深度学习及在图像目标检测中的应用

人识别物体是根据物体的特征来进行识别、分类。而计算机要识别所看到的物体，也需要根据特征来识别。计算机识别区别于人的识别主要在于人是一直在学习，并把学习的东西记忆在大脑。因此，计算机要想识别，也必须事先学习物体的特征。 特征学习，可以说是计算机视觉的核心。目标检测，实际上也就是物体识别，只有了解了物体的特征，才能更好的检测，因此，其核心也是特征学习。

在传统视觉领域，物体检测是一个非常热门的研究方向。受70年代落后的技术条件和有限应用场景的影响，物体检测直到上个世纪90年代才开始逐渐走入正轨。物体检测对于人眼来说并不困难，通过对图片中不同颜色、纹理、边缘模块的感知很容易定位出目标物体，但对于计算机来说，面对的是RGB像素矩阵，很难从图像中直接得到狗和猫这样的抽象概念并定位其位置，再加上物体姿态、光照和复杂背景混杂在一起，使得物体检测更加困难。

检测算法里面通常包含三个部分，第一个是检测窗口的选择， 第二个是特征的设计，第三个是分类器的设计。随着2001年Viola Jones提出基于Adaboost的人脸检测方法以来，物体检测算法经历了传统的人工设计特征+浅层分类器的框架，到基于大数据和深度神经网络的End-To-End的物体检测框架，物体检测一步步变得愈加成熟。

传统的方法怎么做？有什么缺点？

传统的目标检测方法一般分为三个阶段：

首先使用一些算法，如Selective Search算法，在给定的图像上进行区域选择，将那些潜在可能存在目标的区域选择出来，使用Selective Search可以有效的避免在整副图像上穷举。

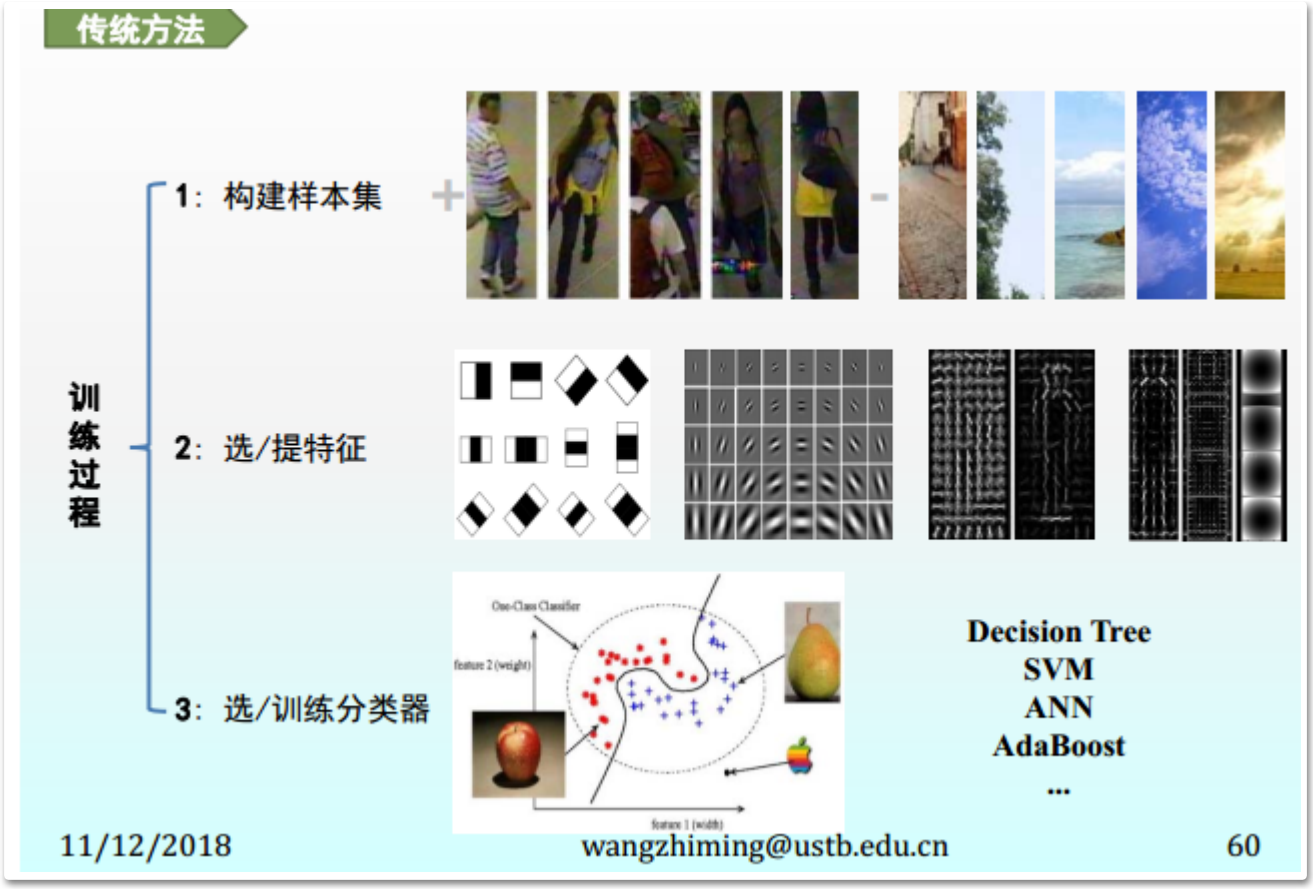
然后对这些选择得到的区域进行特征提取。如在传统方法中，人脸检测常用Harr特征，而行人与普通目标检测常用HOG特征等。但由于目标的形态、颜色、光照情况、背景的多样性，设计出相关鲁棒的特征是比较困难的。

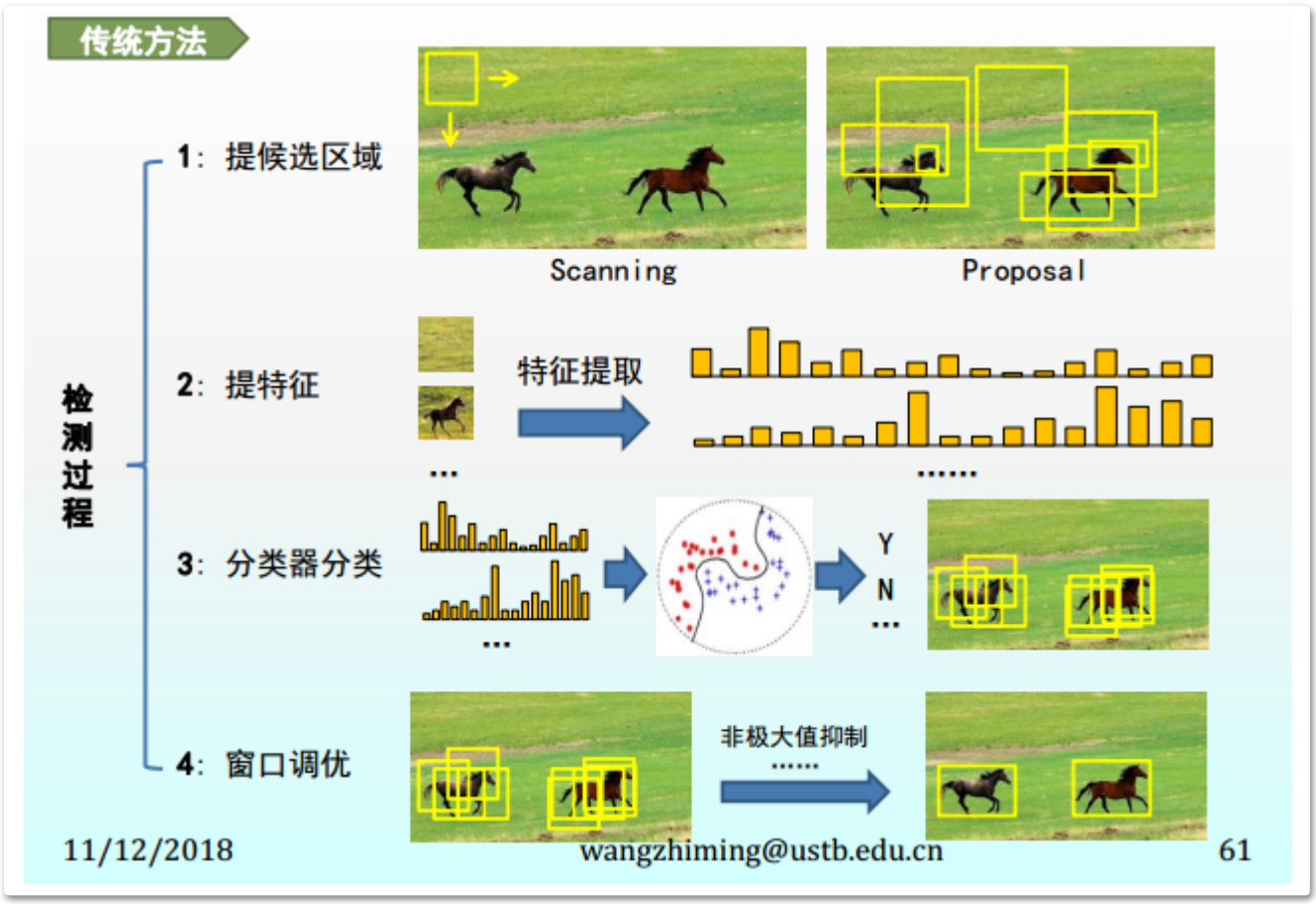
最后使用提取的特征对所选定的区域进行分类，因此特征的提取情况直接影响到最终检测结果的好坏。常用的分类器有支持向量机等。

传统算法大致可以分为目标实例检测与传统目标类别检测两类：

（1）目标实例检测问题通常利用模板和图像稳定的特征点，获得模板与场景中对象的对应关系，检测出目标实例。目标实例检测关注的只是具体目标本身，图像中的其余对象都是无关量。常用的算法有：SIFT算法，PCA-SIFT，SURF等，这些算法的共同点都是通过提取图像上的特征进行检测与匹配。

（2）传统目标类别检测则通过使用 AdaBoost算法框架、HOG特征和支持向量机等方法，根据选定的特征和分类器，检测出有限的几种类别。使用特征检测+分类框架做目标检测，实际上属于浅层次的智能方法，通常也分为连个阶段：训练阶段和识别应用阶段。这类方法额特征检测大多数采用计算机视觉桑的角点检测，特征直方图等。传统的目标类别检测局限性在于手工的特征选取，已经分类算法的高复杂度上，


传统方法
训练过程
1：构建样本集
2：选/提特征
3：选/训练分类器
Decision Tree
SVM
ANN
AdaBoost
...
11/12/2018
wangzhiming@ustb.edu.cn
60


传统方法
检测过程
1：提候选区域
Scanning
Proposal
2：提特征
特征提取
3：分类器分类
Y
N
4：窗口调优
非极大值抑制
11/12/2018
wangzhiming@ustb.edu.cn
61

传统方法的缺点：

总的来说，这些算法的目的都是在保证提取丰富、准确特征的前提下，快速地进行特征计算及预测。但传统算法提取的特征基本都是低层次、人工选定的特征，这些特征相对更直观、易理解，针对特定对象更有针对性，但不能很好地表达大量、多类目标。另外，传统方法如区域检测的时间复杂度很高，且很难针对性的进行目标的搜寻。另一方面，手工设计的特征对于物体的多样性变化、复杂物体等的鲁棒性并不强，且随着检测任务的推广，设计特征变得越来越复杂。

深度学习解决了传统方法的缺点？

传统的目标检测存在一些问题，如区域检测的时间复杂度很高，且很难针对性的进行目标的搜寻。另一方面，手工设计的特征对于物体的多样性变化、复杂物体等的鲁棒性并不强，且随着检测任务的推广，设计特征变得越来越复杂。

近年来，随着计算机硬件进步，深度学习方法得到了极大发展。针对目标检测问题，特别是上文中提到的几个传统方法中存在的问题，深度学习均有着很好的表现。

与传统方法相比，深度学习在分类精度上提高很多。起先，深度学习只是在分类上有非常明显的提升，之后也带动了检测这一块。从物体分类到物体检测，利用了深度学习比较强的feature的表达能力，可以进一步提高检测的精度。传统的物体检测方法因为其特征比较弱，所以每类都需要训练一个检测器。每个检测器都是针对特定的物体训练，如果有20类的话，就需要跑20次前向预测，相当于单次检测的20倍，作为一个2C端产品，时间消耗和精度性能使得传统方法检测的应用场景不是很多。

总结起来：深度学习由于其特殊的网络结构，能存储更多的特征信息，在效率和精度上都有很大的提高。另外就是深度学习利用模拟人脑的自学习过程，能够自动的学习到物体的特征，并用于后续的检测或者识别。

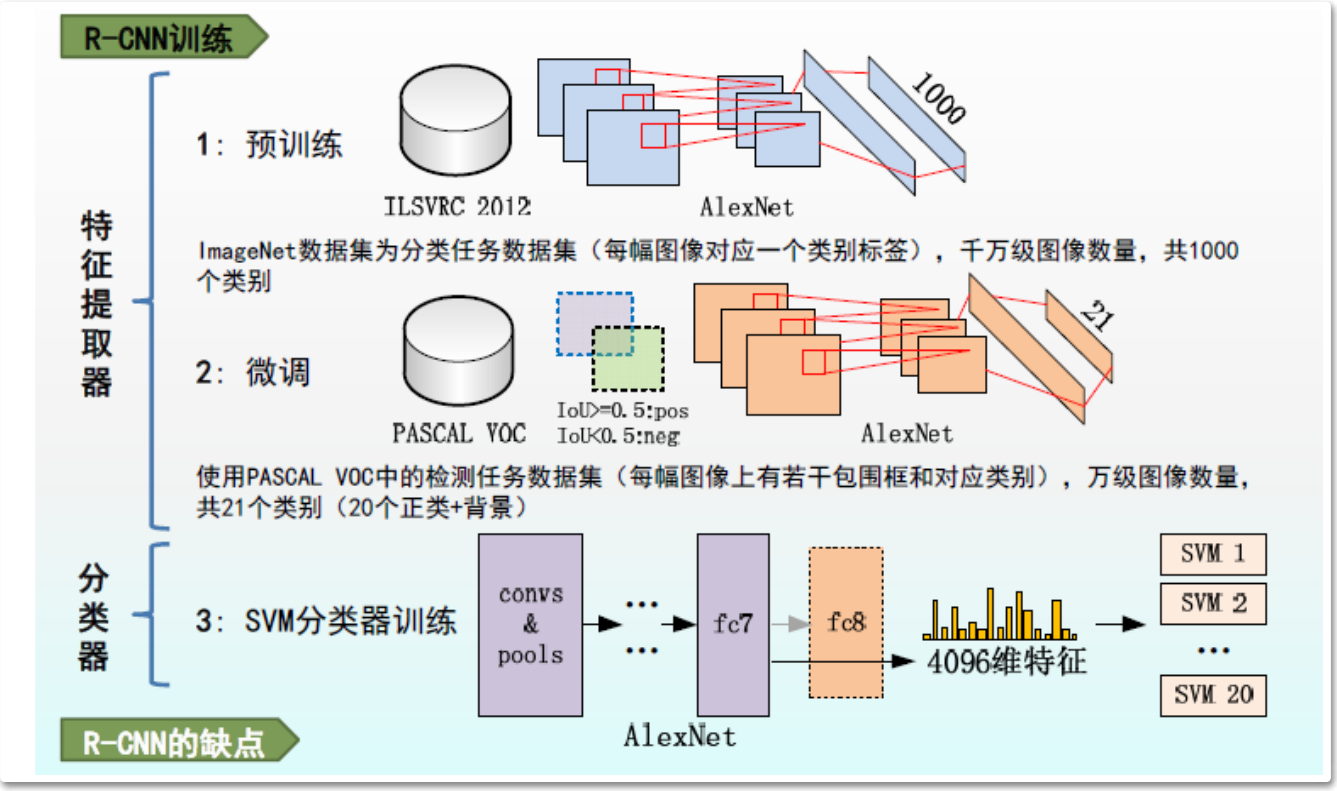
主要有两种主流的算法：一类是结合 region proposal、CNN网络的，基于分类的 R-CNN 系列目标检测框架，也叫两阶段网络（Two Stage）；另一类 则是将目标检测转换为回归问题的算法，也叫一阶段网络（Single Stage）。

现有的方法，各有什么特点？

基于分类的方法

基于分类的方法比较有代表性的工作是Ross B. Girshick与2014年提出的R-CNN，该方法首次将卷积神经网络应用在目标检测领域，并通过 Fast R-CNN、Faster R-CNN这一系列后续工作，将目标识别任务从仅用CNN做特征提取，发展成为使用CNN进行特征提取、Softmax进行分类（Fast R-CNN），最后发展为使用神经网络直接完成从区域检测、特征提取、分类的所有工作并实现了End to End的训练，因为引入用于目标检测的RPN网络相较于Selective Search算法的优异性能表现，Faster R-CNN方法基本实现了接近实时的目标检测。

我们说的基于分类R-NN系列目标检测框架也叫两阶段方法，那么这两阶段体现在哪里了？



上面这幅图，很好的解释了两阶段的过程：R-CNN模型使用Selective Search算法对输入图像进行候选区域的划分，对候选区域使用在大型数据集（如ImageNet ILSVC 2012）上预训练过的CNN模型进行特征提取，得到feature map；将该feature map送入SVM分类器进行分类工作，同时训练一个回归网络对候选区域的bounding box进行精修，让其更加贴合ground truth。

两阶段方法相比于传统的方法，两阶段的方法最大改进在于用CNN网络来表达更多的特征，相比于Haar角点或者HOG特征直方图等具有更强的表达能力。当然CNN的计算过程也是穷举变量图像，因此计算时间复杂度高，效率上没有优势，当时在最终的检测识别效果上有很大的改进。

基于回归的方法

之前基于分类的目标检测算法采用的方案的思路大体上可以分为两步：首先在图像上检测出许多不同尺寸的bounding box，再通过相应的分类器对bounding box内的区域进行评估，确定是否存在某类物体。

而基于回归的方法不同，该类方法把目标检测问题当作回归问题进行处理，所以不需要显式地对其进行划分和分类，而是使用网络直接对输入图像进行处理，输出检测出目标的分类、位置和置信度。基于回归的目标检测代表性方法有YOLO和基于YOLO发展出来的SSD。

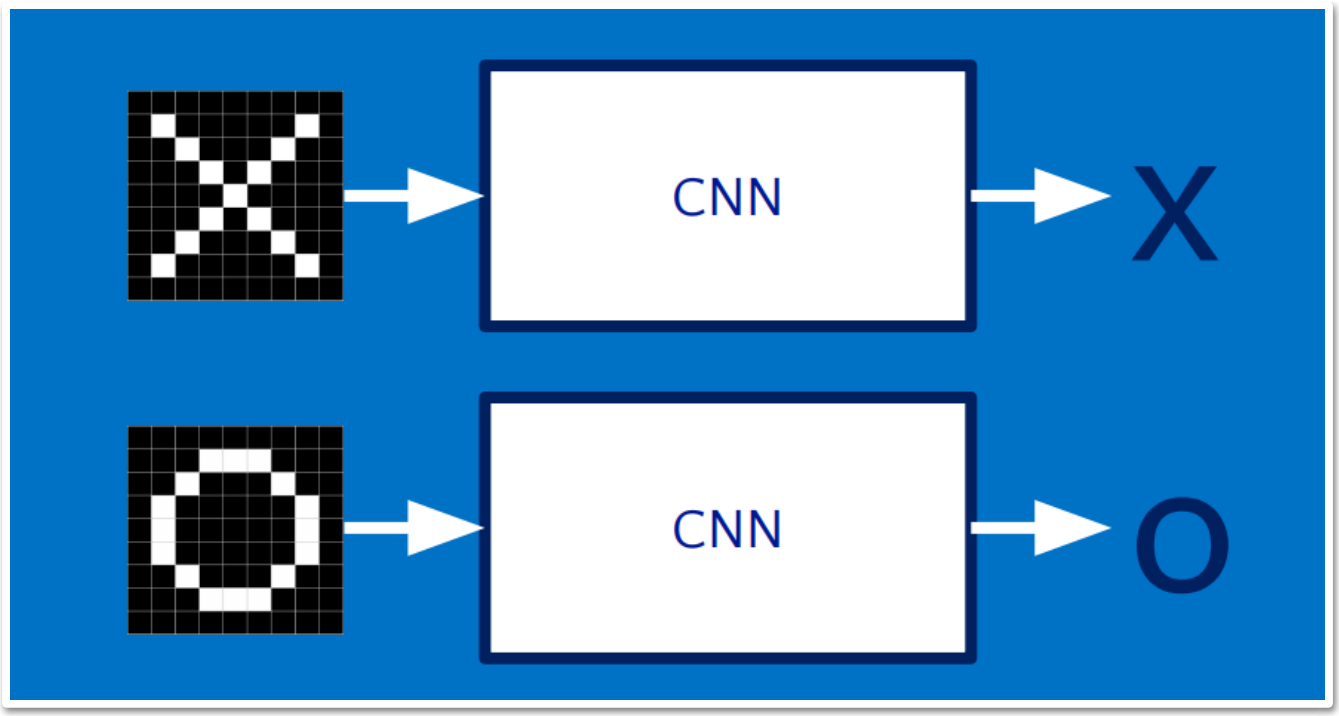
YOLO将输入图像划分为SxS个网格，如果一个物体中心落在某网格内，则该网格负责检测该物体。训练和测试时，每个网格预测B个bounding boxes，其中包括该bounding box的坐标和对应分类的置信度。

YOLO模型虽然速度很快，但是存在着一些缺陷；如每个网格仅能预测一个目标，相邻目标容易产生漏检；对于物体的尺度较为敏感，对尺度变化较大的物体泛化能力差；对小目标检测效果不佳等[15]。

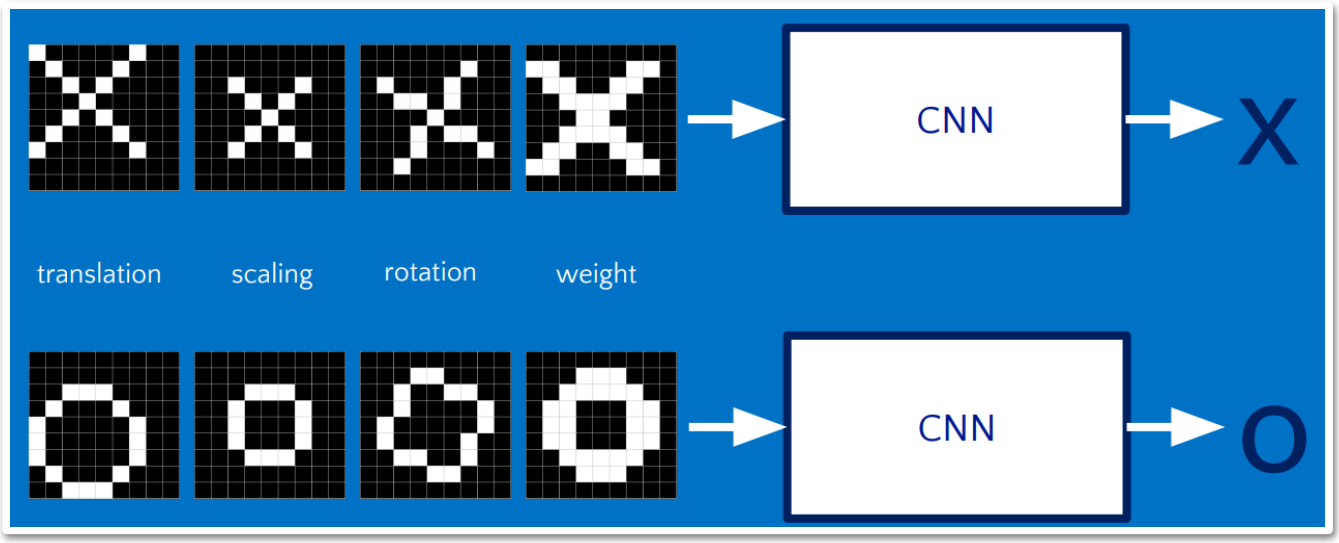
故SSD综合了YOLO和Faster R-CNN的anchor box思路，不再使用统一固定大小的cell，而是在不同层的feature map cell上划分出不同长宽比的default box，不同层的feature map上具有不同的感受野，这样能够有效兼顾不同尺度的目标。

CNN详解

目标 如何识别手写的英文字母？再简单一点点：比如给一个"X"的图案，计算机如何识别这个图案就是"X"?



如上图所示，CNN就是要告诉我们输入的是X还是O。下面带着这个目标，一步一步的推导一下CNN。



在识别过程中，还存在一个问题：图片畸形。图片或多或少会存在畸形，CNN也要能够正确的识别，这才能满足需求，因为在现实中，图片畸形是广泛存在的。

什么是卷积

数学上对卷积的定义：

连续卷积：$(f*g)(n)=\int_{-\infty}^{+\infty}f(\tau)g(n-\tau)$

离散卷积：$(f*g)(n)=\sum_{\tau=-\infty}^{\infty}f(\tau)g(n-\tau)$

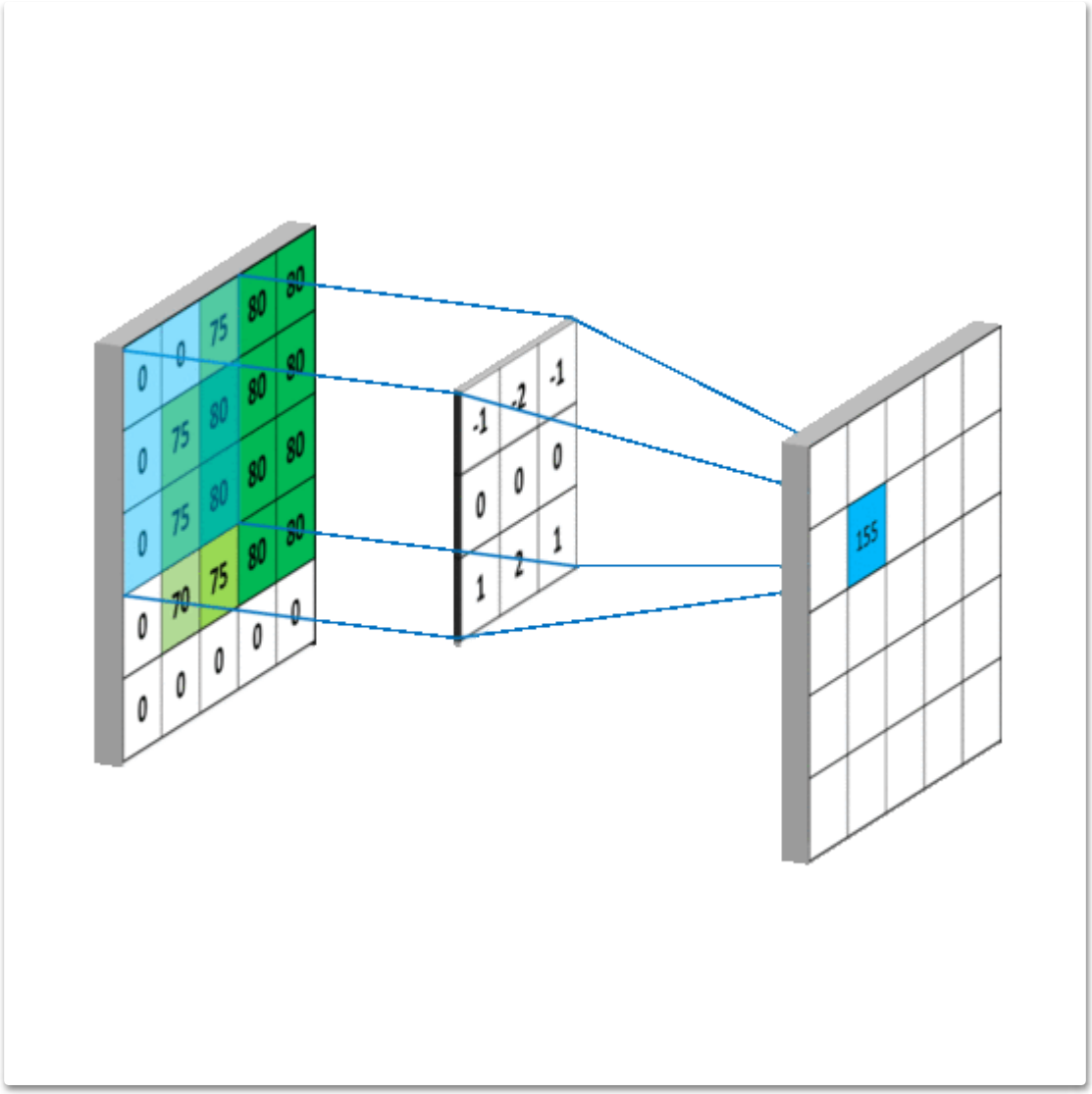
当看这两个公式很抽象，我们来举一些更加形象的例子。

例如：我们有两枚骰子，同时仍出去，现在我们要计算两枚骰子点数加起来为4的概率是多少？

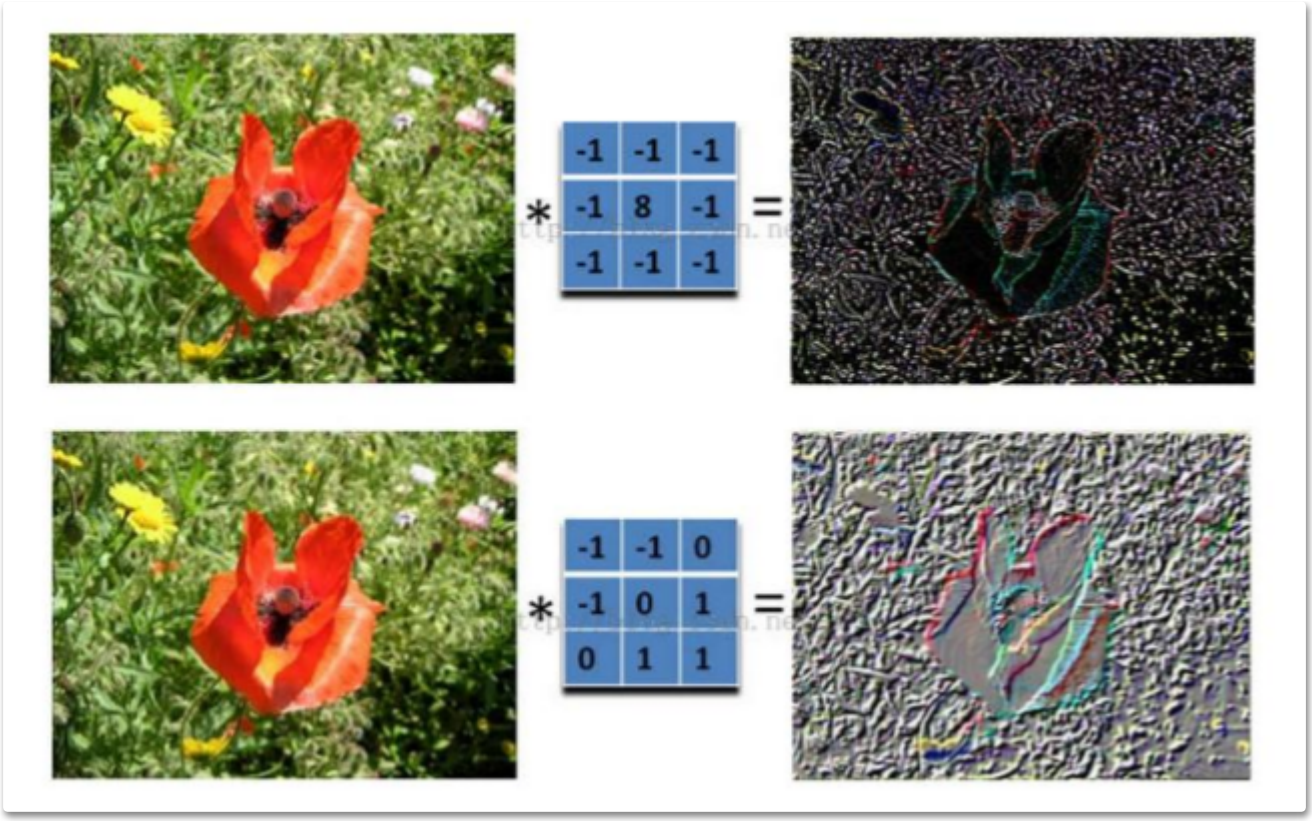
出现总和为4的概率的几种情况有： f(1)g(3),f(2)g(2),f(3)g(1)。可以用如下的卷积公式表示（离散卷积）：

$$(f*g)(4)=\sum_{m=1}^{3}f(4-m)g(m)$$

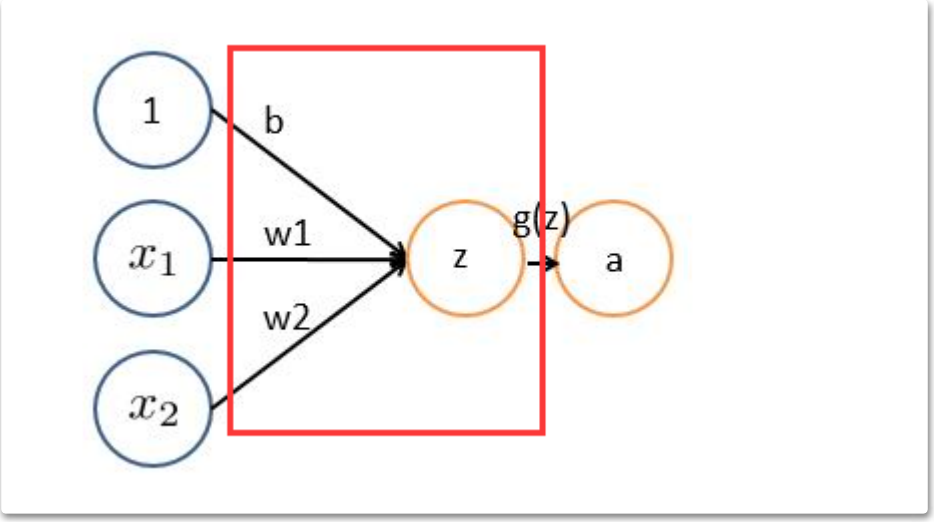
看着这些公式表示懵逼~~~~用图像处理中的思路来解释，看下面的gif动画，我们可以认为f(x)就是原始的图像，g(m)就是那个小的矩阵（可以认为是各种滤波器），卷积操作，就是用小的矩阵g(m)依次与原矩阵f(x)做内积（相乘再相加），然后用内积替换原图像f(x)总相应位置的值。


0 0 75 80 80
0 75 80 80 80
0 75 80 80 80
0 75 75 80 80
0 70 0 0 0
0 0 0 0 0
-1 -2 -1
0 0 0
1 2 1
155

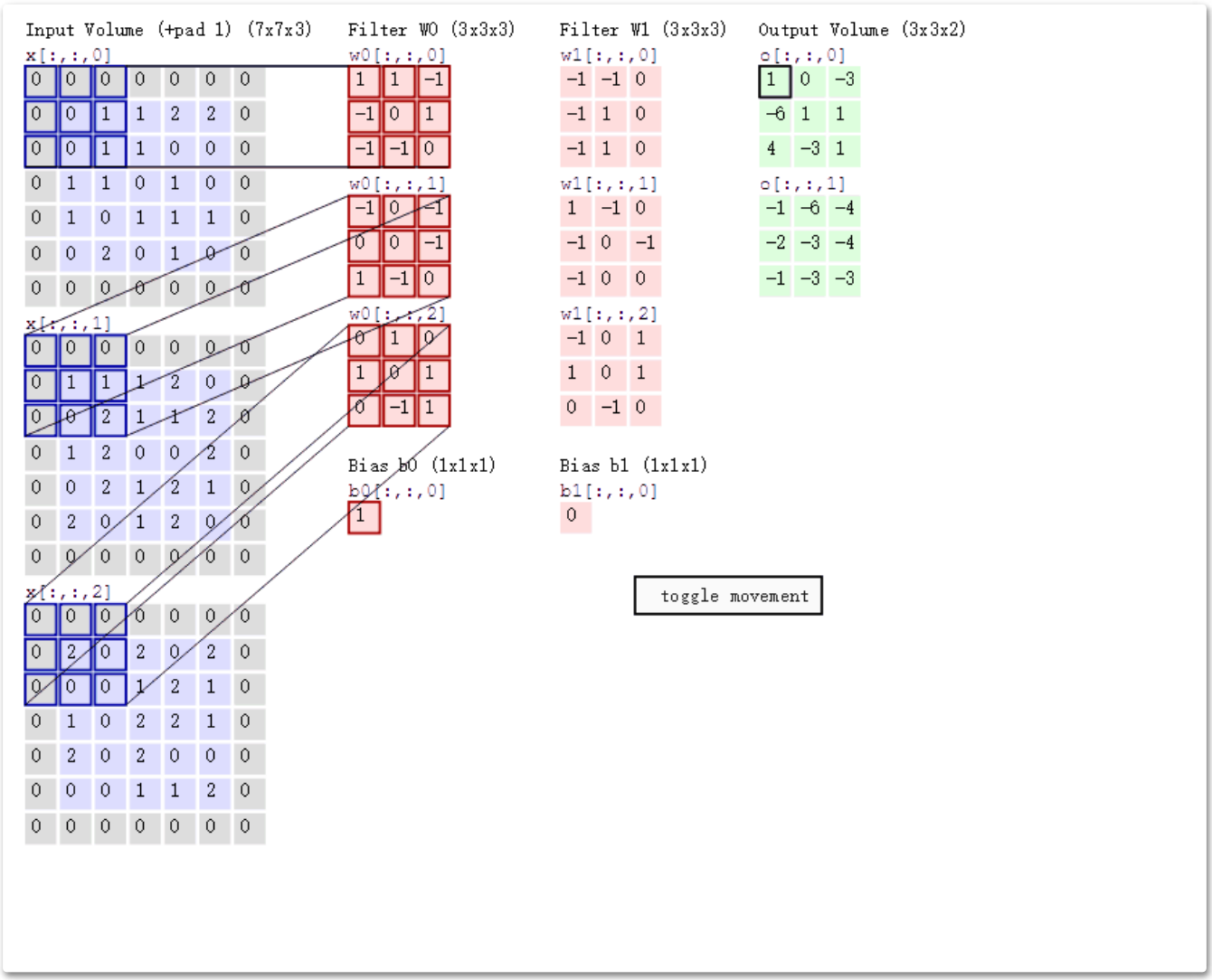
图像卷积的例子：



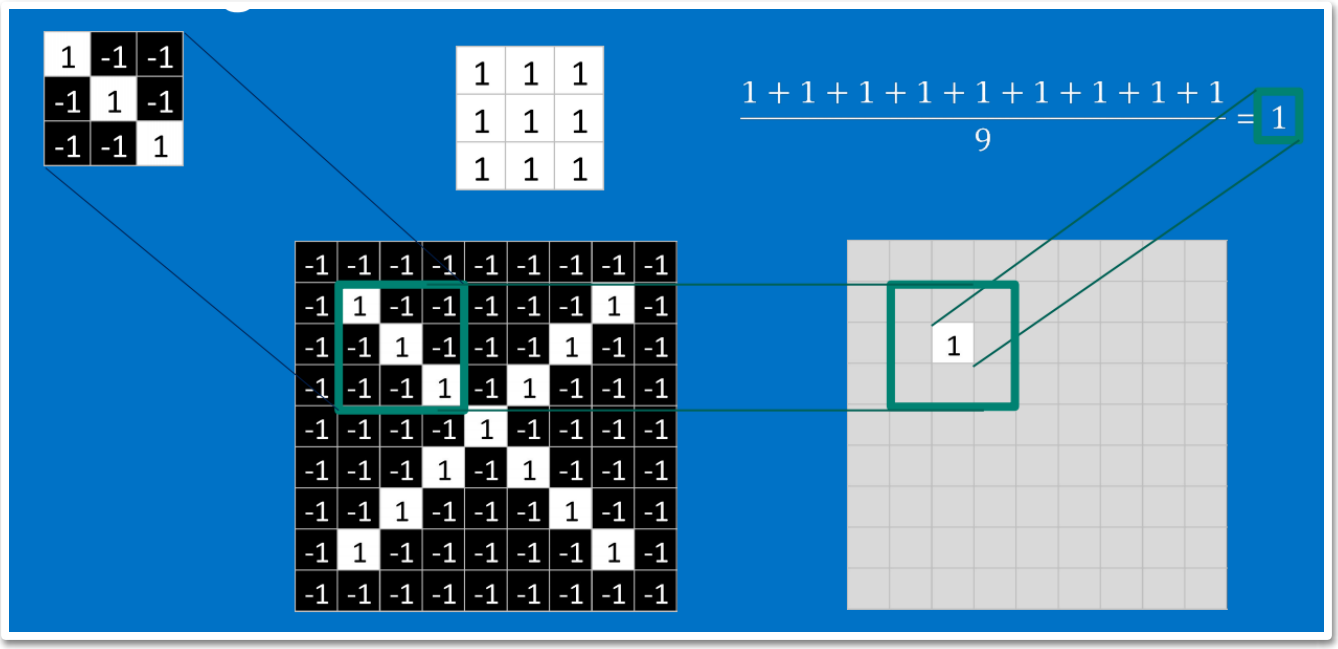
在CNN中，卷积指啥意思呢？ 拿上面的图片识别的例子，卷积就是未知图案的局部和标准图片的局部一个一个比对时的计算过程，卷积计算结果为1表示匹配，否则不匹配。 说的太笼统了~~能有更加具体一点的吗？ 在CNN中卷积操作到底指啥呢？ 前面图像的例子也说了，卷积是原图像与另外一个矩阵的内积，那么在CNN中，卷积也是内积，是什么的内积呢？其实CNN的卷积操作，就是神经元与输入的内积。继续回顾一下神经元的组成（下图），$x1, x2, x3$ 都是输入，在每一次数据上都有一个固定的权重（w）以及偏置，因此我们可以类比图像的卷积：固定的权重就是小的矩阵，输入就是原始图像，那么在CNN中卷积就是输入经过神经元的处理得到的输出这个一个过程。



再来一个更形象的例子： input Volume表示输入， Filter w0是第一次卷积层神经元的权重，Filter w1是第二层卷积层神经元的权重。



这里有必要解释一下卷积内积是怎么计算的？回忆一下向量的内积：$x(x1, x2, x3)$* $y(y1, y2, y3)$ =$x1y1 + x2y2 + x3y3$，也就是向量的各分量相乘再相加。矩阵，可以想象层高维的向量，那么矩阵的内积也可以类比到向量的内积：**各对应位置的分量相乘相加，然后再除以矩阵的总元素个数。** 这就是CNN的核心计算逻辑--矩阵对应位置相乘相加除以元素个数，没有什么高深的。计算的CNN内积放置的位置，是原始图像的矩阵的中心位置。下面图解释了CNN的计算过程：

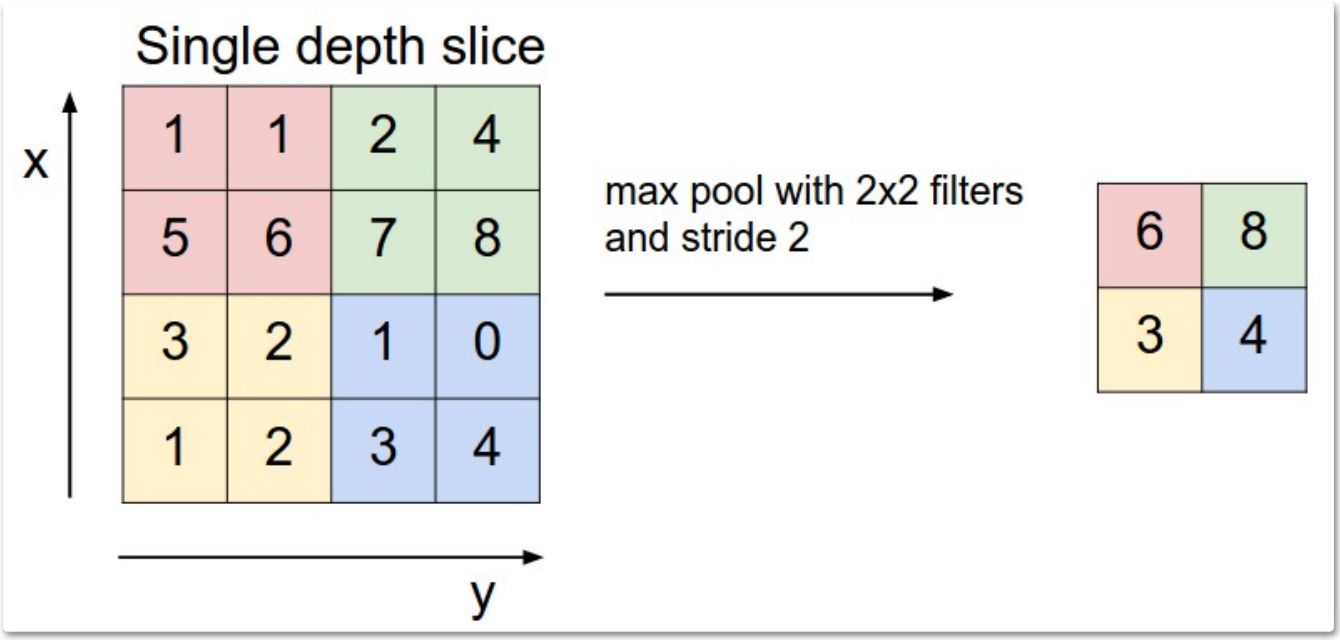


RULE 激励函数

在CNN中RULE激活函数其实叫做 修正线性单元，做的事情也很简单：对于计算结果为正数的保留，对于计算结果为负数的，将其值设置为0.

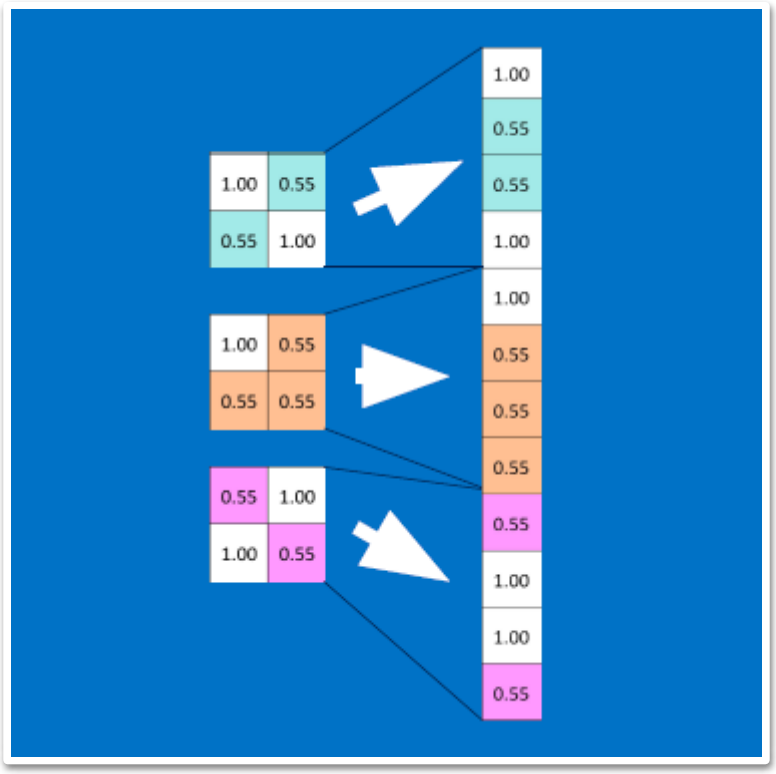
池化 POOLING

池化，简言之，即取区域平均或最大：



全连接层

全连接层是CNN的最后一层了，经过这一层的处理，就能得到最终的结果。



上面的图很简单的解释了全连接层是如何构造的，当然实际上可能跟上面并不是很像，简单起见还是就上图那样的吧。 全连接把最终卷积的结果线性排列。

得到这个排列后，我们这么得到我们最终的结果呢？比如说：图案是X吗？ 这里有一个很大的前提就是：CNN分为训练过程和验证过程，在训练阶段，我们是知道图案到底是不是X。因此，简单的说，全连接层的线性排列实际上就代表了一个训练结果，在验证阶段，如果图案的全连接与训练的全连接层类似，可以认为是X图案。

在实际中，我们不可能比较全连接层的线性排列，而是把全连接层也看着一个卷积层，这是这个卷积层有点特殊：是1*n的卷积核，n表示全连接层线性排列的元素个数。我们用卷积来表示全连接层的输出：

$$
\begin{aligned}
y &= W * x + b \\
p &= sigmoid(y) \\
Loss &=
\end{aligned}
$$

x就是全连接层的排列，W就是卷积核（$1 * n$），b偏置，y输出值。经过sigmod函数处理，把y值归一化。此时的y值就作为训练的结果了。为了保证训练的模型更加精确，我们要保证y的值是最优的，因此还衍生出最后一步：优化。这里的优化就是计算Loss损失函数的最小值，反推W和b的最优值。在计算最优值的过程中，用的的数学方法就是梯度下降法~~到这里，是不是很明白了，之前看了很多地方在全连接层都说梯度下降，但是没有说明怎么用。

reference

https://www.jiqizhixin.com/articles/2017-04-06

https://www.zhihu.com/question/34223049

http://www.voidcn.com/article/p-kkznyico-ue.html

http://www.voidcn.com/article/p-vllyzrme-bon.html

https://t.cj.sina.com.cn/articles/view/6552764637/1869340dd01900d15p

https://zrstea.com/264/

https://www.tinymind.cn/articles/685

https://www.cnblogs.com/houjun/p/8424893.html

https://blog.csdn.net/v_JULY_v/article/details/80170182