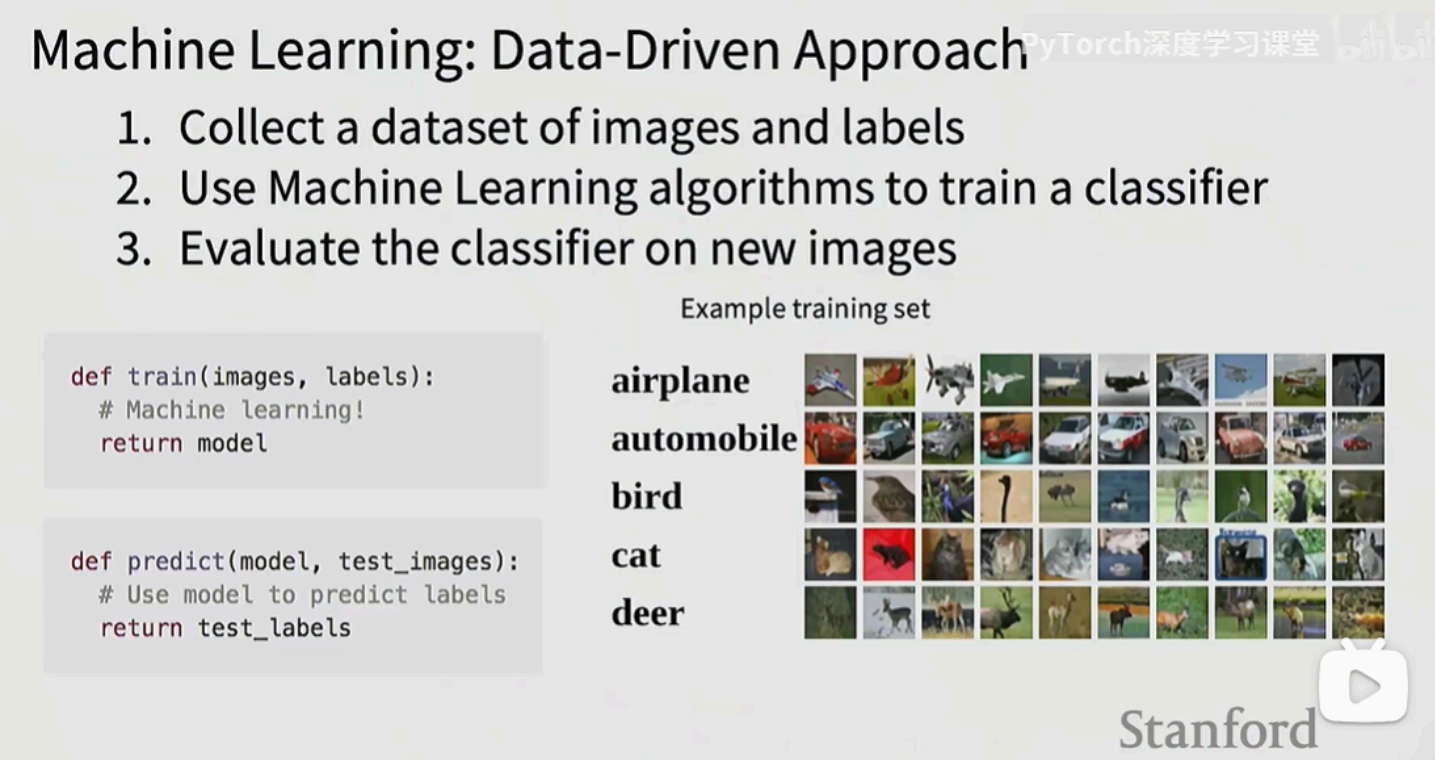
Lecture 2:Image Classification with Linear Classifiers

Two basic data-driven approaches to image classification：K-nearest neighbor and linear classifier

challenge：

viewpoint variation,illumination(光照),background clutter(杂乱)，occlusion(遮盖),deformation(变形),intraclass variation,context(环境)

Machine Learning 主要步骤

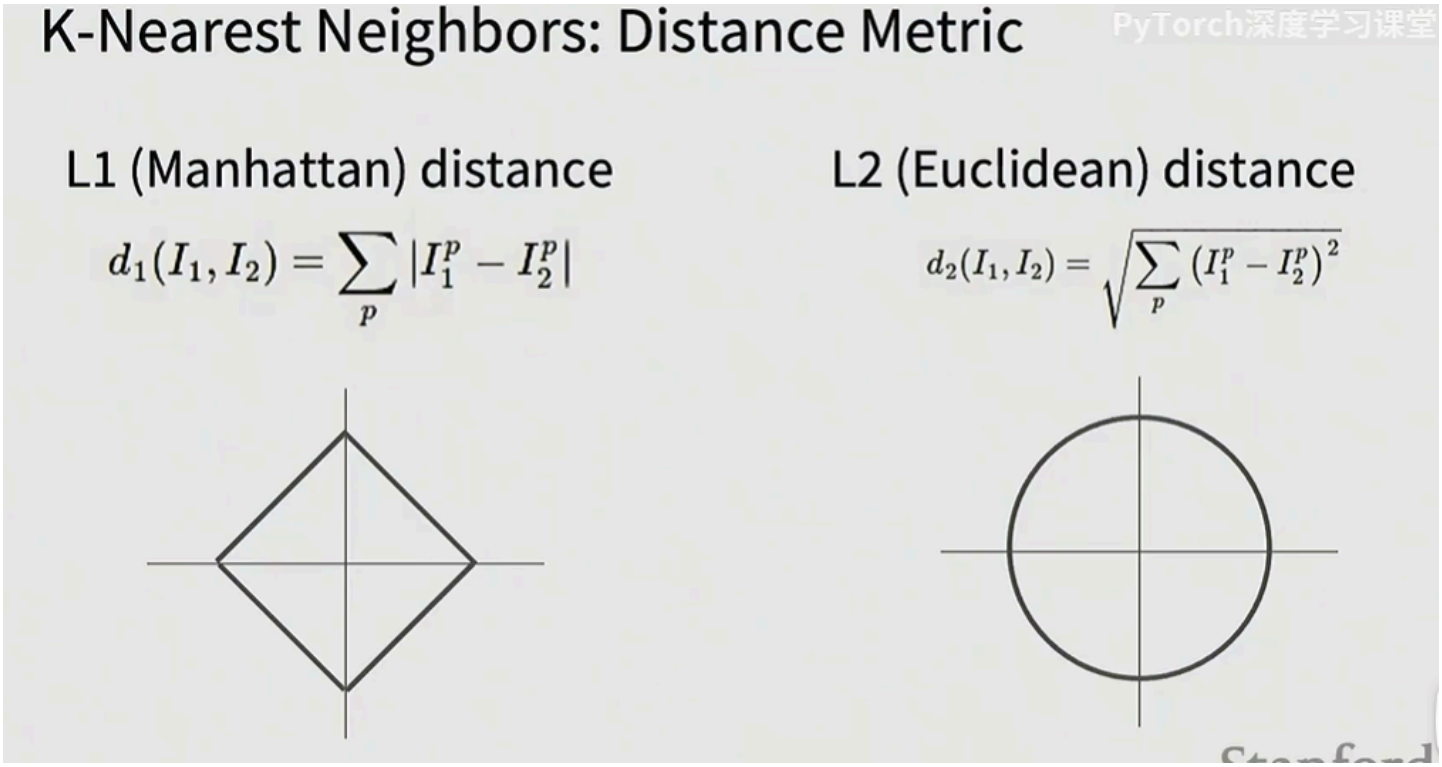


Nearest Neighbor Classifier

train就是memory the data
predict就是把要预测的点与所有之前train时存下的数据算距离（至于用什么距离下文会谈到），找到最近的点的label就是预测值。如果是k-Nearest Neighbor就是找到k个最近的点，出现最多次的label就是预测值。
所以训练速度：train step O（1）；predict O（N）

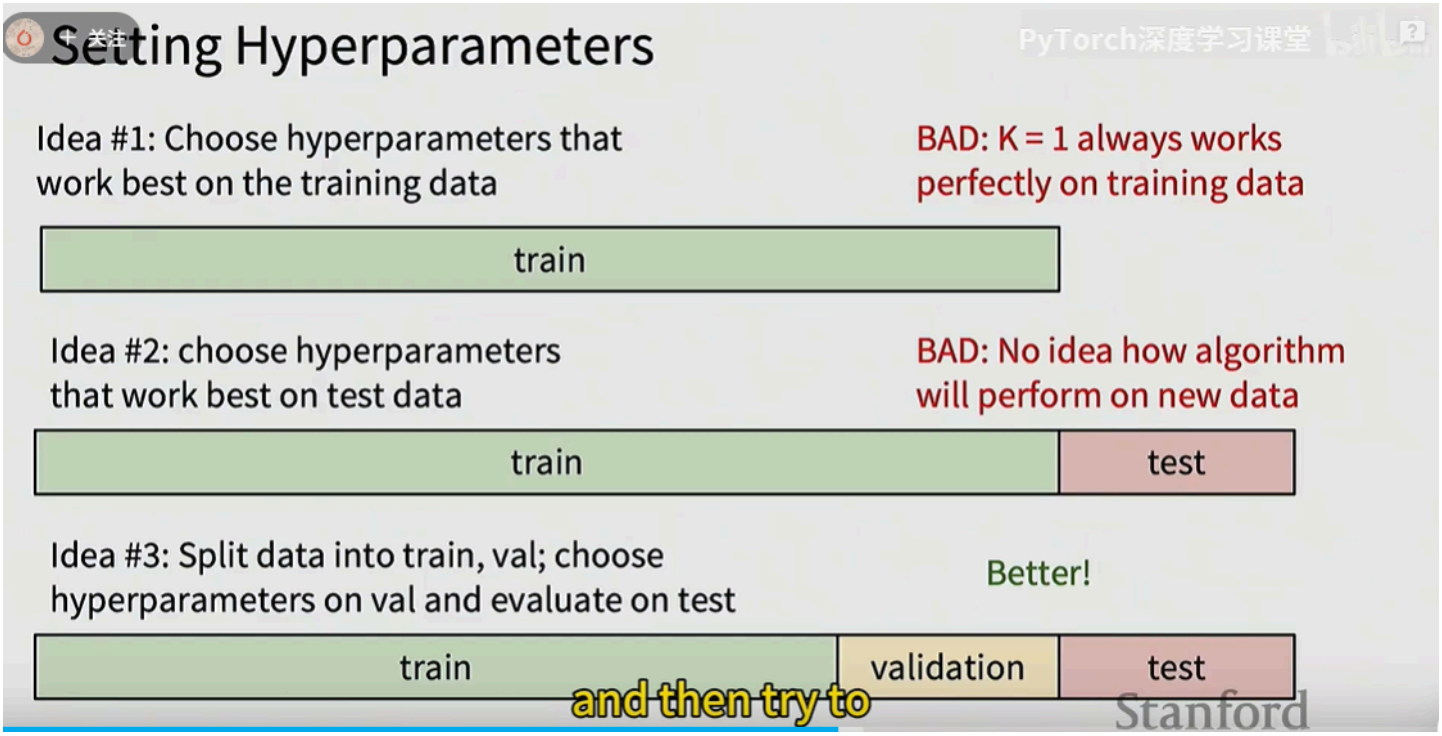
由此可以看出我们实际不可能使用Nearest Neighbor 算法，我们都是希望开发者花很长的时间把模型训练好，我们使用时可以快速预测才行。

Distance



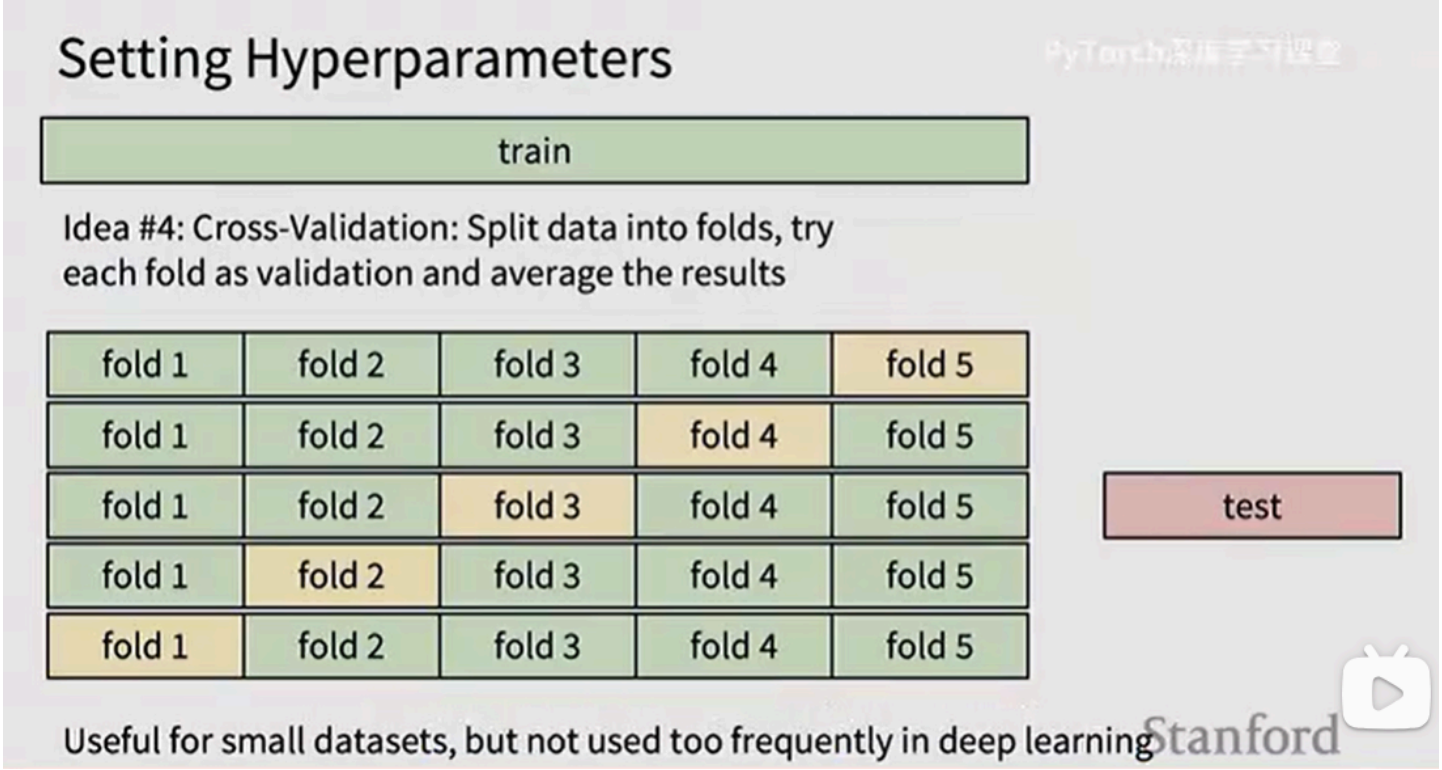
怎么选？比如：如果你的feature对旋转敏感，选L1；不敏感选L2

Hyperparameters



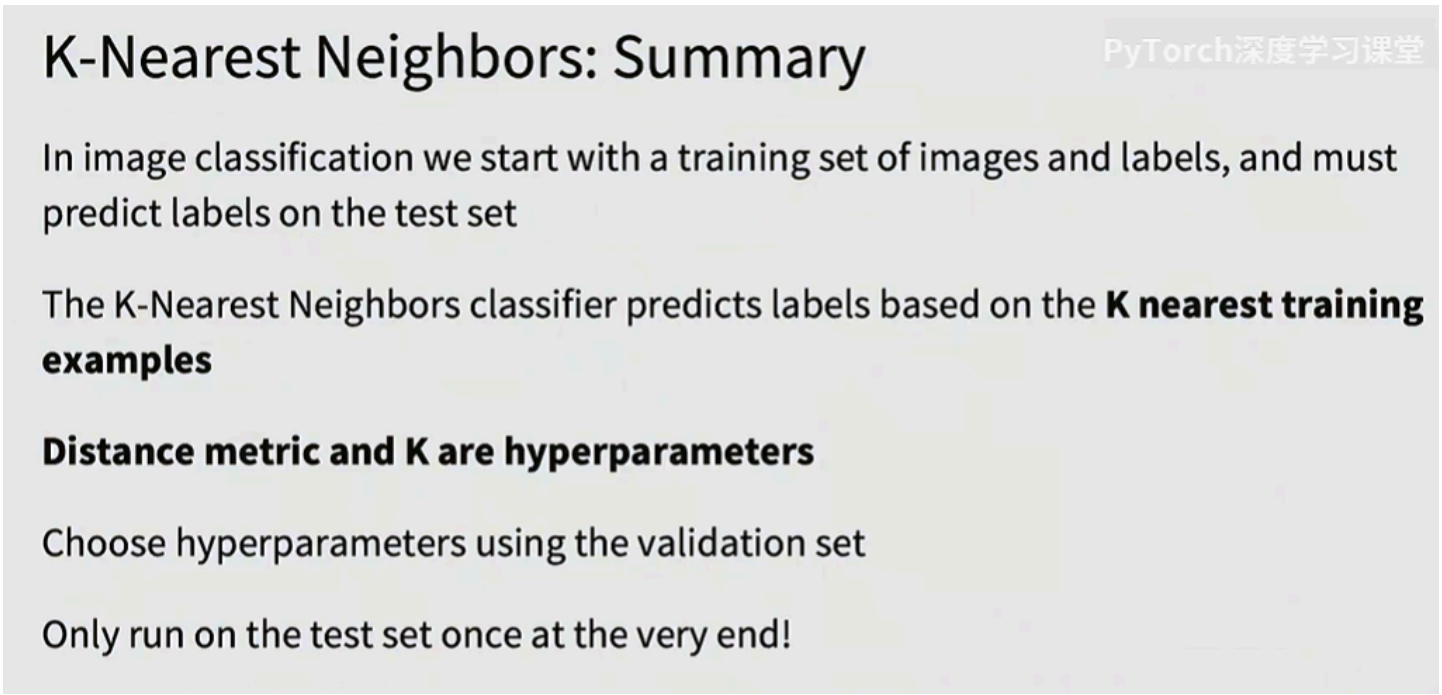
要利用validation来选取Hyperparameters。

怎么从test集分出一块作为validation集又是一个问题？于是又提出cross-validation

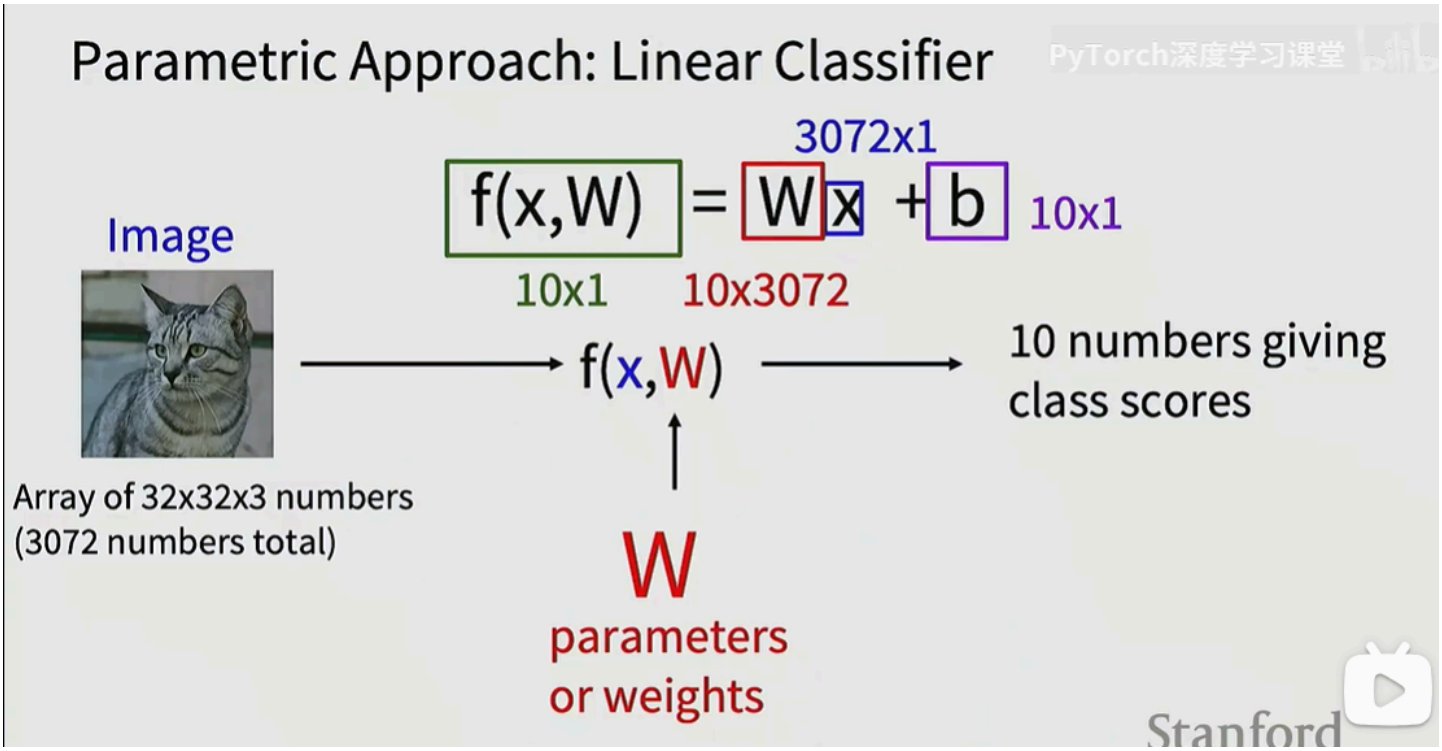


注意数据量很大时不会这么做，计算成本太高。

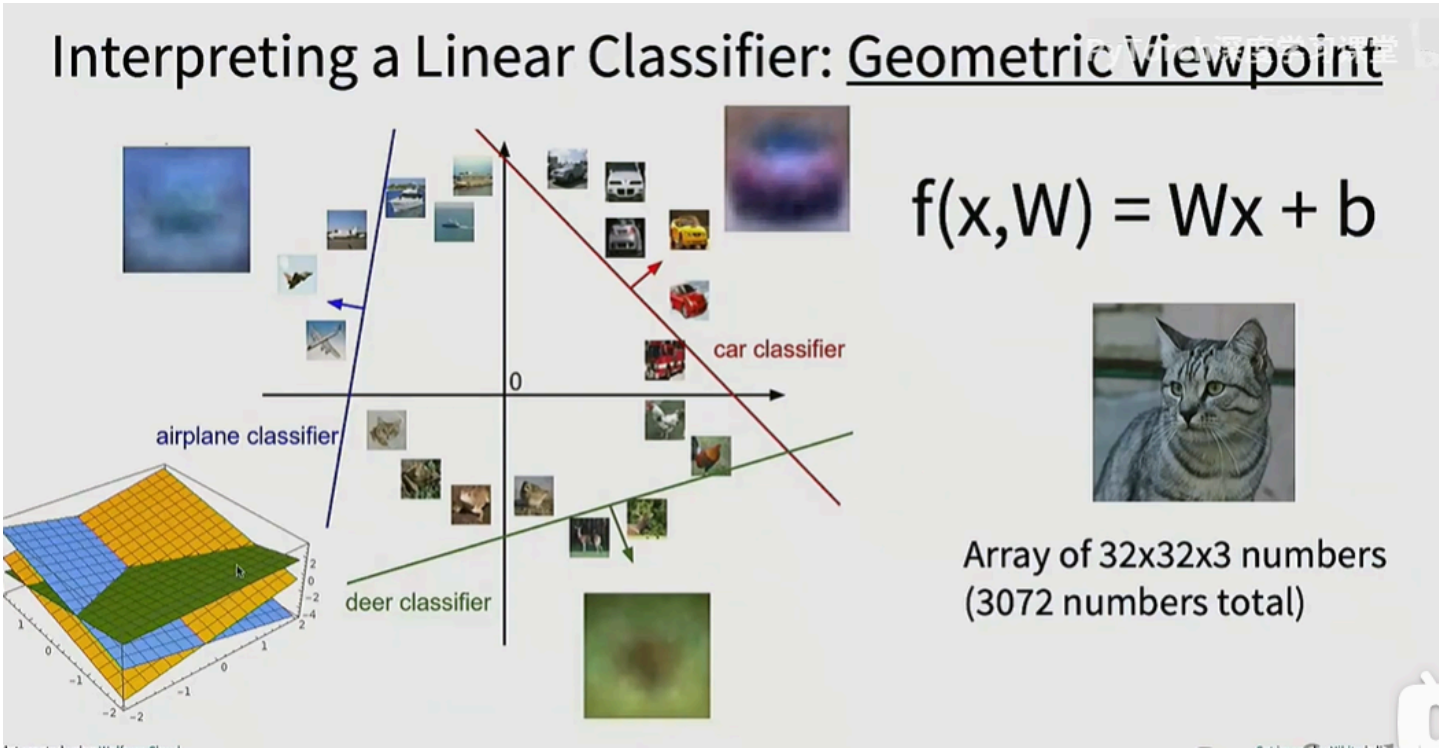
Summary



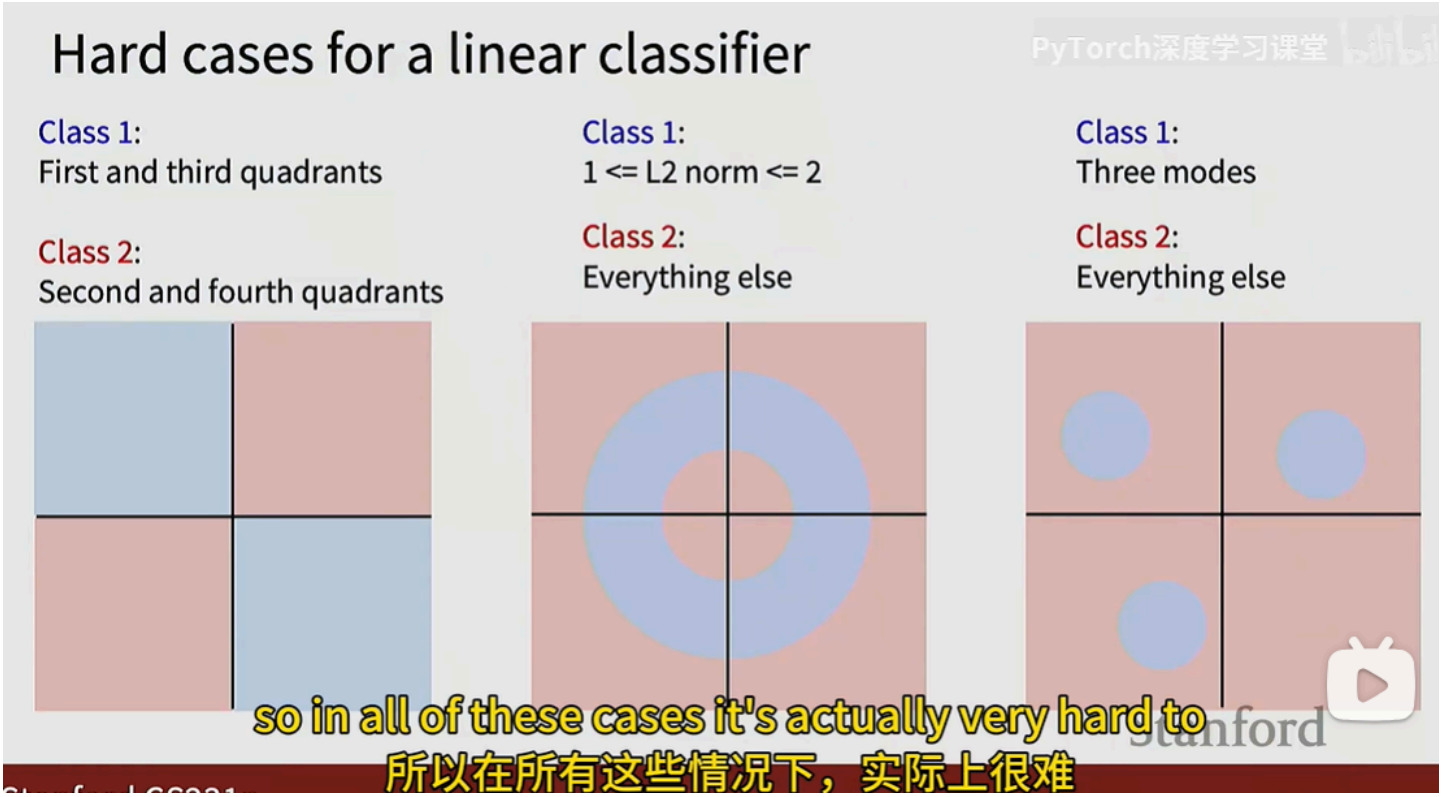
Linear Classifier



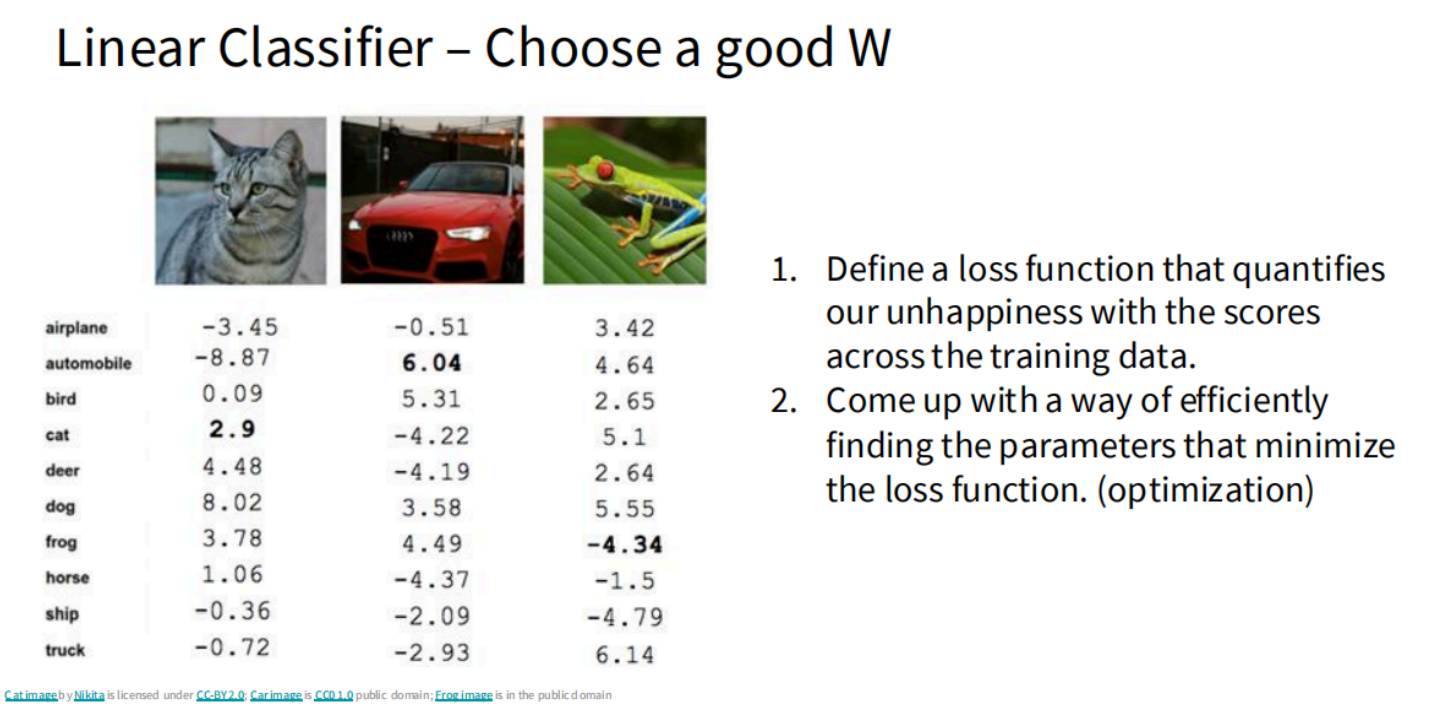
几何解释：找到超平面把数据分隔开



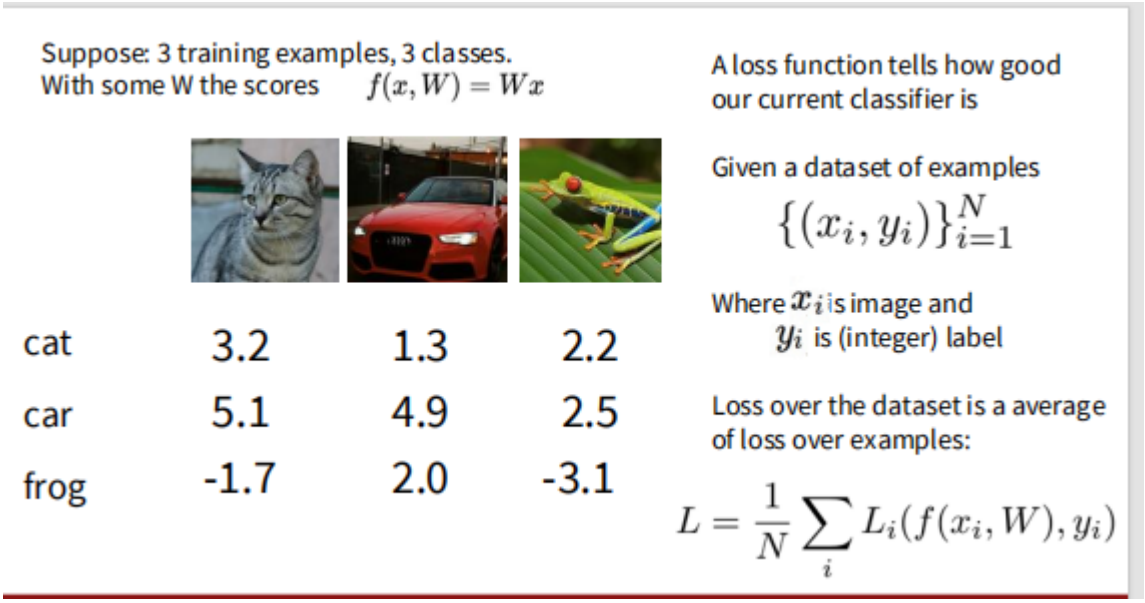
线性分类显然难以解决以下问题：



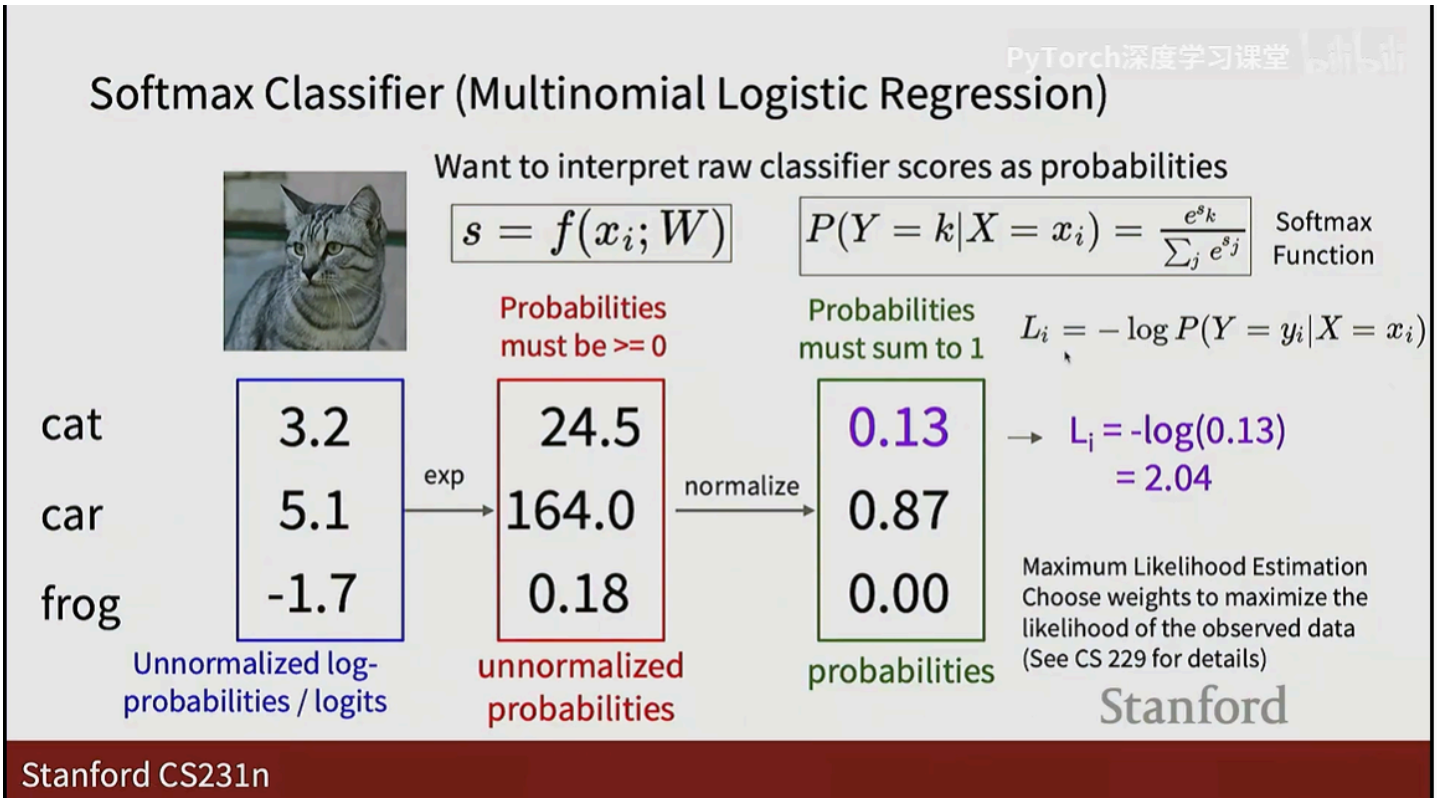
对于Linear Classifier，本质就是要找到W矩阵使得输入x，能使得结果在正确的分类上分数最大。那W怎么确定？刚开始就是随机初始化了，然后再通过某种方式不断调整。



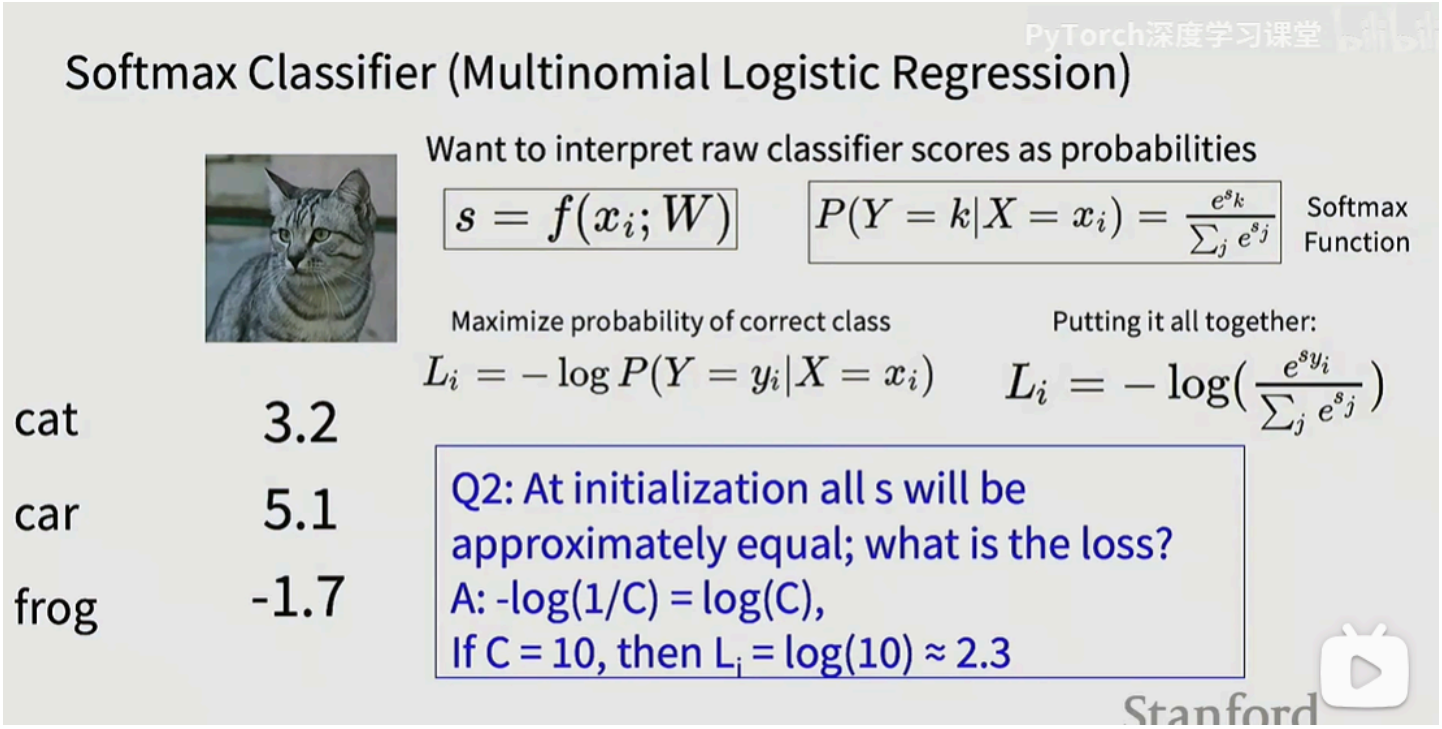
损失函数则用于描述该模型的结果和真实结果之间的差距。



显然损失函数有很多种，下面提到的Softmax Classifier（**依然是Linear classifier，虽然过程中有非线性的操作，但本质上决策边界依然是线性的**），则用了-log(softmax function),softmax可以把Wx+b算出来的分数转化为0~1看成概率，并且最理想的情况就应该是一张图片被分类到正确的类概率为1，其他类全为0，而-log（1）刚好为0，所以我们只需要最小化损失函数L即可



这里L_i最小到0，最大到正无穷。注意$L_i = -\log P(Y = y_i|X = x_i)$,只计算了正确分类所对应的值！



这里给出了一个验证自己代码是否正确的方式，如果刚开始初始化时把参数基本设为基本相同的，那L_i应该就在2.3左右

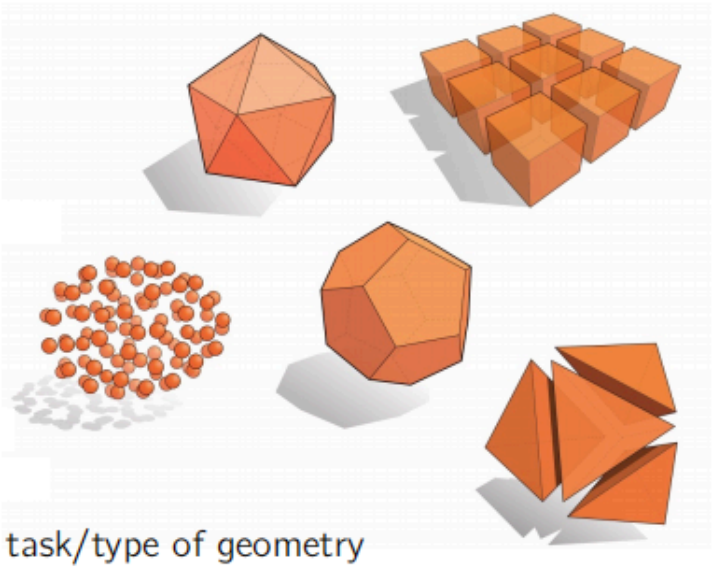
另外，单个样本的交叉熵就等于L_i
并且**交叉熵=熵＋KL散度**，而对于分类问题，正确分布是one-hot型的，熵很低，基本为0，所以可以说认为任务就是最小化KL散度

Lecture 3D version

how to represent 3D?

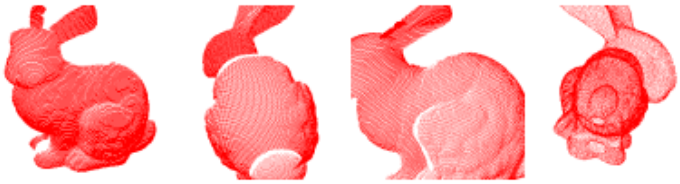
Many Ways to Represent Geometry

- Explicit
  - Point cloud
  - Polygon mesh
  - Subdivision, NURBS
  - ...
- Implicit
  - Lever sets
  - Algebraic surface
  - Distance functions
  - ...
- Each choice best suited to a different task/type of geometry

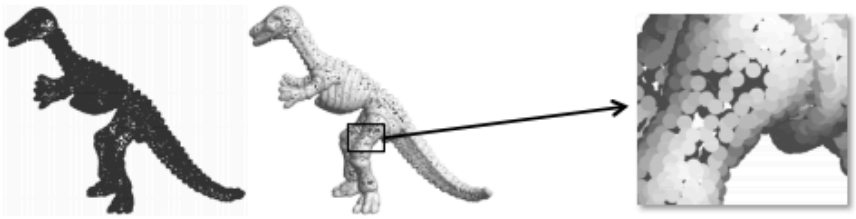


Slide credit: Ren Ng

Point Clouds



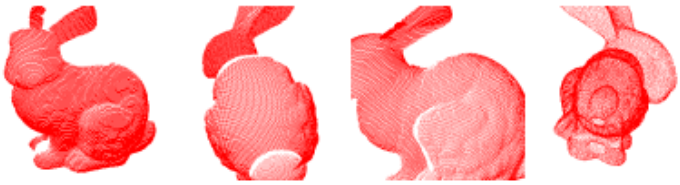
- Simplest representation: **only points**, no connectivity
- Collection of (x, y, z) coordinates, possibly with normal
- Points with orientation are called **surfels**



Shading needs normals!

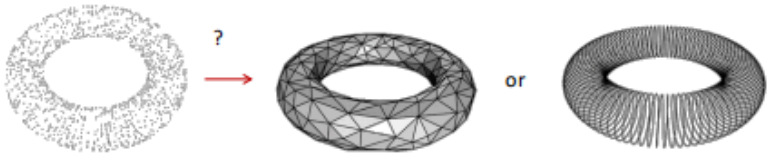
point clouds advantages and disadvantages

Point Clouds



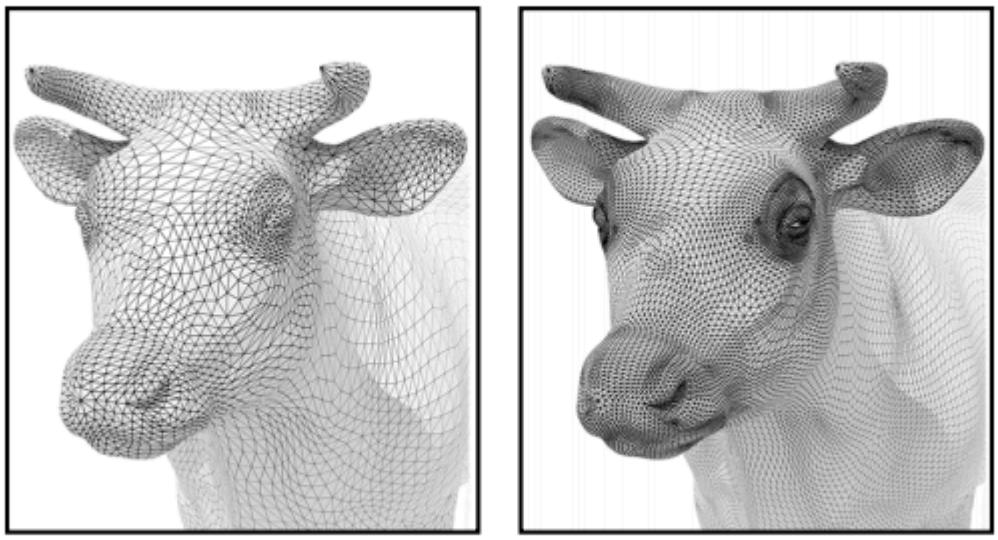
- Easily represent any kind of geometry
- Useful for large datasets
- Difficult to draw in undersampled regions

- Other limitations:
  - No simplification or subdivision
  - No direction smooth rendering
  - No topological information



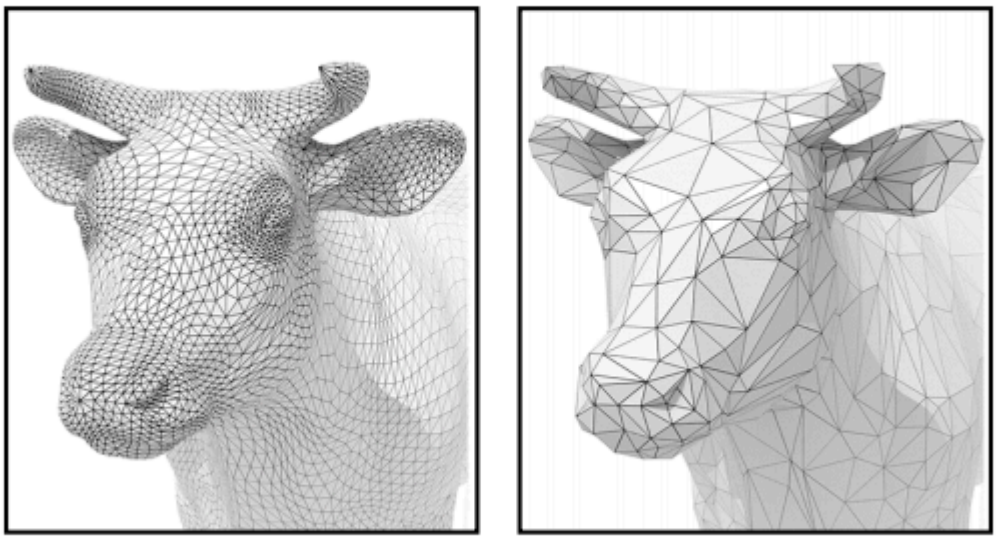
Polygonal Meshes

Mesh Upsampling - Subdivision



Increase resolution via interpolation

Mesh Downsampling - Simplification



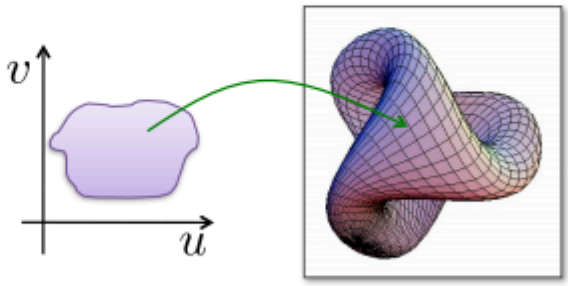
Decrease resolution; try to preserve shape/appearance

Parametric Representation

Parametric Representation

Range of a function $f : X \to Y, X \subseteq \mathbb{R}^m, Y \subseteq \mathbb{R}^n$

Surface in 3D: $m = 2, n = 3$

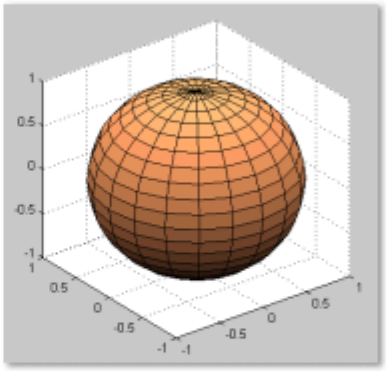


$$s(u, v) = (x(u, v), y(u, v), z(u, v))$$

Parametric Surfaces

Sphere in 3D

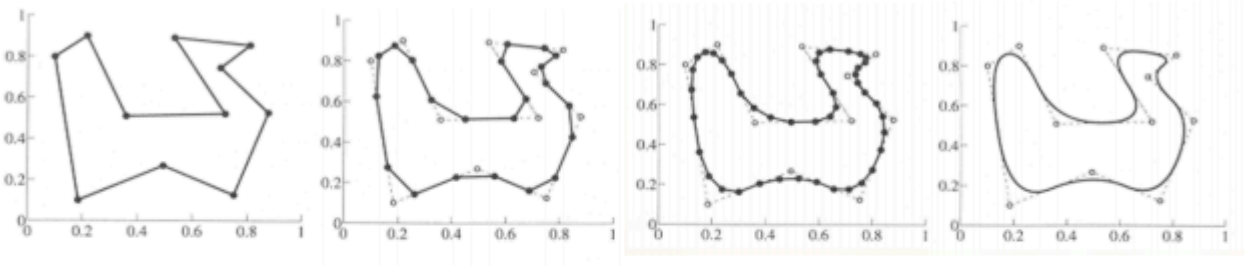
$$s : \mathbb{R}^2 \to \mathbb{R}^3$$



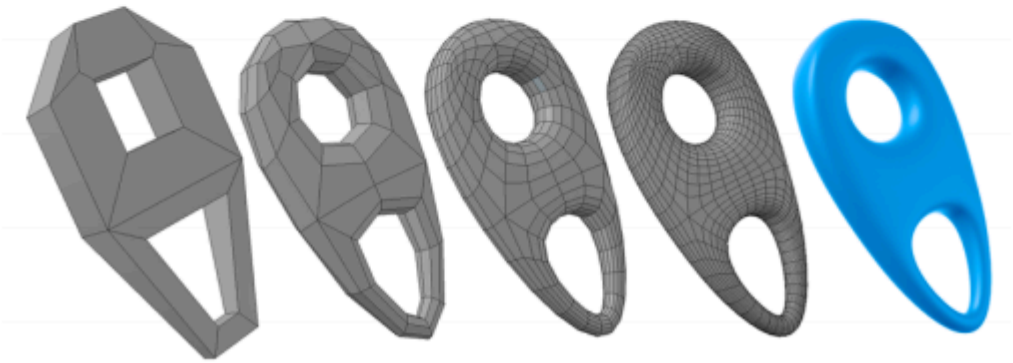
$$s(u, v) = r\,(\cos(u)\cos(v), \sin(u)\cos(v), \sin(v))$$

$$(u, v) \in [0, 2\pi) \times [-\pi/2, \pi/2]$$

Subdivision Curves/Surfaces



Slide cribbed from Keenan Crane, cribbed from Don Fussell.

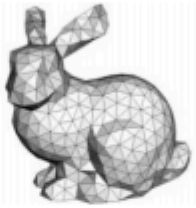


Shape Representations

Explicit

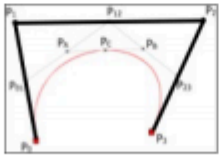
Non-parametric

Points

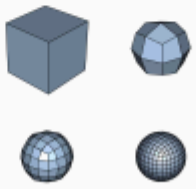


Meshes

Parametric



Splines



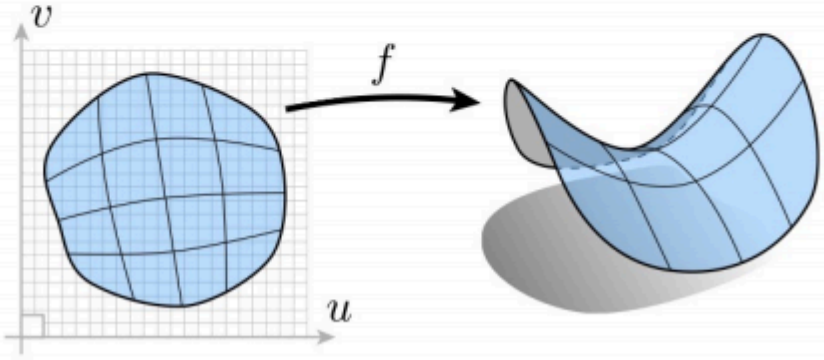
Subdivision Surfaces

对于显示参数化表达式，是容易采样的，随便给定几个自变量，带入方程计算函数值即可。但任意给定一个点，很难判断这个点在图形内部还是外部（知道点在内外部对于渲染的时候是重要的！）。隐式参数化则刚好相反。

"Explicit" Representations of Geometry

All points are given directly.

Generally: $f : \mathbb{R}^2 \to \mathbb{R}^3; (u, v) \mapsto (x, y, z)$

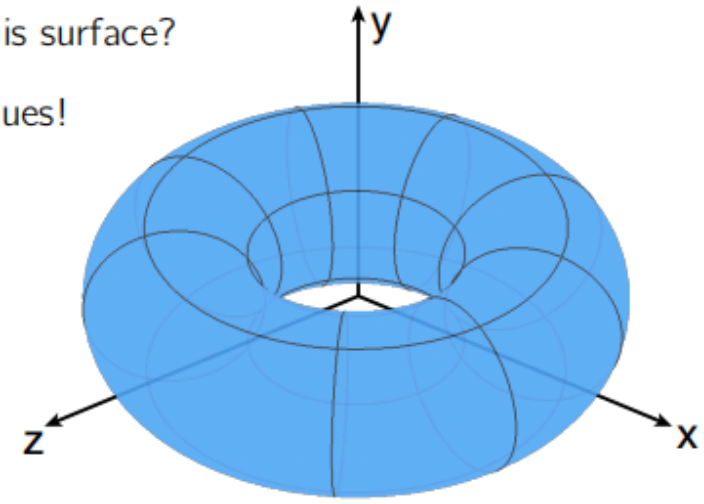


Explicit Surface – Sampling Is Easy

$$f(u,v) = ((2+\cos u)\cos v, (2+\cos u)\sin v, \sin u)$$

What points lie on this surface?

Just plug in (u, v) values!



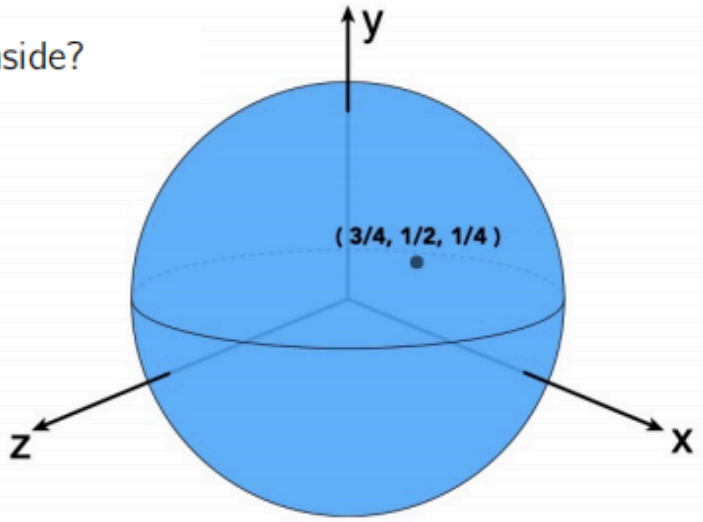
Explicit representations make some tasks easy.

Slide credit: R

Explicit Surface – Inside/Outside Test Hard

$$f(u,v) = (\cos u \sin v, \sin u \sin v, \cos v)$$

Is $(3/4, 1/2, 1/4)$ inside?



Some tasks are hard with explicit representations.

Slide credit: R

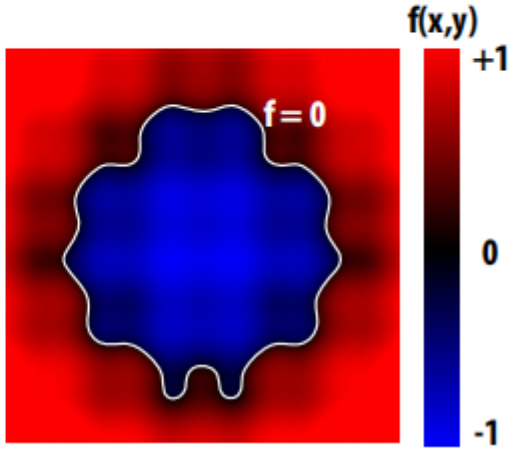
"Implicit" Representations of Geometry

Based on classifying points

- Points satisfy some specified relationship.

E.g., sphere: all points in 3D, where $x^2 + y^2 + z^2 = 1$

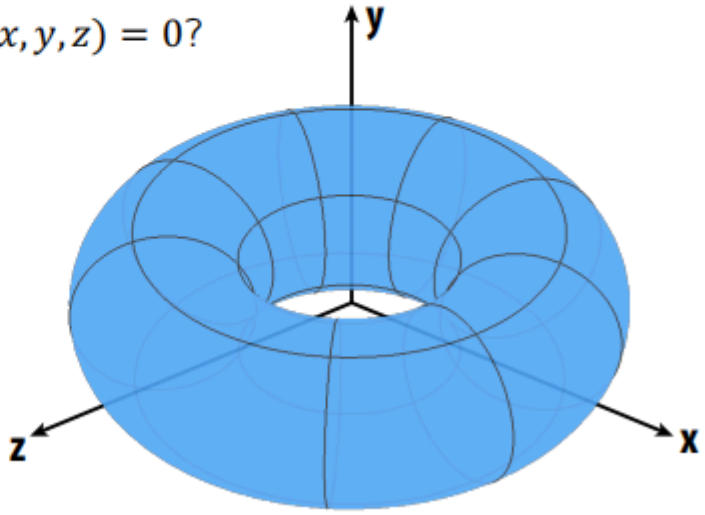
More generally, $f(x, y, z) = 0$



Implicit Surface – Sampling Can Be Hard

$$f(x, y, z) = (2 - \sqrt{x^2 + y^2})^2 + z^2 - 1$$

What points lie on $f(x, y, z) = 0$?



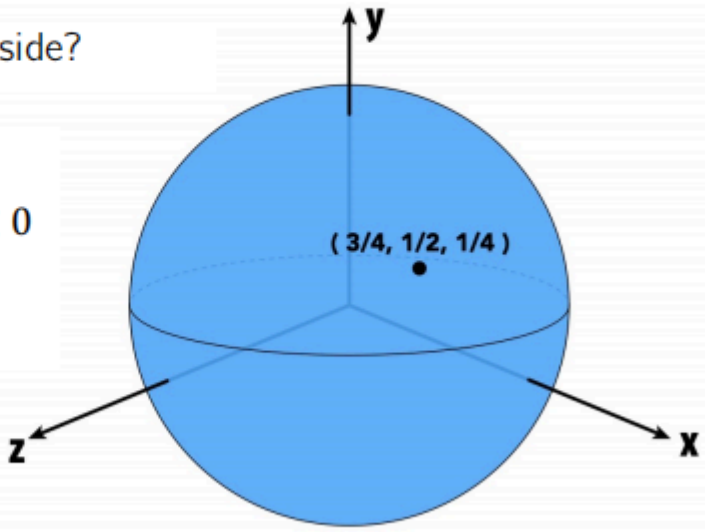
Some tasks are hard with implicit representations.

Implicit Surface – Inside/Outside Tests Easy

$f(x,y,z) = x^2 + y^2 + z^2 - 1$

Is $(3/4, 1/2, 1/4)$ inside?

Just plug it in:
$f(x,y,z) = -1/8 < 0$
Yes, inside.



Implicit representations make some tasks easy.

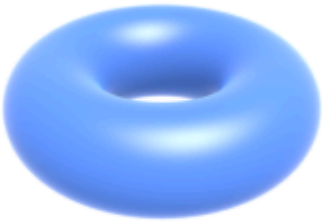
Slide credit:

对于复杂的图像，需要用到布尔运算

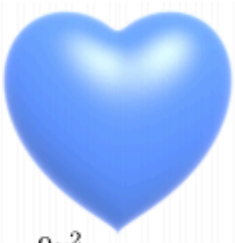
Algebraic Surfaces (Implicit)

Surface is zero set of a polynomial in x, y, z.

$x^2+y^2+z^2=1$



$(R - \sqrt{x^2+y^2})^2 + z^2 = r^2$



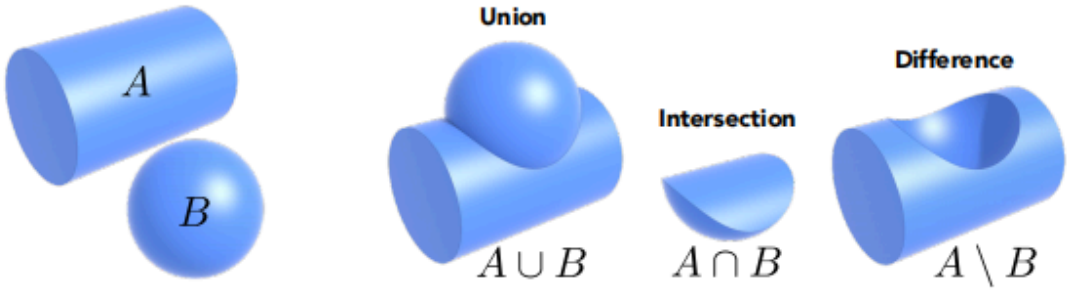
$(x^2 + \frac{9y^2}{4} + z^2 - 1)^3 = x^2 z^3 + \frac{9y^2 z^3}{80}$

More complex shapes?

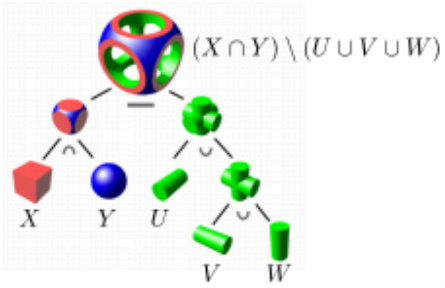
Slide cred

Constructive Solid Geometry (Implicit)

Combine implicit geometry via Boolean operations



Boolean expressions:



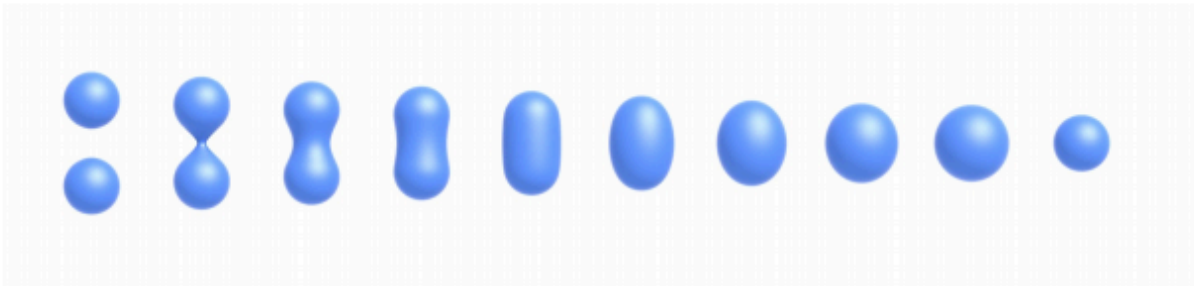
Slide credit: Re

其他隐式方法：

Distance Functions (Implicit)

Instead of Boolean, gradually blend surfaces together using

Distance functions:
Giving minimum distance (could be **signed** distance) from anywhere to object



Scene of Pure Distance Functions (Not Easy!)

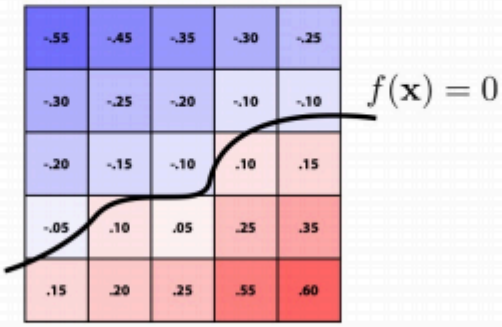
See http://iquilezles.org/www/material/nvscene2008/nvscene2008.htm

Level Set Methods (Implicit)

Implicit surfaces have some nice features (e.g., merging/splitting).

But hard to describe complex shapes in closed form

Alternative: store a grid of values approximating function

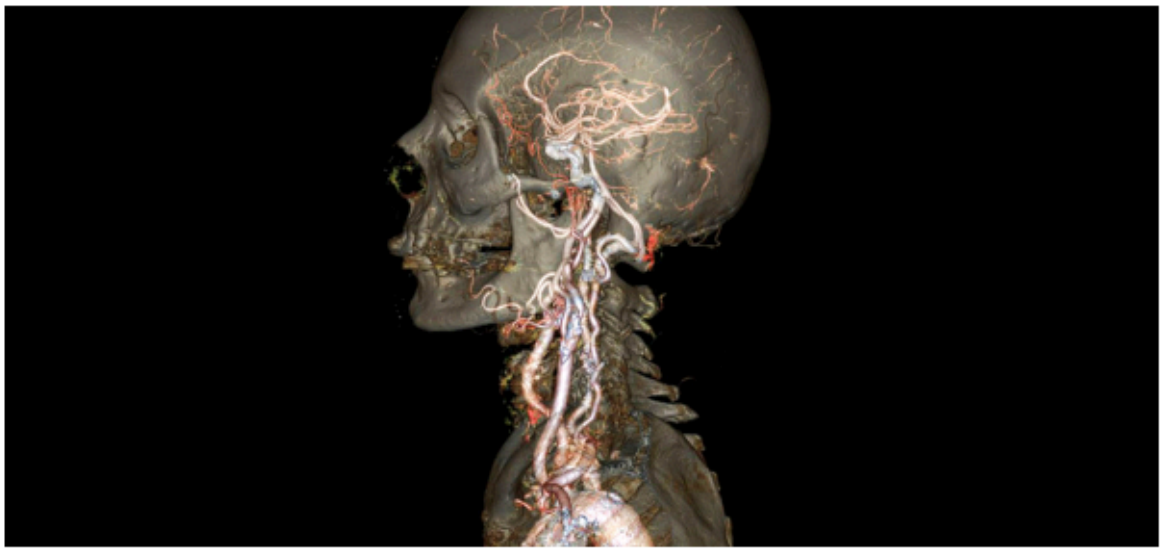


Surface is found where interpolated values equal zero.

Provides much more explicit control over shape (like a texture)

Slide credit:

Level Sets from Medical Data (CT, MRI, etc.)

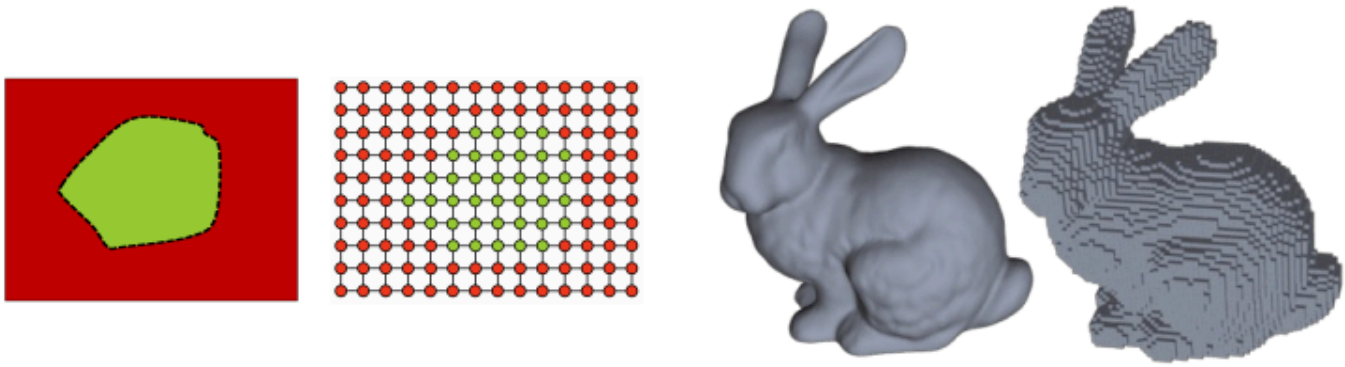


Level sets encode, e.g., constant tissue density

Slide credit: Ren

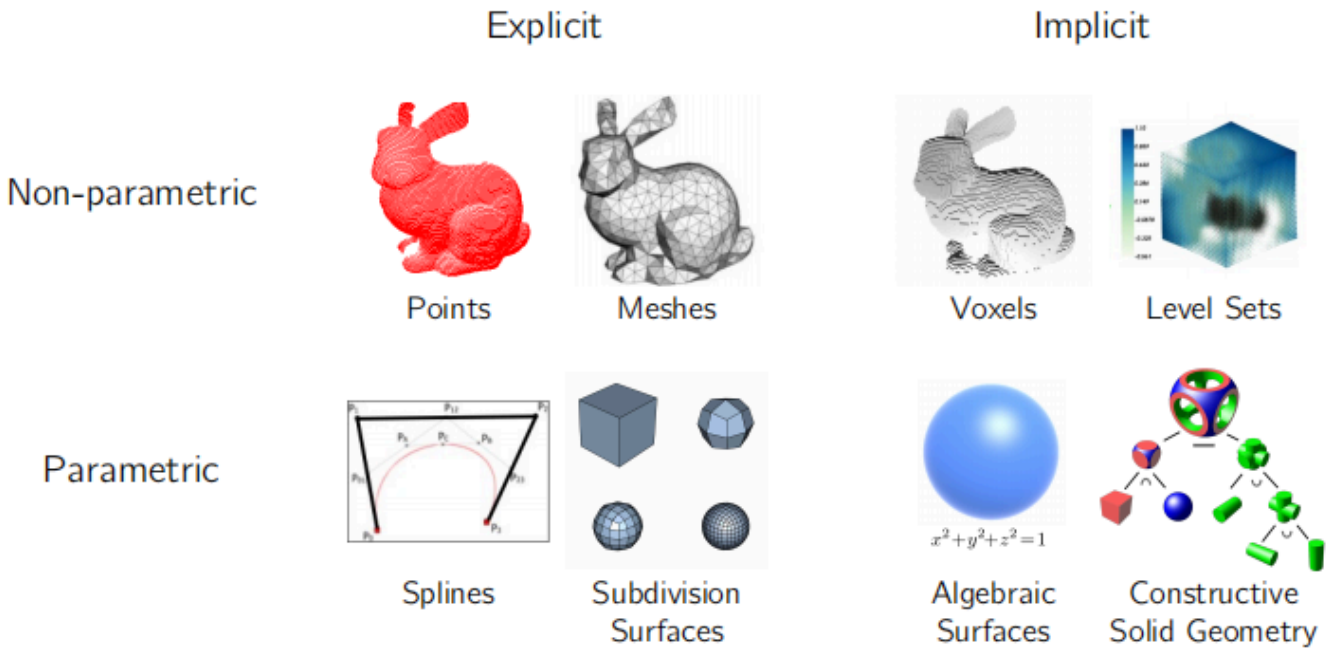
Related Representation: Voxels

- Binary thresholding the volumetric grid



总结：

Shape Representations



dataset：ShapeNet CO3D

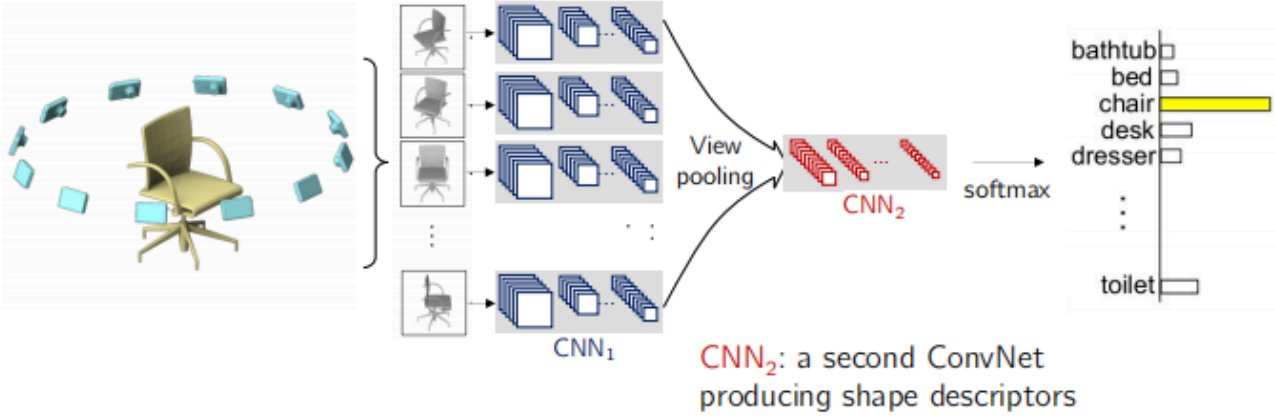
目前的tasks：

AI + Geometry: Tasks

- $P(S)$ or $P(S|c)$ --- Generative models
  - Learning (conditional) shape priors
  - Shape generation, completion, & geometry data processing
- $P(c|S)$ --- Discriminative models
  - Learning shape descriptors
  - Shape classification, segmentation, view estimation, etc.
- Joint modeling of 3D and 2D data
  - Large-scale 2D datasets & very good pretrained models
  - Differentiable projection/back-projection & differentiable/neural rendering
- Joint modeling of multi-modal data beyond visual (e.g., text)

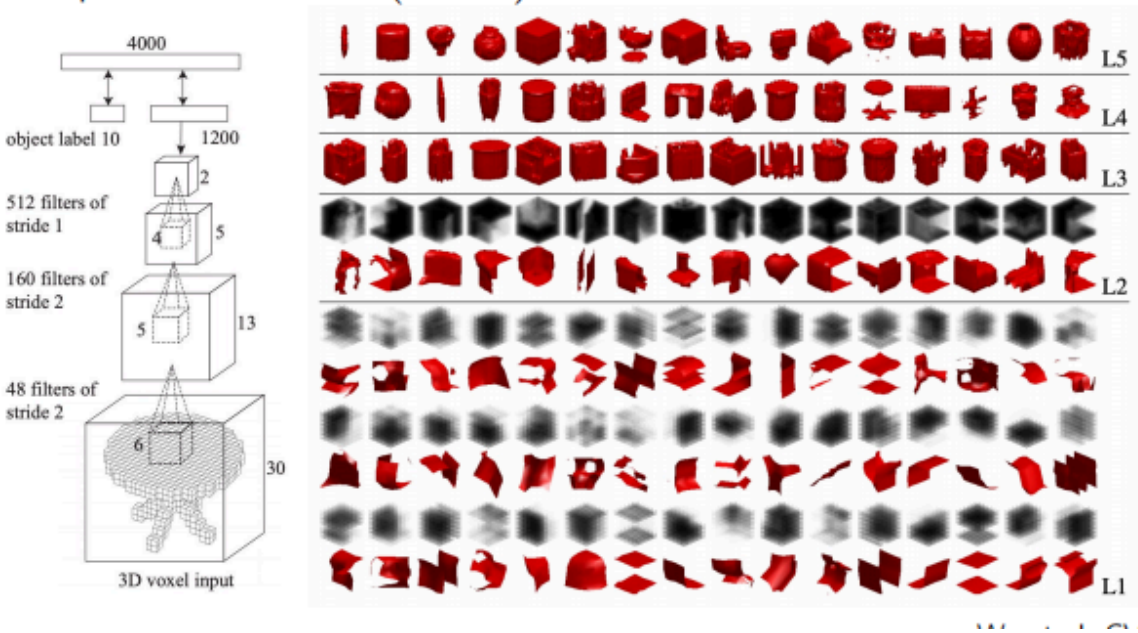
由于2d情况的已经很成熟，所以我们可以借鉴或者利用2d的情况

Multi-View CNN



Pixels -> Voxels

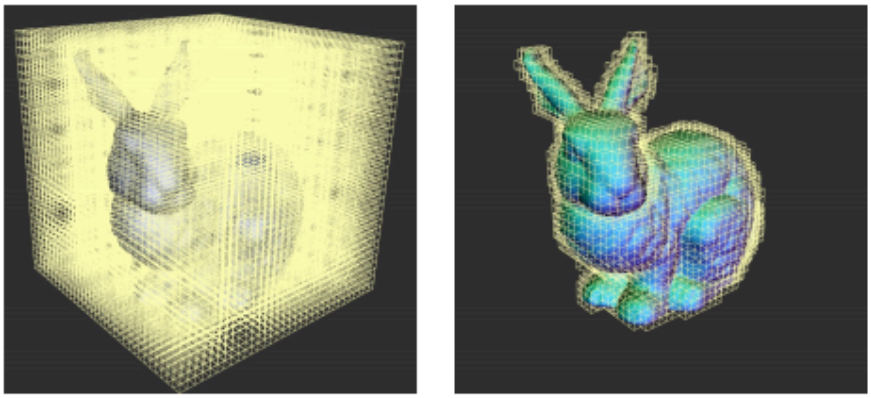
- 3D Conv Deep Belief Networks (CDBN)



其他方法：

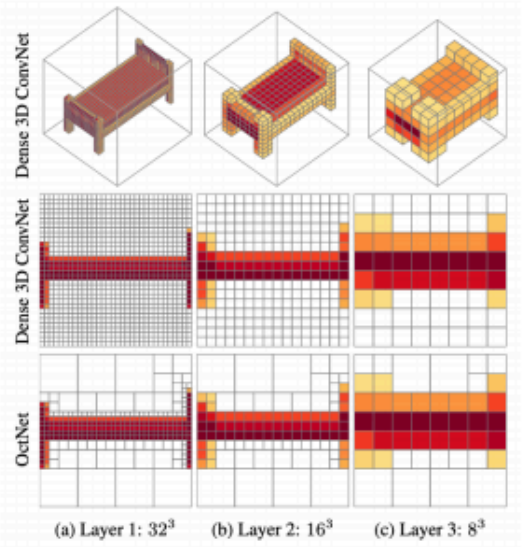
Octave Tree Representations

- Store the sparse surface signals
- Constrain the computation near the surface

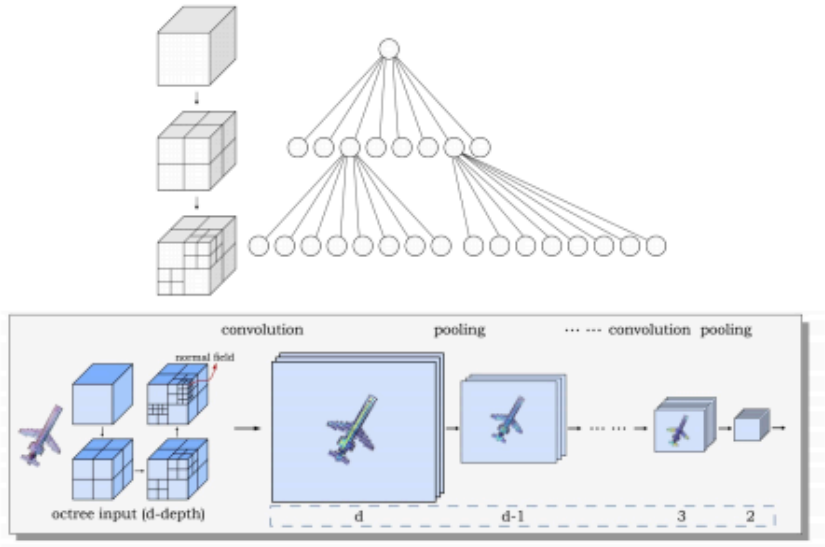


Slide Credit: Hao Su

Octree: Recursively Partition the Space

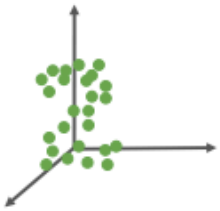


Riegler et al. OctNet. CVPR 2017



Wang et al. O-CNN. SIGGRAPH 2017

PointNet: Learning on Point Clouds



PointNet

Object Classification

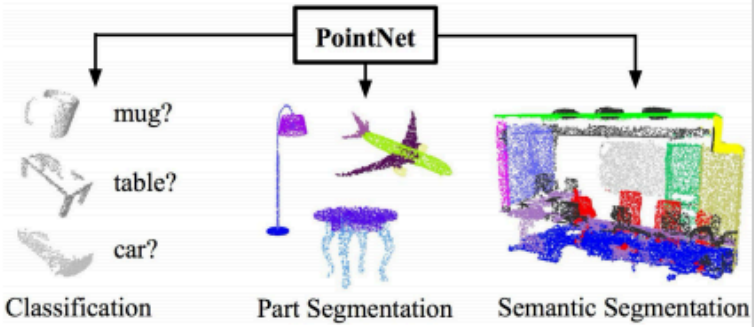
Object Part Segmentation

Semantic Scene Parsing

...

End-to-end learning for irregular point data

Unified framework for various tasks



Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. (CVPR'17)

Slide credit: He Wang

考虑物体的对称型，并且point cloud的特性

Invariances

The model has to respect key desiderata for point clouds:

Point Permutation Invariance

Point cloud is a set of unordered points

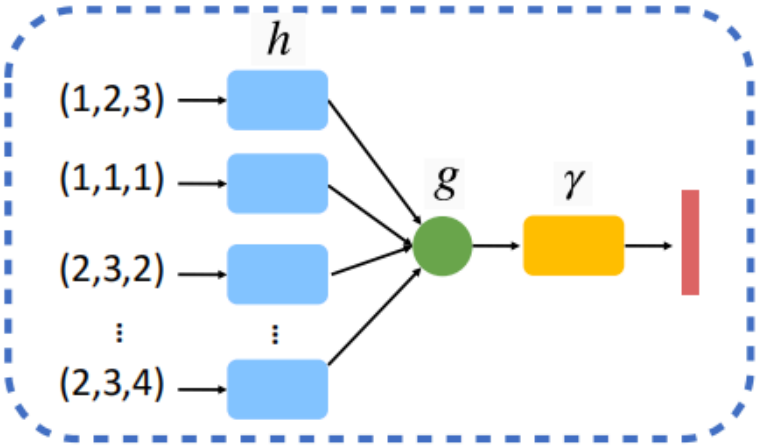
Sampling Invariance

Output a function of the underlying geometry and not the sampling

所以我们也可以考虑

Construct Symmetric Functions by NNs

$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



PointNet (vanilla)

Slide credit: He Wang

由于此时output是point cloud不再是分类的标签or体积块，不能用交叉熵做loss function 所以我们需要引入新的方式计算output的point cloud和真值的差异

Distance Metrics for Point Clouds

Chamfer distance We define the Chamfer distance between $S_1, S_2 \subseteq \mathbb{R}^3$ as:

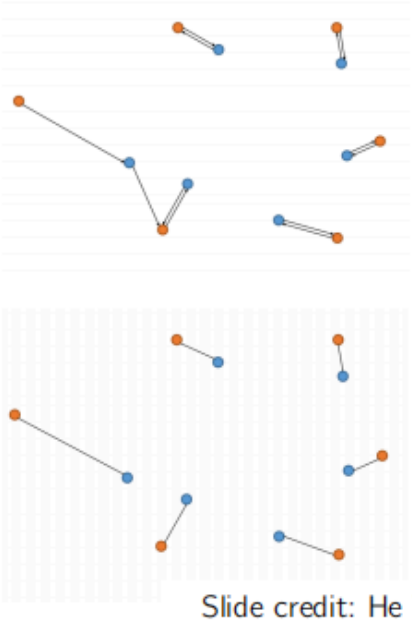
$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2$$

Earth Mover's distance Consider $S_1, S_2 \subseteq \mathbb{R}^3$ of equal size $s = |S_1| = |S_2|$. The EMD between A and B is defined as:

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

where $\phi : S_1 \to S_2$ is a bijection.

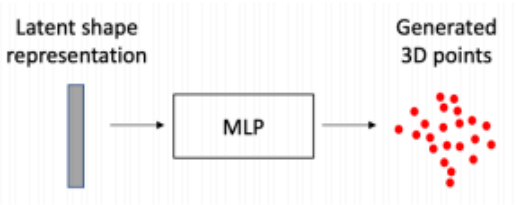
A Point Set Generation Network for 3D Object Reconstruction from a Single Image, CVPR 2016

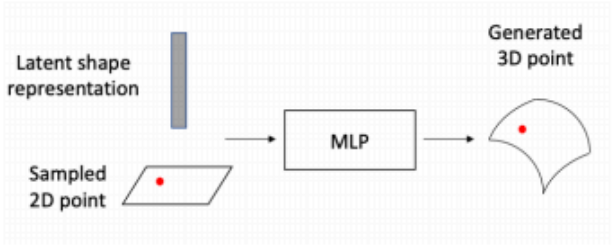


Slide credit: He Wang

AltasNet想获得更光滑的曲面

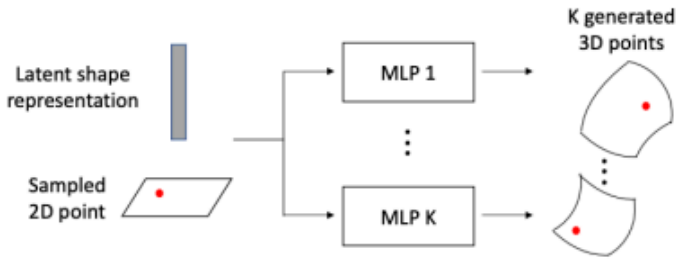
Parametric Decoder: AtlasNet





Given the output points form a smooth surface, enforce such a parametrization as input.

$\text{MLP}(z, u, v)$ -> point



Groueix et al. CVPR 2018

后续还涉及到了Deep Implicit Functions，Reconstruction & Novel View Synthesis with NeRF等