# 2.1图像分类-数据驱动方法

Robust 鲁棒：鲁棒性（robustness）就是系统的健壮性。

语义鸿沟：sementic gap 图片 -- 文字

以‘一只猫’为例，

角度(viewpoint variation)，

亮度(Illumination)，

姿势(Deformation)

遮盖(Occlusion)

背景混入(Background Clutter)

组内变异(intraclass variation):有很多只小猫，都是猫，但有或多或少的不同

的不同，代表同一只猫

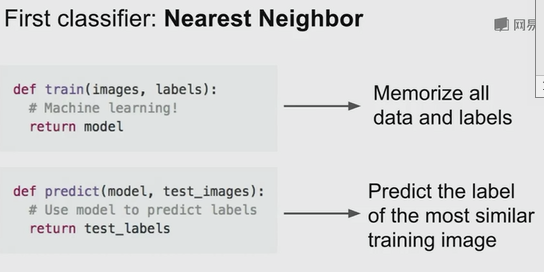
不能直接写一个判断‘猫’的函数来做这样的事

## 数据驱动逼近：Data-Driven Approach

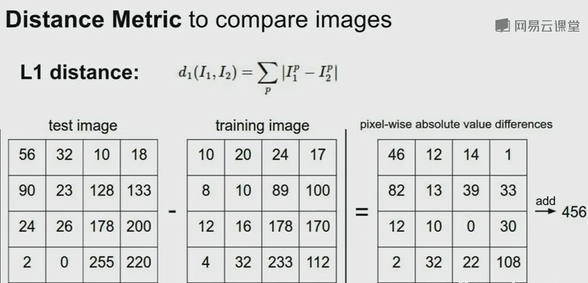
1. Collect a dataset of images and labels 找到同类的，已经标注的大量样本图片
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images



### 最简单的例子就 K近邻



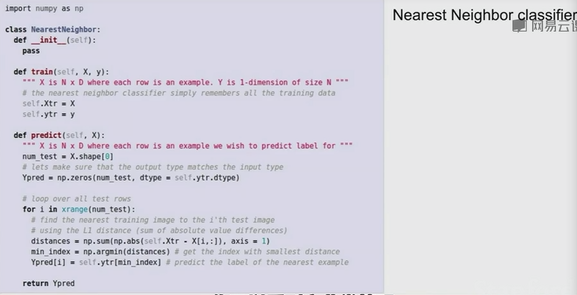
CIFAR-10的例子中，应用了L1 distance,



下面是K近邻的python, 应用numpy的计算，计算L1 distance

Train阶段什么也不做

Predict阶段，计算所有test图片和预测图片的L1距离



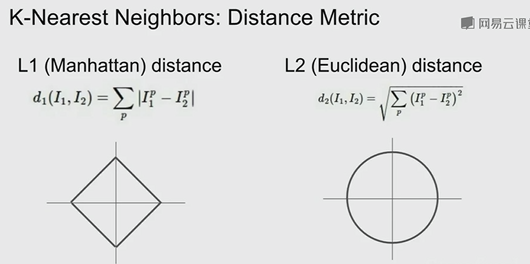
Training 的复杂度： O(1), 只是复制

Predict的复杂度： O(N), 要计算所有样本与预测图片的L1距离

We want classifier who are fast at prediction, slow only for training

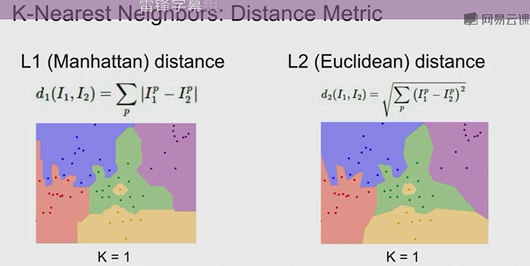
除了L1距离，还有L2距离等

注意，L1距离认为，方形上的距离和原点都是相同的， 例如在[平面](https://zh.wikipedia.org/wiki/%E5%B9%B3%E9%9D%A2" \o "平面)上，坐标（*x*1, *y*1）的点*P*1与坐标（*x*2, *y*2）的点*P*2的曼哈顿距离为：



K-NN 认为所有的像素点都是同等重要的

‘距离’的概念不仅局限于向量，图片，还可以应用于文本，等



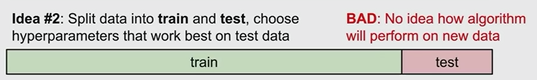
当使用L1时（左图），决策边界更倾向于沿这样的方向，因为这样的方向的距离所取的点的距离是一样的，

因为L1更依赖坐标，如果当坐标有意义时，可能比L2距离更好一些

K:超参数 hyperparameters, 设置的而非是习得的

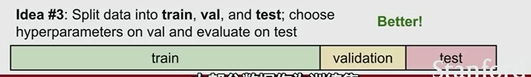
设置超参数：depends

1. K=1 对训练集的拟合成都往往较高，但泛化能力往往很差
2. 把train data 分割成 train 和 test ,进行训练和验证



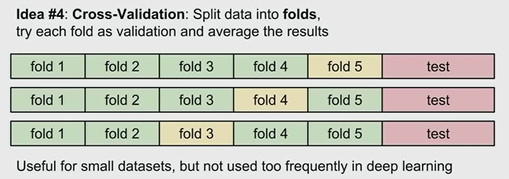
1. Better way ,把data split to 3 parts: train, val , test.

Choose hyperparameters on val , and evaluate on test!



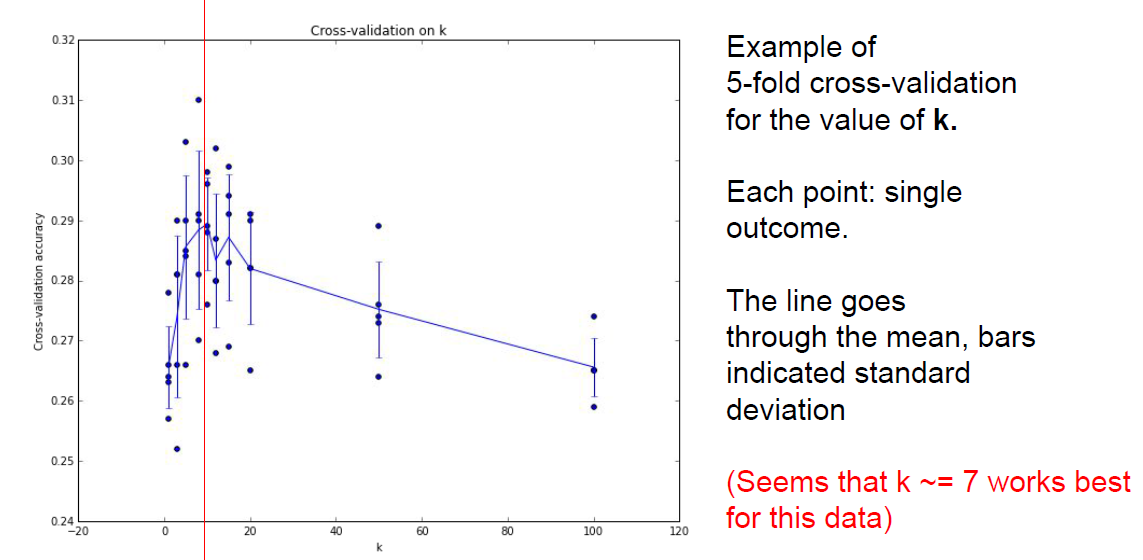
最后才使用test集

1. 交叉验证，在小数据集中应用，在深度学习中不常用



数据集的数据应该是随机的，默认每个点服从统一分布，相互独立，有种方法，是一次获取大量数据，预处理时再打乱，分为训练和测试集

经过交叉验证，会得到pic bellow:



X轴：K超参的取值 , 这里k取值为约为7时accuracy最高

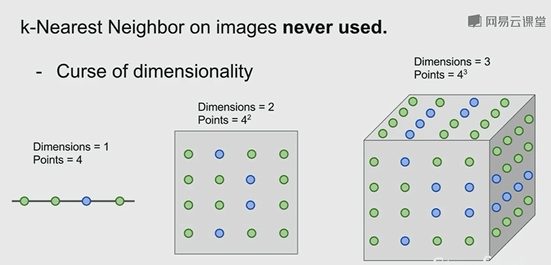
Y轴：交叉验证的accuracy 的

Deviation : 误差

另外一个KNN不适合用在图像分析上的原因是，这样的‘距离’概念不适合表示图像的相似度，



Curse of dimensionality 维度灾难



KNN学习训练集，目的是把样本空间分成几块（进行分类）

如果我们希望分类器分的好，需要训练数据能密集的cover样本空间，否则就会出现样本中有一些‘稀疏’的区域，样本点的距离就会比较远，没有太高的相关度

问题是如果想要密集cover空间，我们需要维度指数倍这么多的data，在高维度，这么多的data数量是很难做到的

KNN对数据的分布情况 没有任何预估，所有只能‘祈祷’数据的分布在高密度的