**2020.12.19**

**1. 阅读论文：**

《You Only Look Once Unified, Real-Time Object Detection》:YOLO，目标检测，

《Rich feature hierarchies for accurate object detection and semantic segmentation》R-CNN，目标检测，主要分为三个步骤：

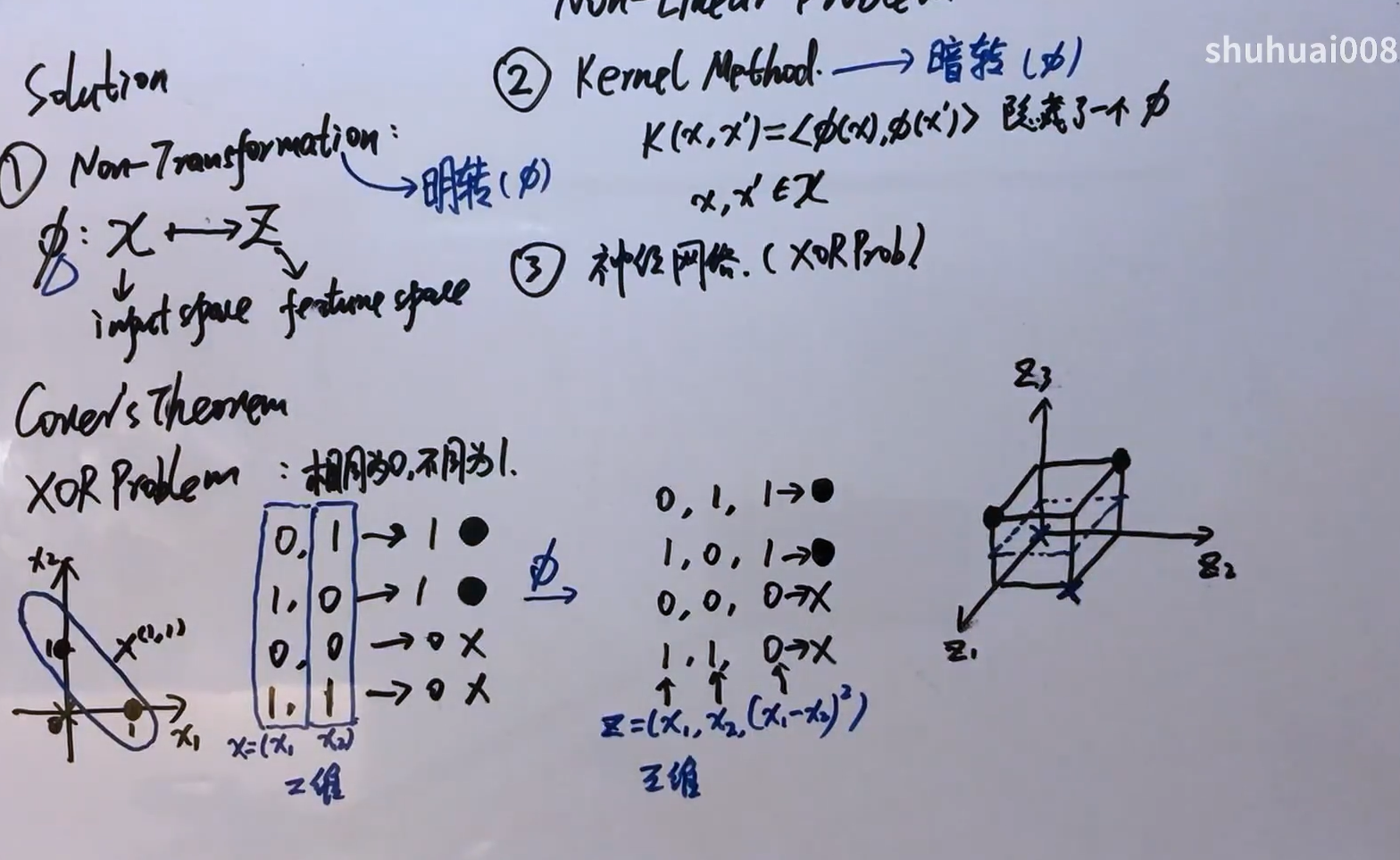
1. SS算法提取候选框，了解了SS算法，跟一种多边形分割思想类似，都是从很小的基本单元自下而上合并
2. 卷积神经网络：提取一个固定长度的特征向量
3. 线性SVM分类

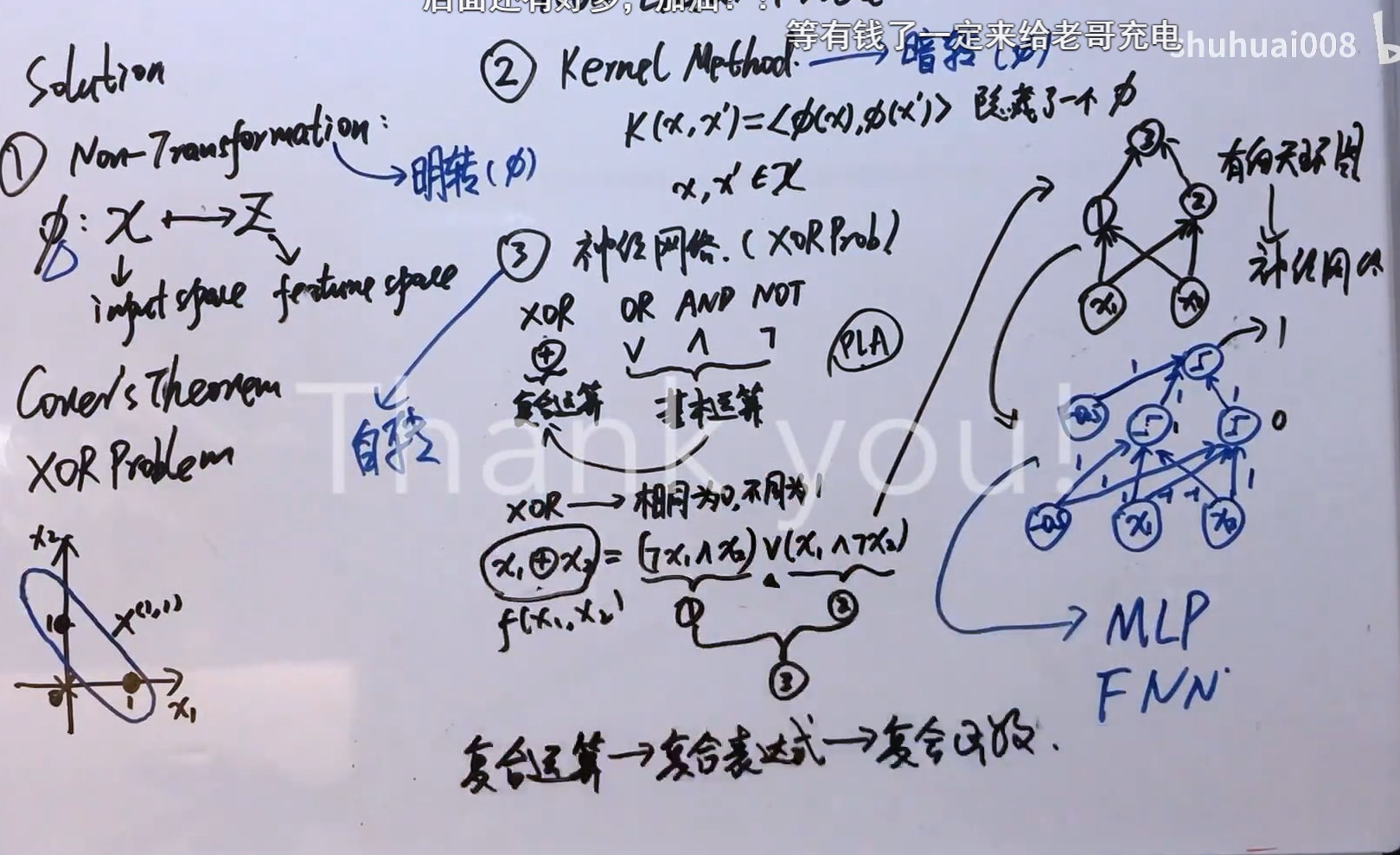
《A Maximum Likelihood Approach to Extract Finite Planes from 3-D Laser Scans》：看了一半，18年IROS获奖教授的学习的最新论文，介绍平面提取，每个点对应一个平面，然后多层次聚类，不怎么懂

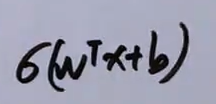
**2. 学习卷积神经网络：**

学习白板推导：前馈神经网络，理解了**异或问题**的神经网络模型，实际是一个复合函数，解决非线性问题，系数就是权重，偏置是阈值

比较经典的是这个图：这个图介绍了处理非线性问题的三种方式：①通过数学人为定义函数，变换到另一个空间/高维空间，例如异或问题x1 XOR x2，定义变换（x1, x2, (x1 - x2)^2），到可分类的三维空间，右下的蓝色虚线就是分类超平面②核方法，没懂③神经网络





y=wx+b

w—学习到的权重/要学习的参数

b—偏置 就是阈值



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 → 1

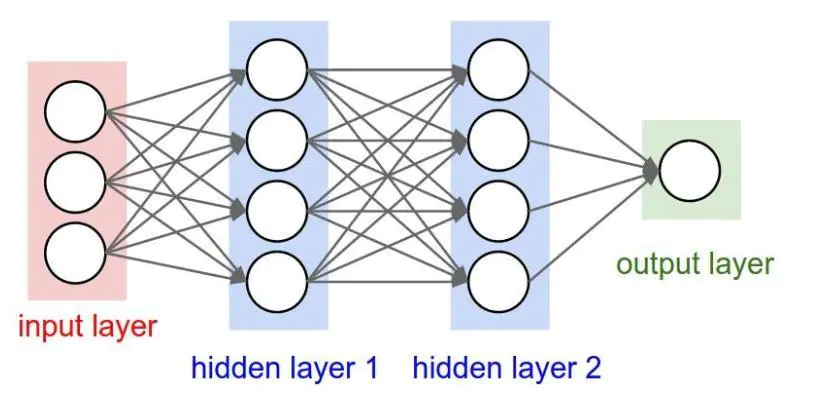
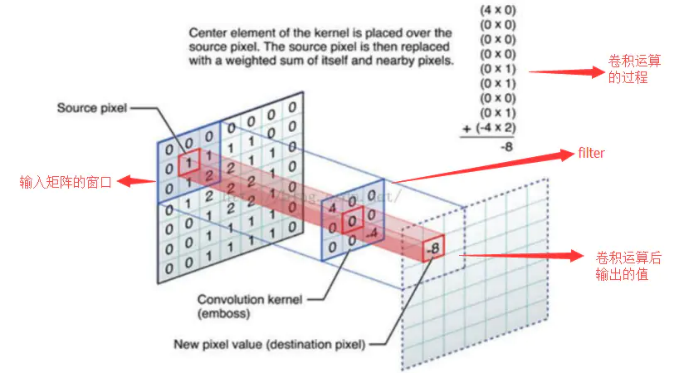
1\*(-1) – 0\*1 – 0.5 = -1.5 → 0

**中间层→输出层**：w1 = (1, 1)’ b1 = -0.5

1\*1 – 0\*1– 0.5 = 0.5 → **1 = 1 XOR 0**

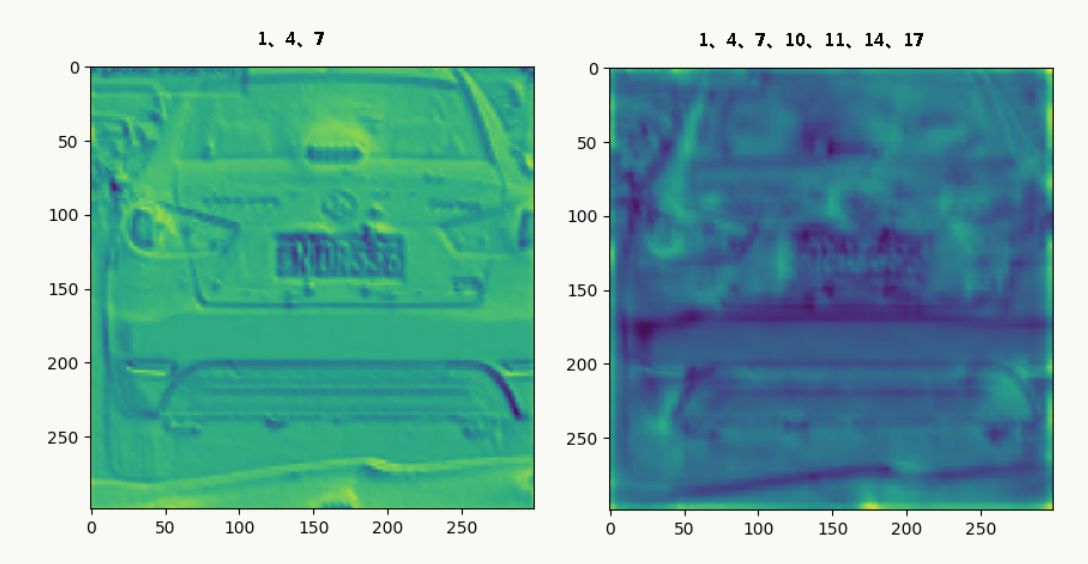
博客**https://www.jianshu.com/p/1ea2949c0056**

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 。。。

**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

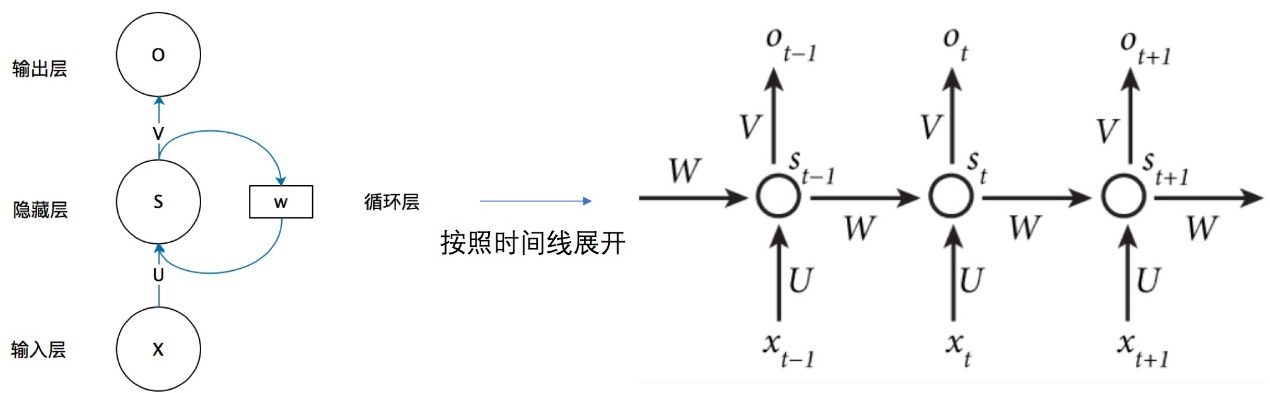
**《Deep Learning for 3D Point Clouds: A Survey》**：

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出影响，t时刻/次的中间层St受到两个方面的影响：1，当前输入xt；2，t-1时刻/次输入产生的中间层St-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