

$$y=wx+b$$

w—学习到的权重/要学习的参数

b—偏置 就是阈值

$$\sigma(w^T x + b)$$

阶梯函数（二值函数/分类）

这个图右边就是一个简单的神经网络，输入层，一个中间层，最后输出，以一个计算实例理解：

? : 1 XOR 0 = 1

输入：x1 = 1, x2 = 0;

输入层→中间层：w1 = (1, -1)', w2 = (-1, 1)' b1 = b2 = -0.5

$1*1 - 0*(-1) - 0.5 = 0.5 \rightarrow 1$

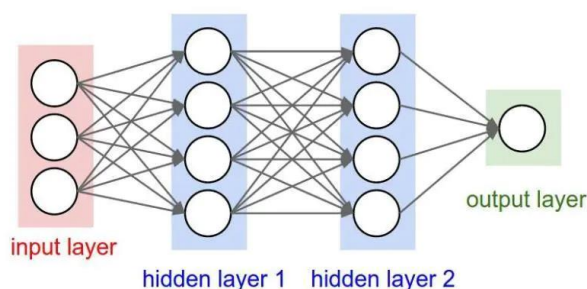
$1*(-1) - 0*1 - 0.5 = -1.5 \rightarrow 0$

中间层→输出层：w1 = (1, 1)' b1 = -0.5

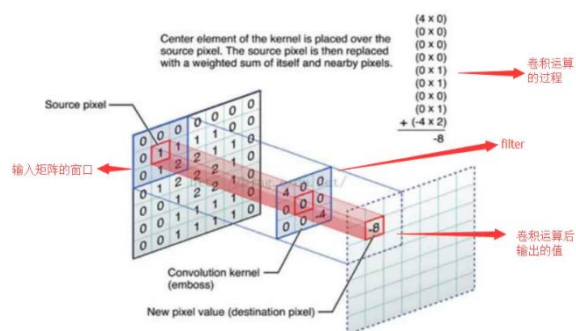
$1*1 - 0*1 - 0.5 = 0.5 \rightarrow 1 = 1 \text{ XOR } 0$

博客 <https://www.jianshu.com/p/1ea2949c0056>

中间层可能有很多层，结合网页理解不同的神经元提取不同特征，进行运算，更抽象的表示：



模型

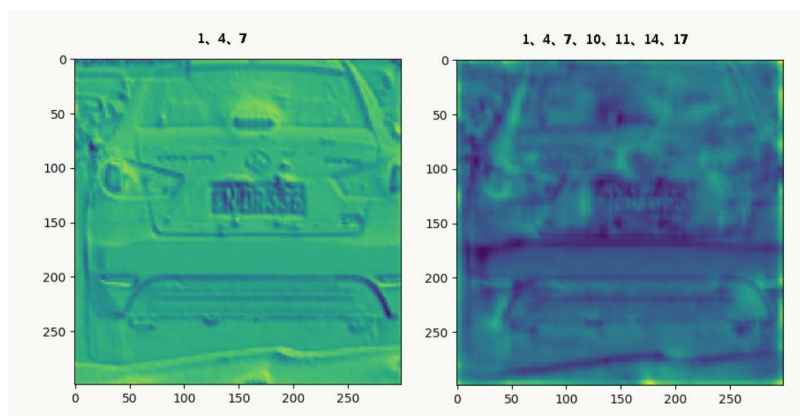


中间层具体计算：卷积，卷积核与像素及其领域做运算

逐层卷积之后越来越抽象，以图像为例，输入是像素的矩阵，最开始的几层可以看出来大致提取了什么，比如可能是一些点特征，线特征提取算子。“神经”一次也来自于这个抽象过程，在生物学研究人眼感知时，猜测感知过程也是一层一层不断抽象



原图



1,4,7 层平均

ooo

3. primer C++学习：智能指针，编译原理

智能指针内部保存一个计数器，记录（任意地方）由智能指针生成的共用这个地址的指针的个数，复制（改变等号两边指针的计数），作为参数返回值（销毁返回对象减一，外界调用指针加一）等操作都会改变计数，计数为 0 就会自动销毁。所以如果混合智能指针和普通指针就会因为两种销毁造成问题

2020.12.27

阅读论文：

《Simultaneous Localization and Mapping: Part I》：比较早，类似教程，学习了 slam 的定义，早期发展，介绍了 EKF 卡尔曼滤波（大致知道怎么做的）和粒子滤波方法（没看懂）

1. 问题定义：移动机器人是否有可能被放置在一个未知环境中的未知位置，是否有可能逐步建立一个一致的地图，同时确定自己在该地图中的位置。
2. 已知：初始坐标，每个时刻的控制输入（状态转移，也就是给出了机器人每步的运动），每时刻对 landmark 的观测值，分别对应状态转移模型和观测模型
解求：不断循环，解求当前机器人的位姿和更新 landmark 的位置

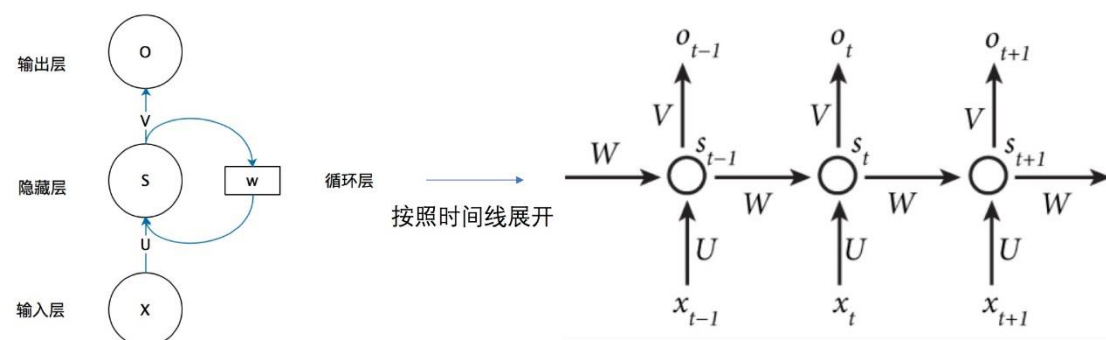
《Deep Learning for 3D Point Clouds: A Survey》：

1. 点云处理的综述，比较长，挑着看了一下，介绍了形状分类，目标检测，跟踪，分割（语义，实例，部件）的研究，
2. 从语义到实例到部件的分割，对象越来越小，越精细，方法大致都是逐点，分块，图方法，主要问题是领域信息和效率的矛盾，
3. 涉及概念比较多，介绍的方法不是很懂，重点了解了 RNN（Recurrent Neural Networks (RNN)）的概念

x: 输入

s: 隐藏层

o: 输出



RNN 用于自然语言处理领域，和一般 CNN 的区别就是更好的利用上下文信息，例如输入一个序列的词语，不同的语序也会对预测结果产生影响，t 时刻/次的中间层 s_t 受到两个方面的影响：1，当前输入 x_t ；2，t-1 时刻/次输入产生的中间层 s_{t-1} 。

本文介绍的用在语义分割中 RNN 主要是利用不同尺度下的信息，将点云聚类成不同的尺度块，形成不同尺度的点云块上下文信息，进行训练