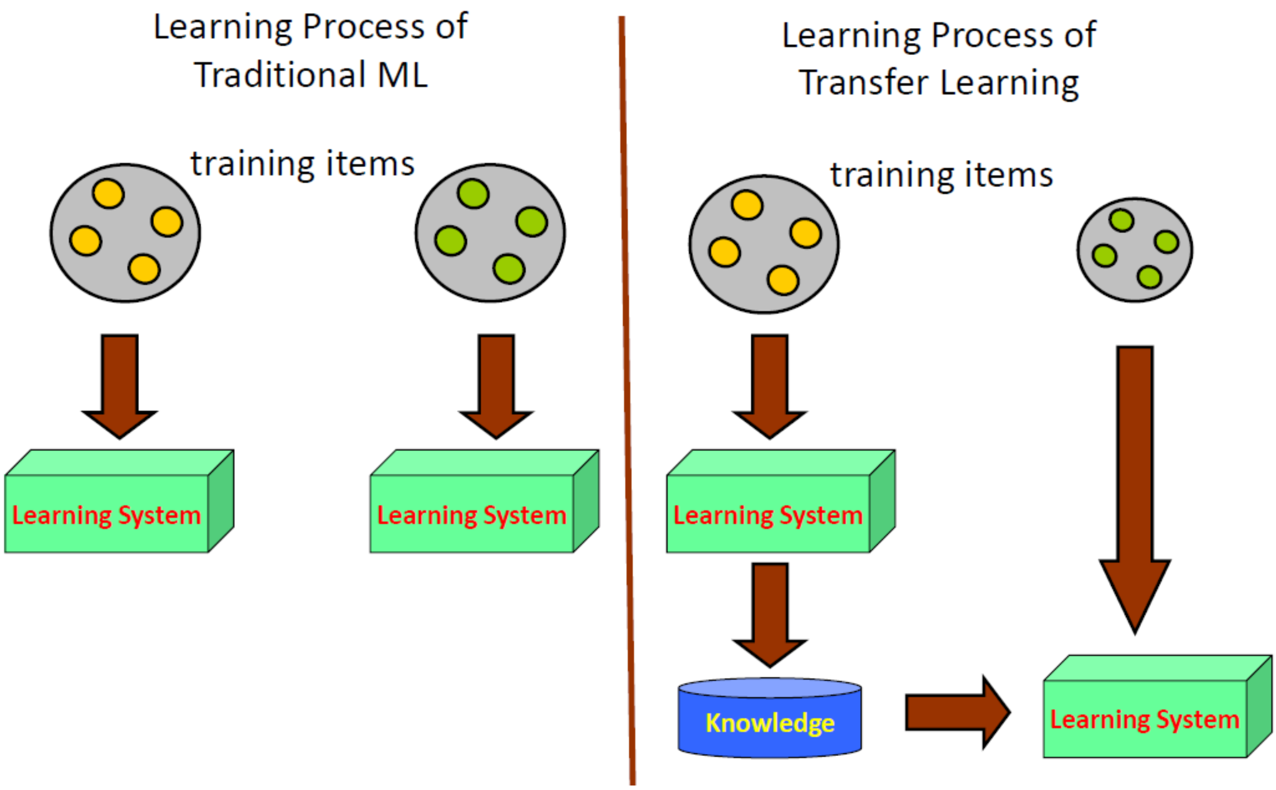


注：m 是 mean，σ 是标准差

Transfer Learning (TL)

把以前在其他领域（或任务）学到的东西，应用到新的领域（或任务）。

Traditional ML vs. TL



Deep CNN for Knowledge Transfer