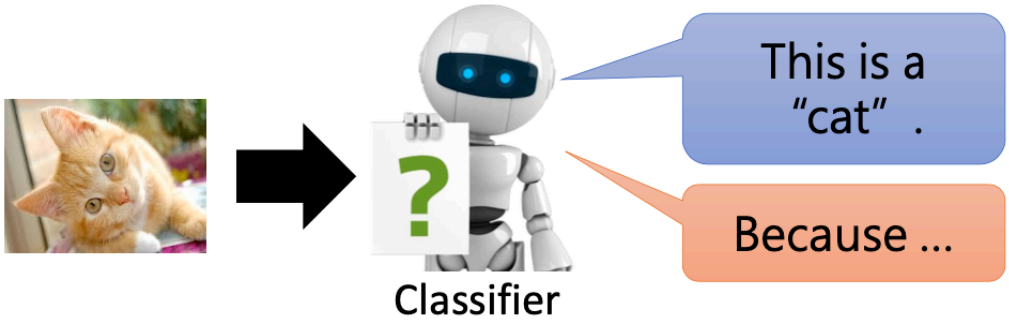
Introduction

机器不仅要告诉我们结果cat，还要告诉我们为什么



Local Explanation

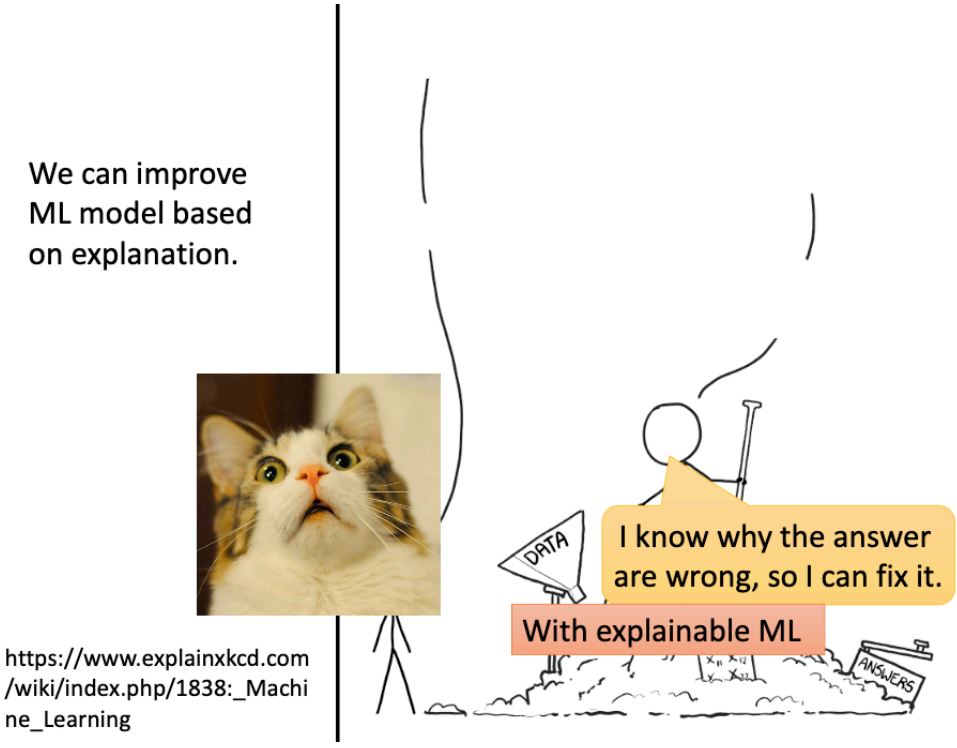
Why do you think *this image* is a cat?

Global Explanation

What do you think a "cat" looks like?

Why we need Explainable ML?

我们不仅需要机器结果的精确度，还需要进行模型诊断，看机器学习得怎么样；有的任务精确度很高，但实际上机器什么都没学到



有模型诊断后，我们就可以根据模型诊断的结果再来调整我们的模型

Interpretable v.s. Powerful

- Some models are intrinsically interpretable.
  - For example, linear model (from weights, you know the importance of features)
  - But ……. not very powerful.
- Deep network is difficult to interpretable.
  - Deep network is a black box.

Because deep network is a black box, we don't use it.

削足適履 ☹

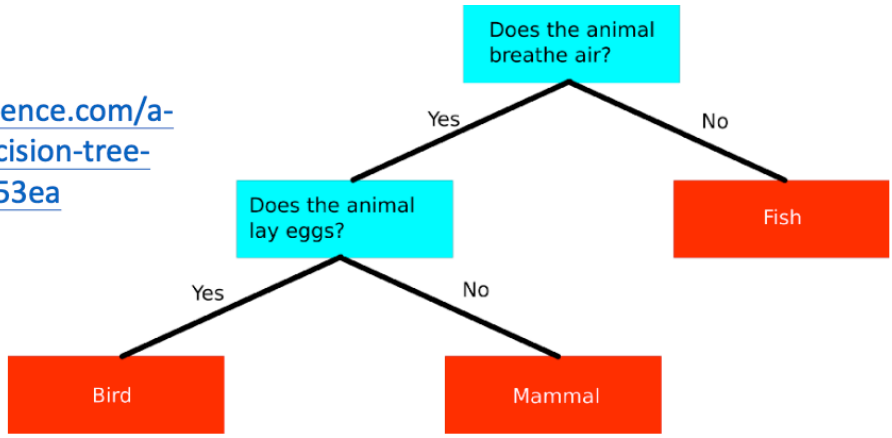
  - But it is more powerful than linear model …

Let's make deep network interpretable.

那么有没有model是Interpretable，也是powerful的呢？

决策树可以interpretable，也是比较powerful的；对于第一个分支节点，“这些动物呼吸空气吗？”，就包含了interpretable的信息

Source of image: https://towardsdatascience.com/a-beginners-guide-to-decision-tree-classification-6d3209353ea



当分支特别多的时候，决策树的表现也会很差

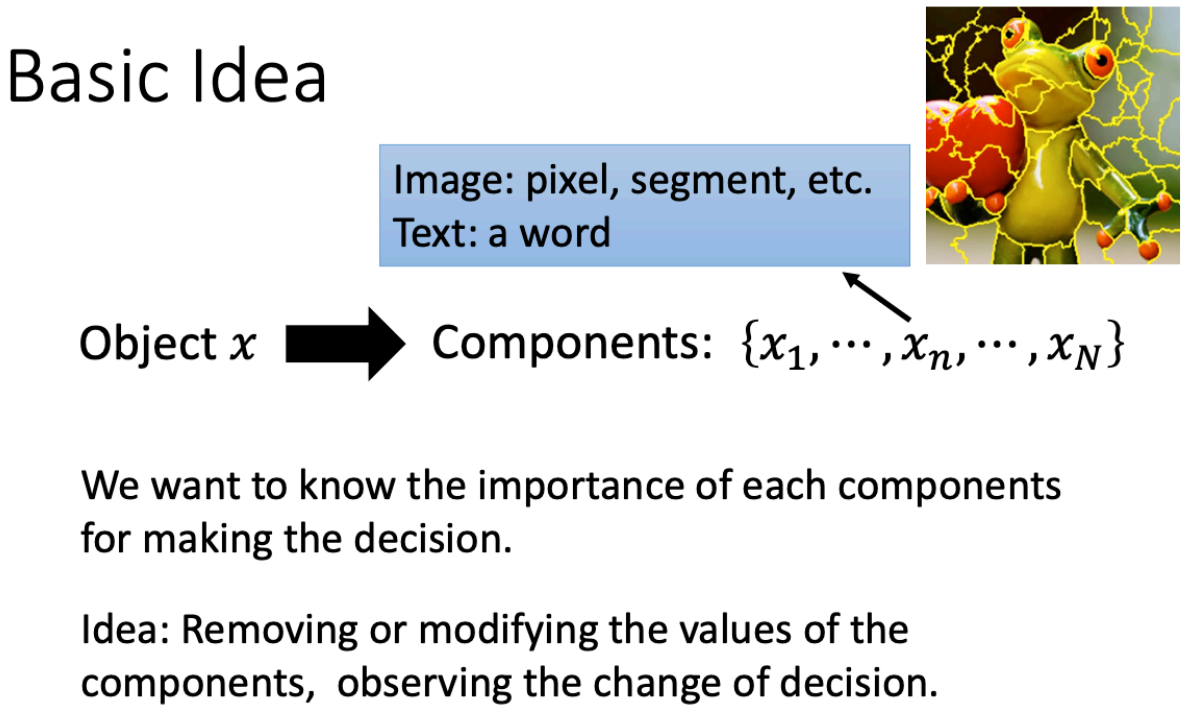
- A tree can still be terrible!
- We use a forest!

Local Explanation

Basic Idea

对于输入的x，我们将其分成components $\{x_1, \ldots, x_n, \ldots x_N\}$，每个component由一个像素，或者一小块组成

我们现在的目标是知道每个component对making the decision的重要性有多少，那么我们可以通过remove或者modify其中一个component的值，看此时的decision会有什么变化

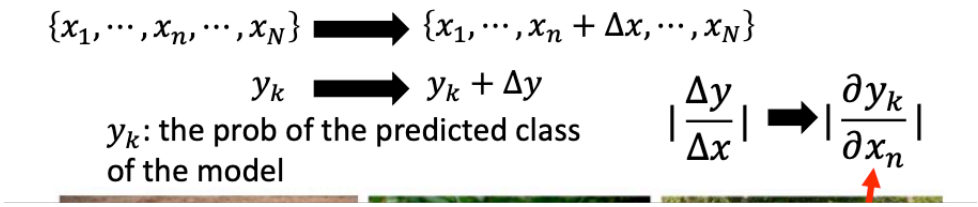


把灰色方块放到图像中，覆盖图像的一小部分；如果我们把灰色方块放到下图中的红色区域，那么对解释的结果影响不大，第一幅图还是一只狗

还有另一种方法

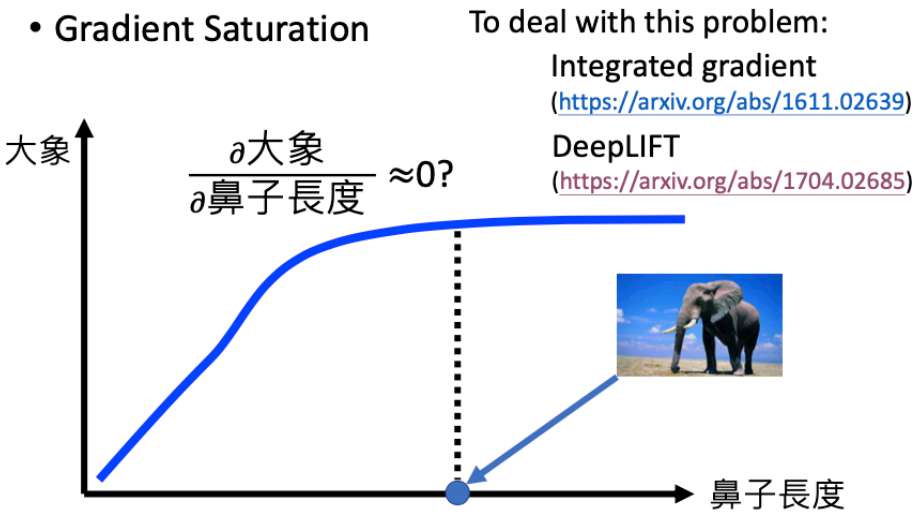
对于输入的$\{x_1, \ldots, x_n, .., x_N\}$，对于其中的某个关键的pixel x_n加上Δx，这个pixel对我们识别这是不是一只狗具有很重要的作用

那么我们就可以用$\frac{\Delta y}{\Delta x}$来表示这个小小的扰动对y的影响，可以通过$\frac{\partial y_k}{\partial x_n}$来进行计算，表示$y_k$对$x_n$的偏微分，最后取绝对值，表示某一个pixel对现在y影响的大小

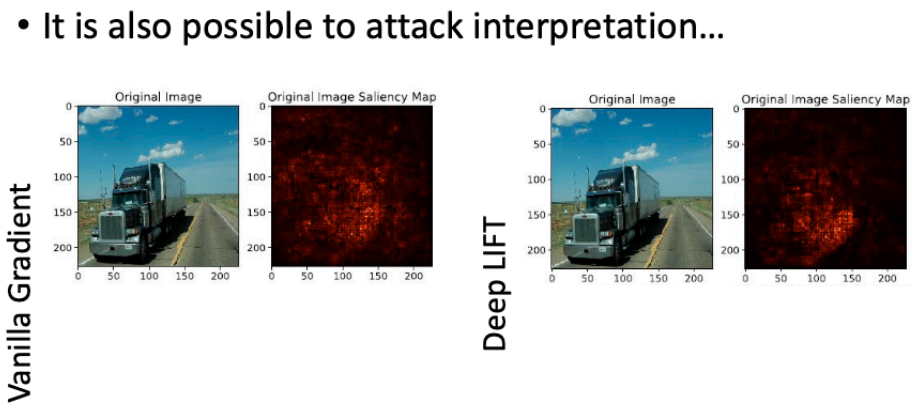


在上图中，下半部分由3幅图saliency map，亮度越大，绝对值就越大，亮度越大的地方就表示该pixel对结果的影响越大

Limitation of Gradient based Approaches



Attack Interpretation



The noise is small, and do not change the classification results.

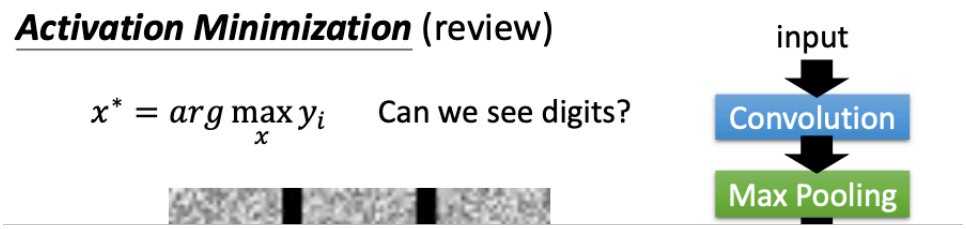
Global Explanation

Interprete the whole Model

Activation Minimization (review)

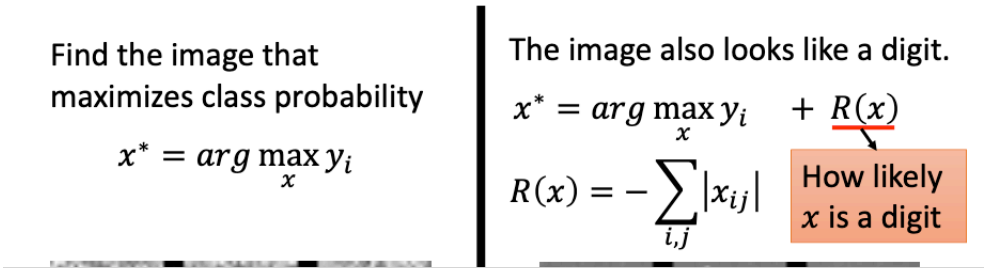
让我们先review一下activation minimization，现在我们的目标是找到一个x^*，使得输出的值y_i最大

我们可以加入一些噪声，加上噪声后人并不能识别出来，但机器可以识别出来，看出来下图中的噪声是0 1 2 3 4 5 6 7 8



之前我们的目标是找到一个image，使得输出的y达到最大值；现在我们的目标不仅是找到x使输出y达到最大值，还需要把image变得更像是一个digit，不像左边那个图，几乎全部的像素点都是白色，右边的图只有和输出的digit相关的pixel才是白色

这里我们通过加入了一个新的限制$R(x)$来实现，可以表示图像和digit的相似度



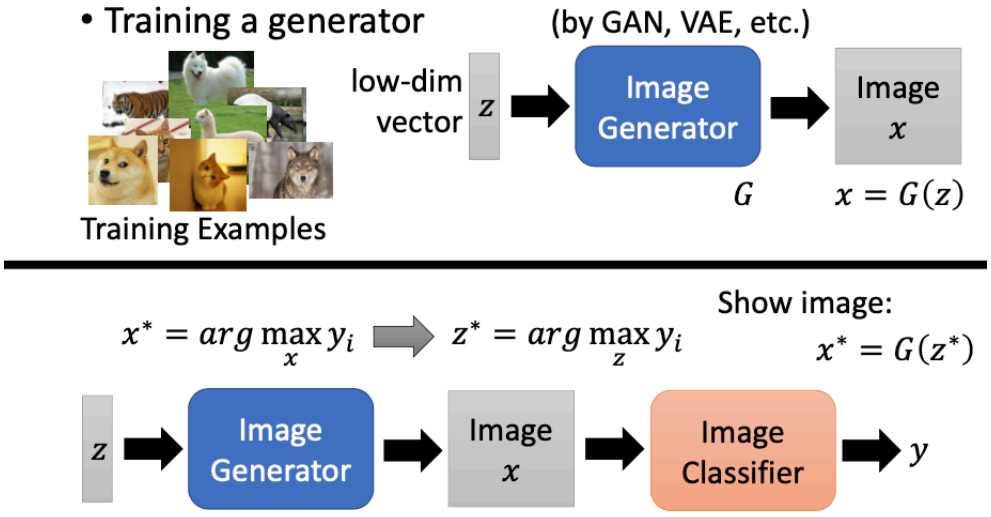
Constraint from Generator

如下图所示，我们输入一个低维的vector z到generator里面，输出Image x；

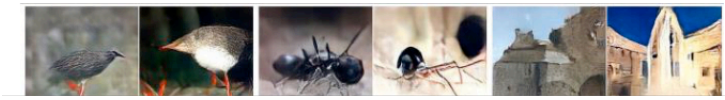
现在我们将生成的Image x再输入Image classifier，输出分类结果y_i，那么我们现在的目标就是找到z^*，使得属于那个类别的可能性y_i最大

$$z^* = arg\ maxy_i$$

找到最好的z^*，再输入Generator，根据$x^* = G(z^*)$得出x^*，产生一个好的Image

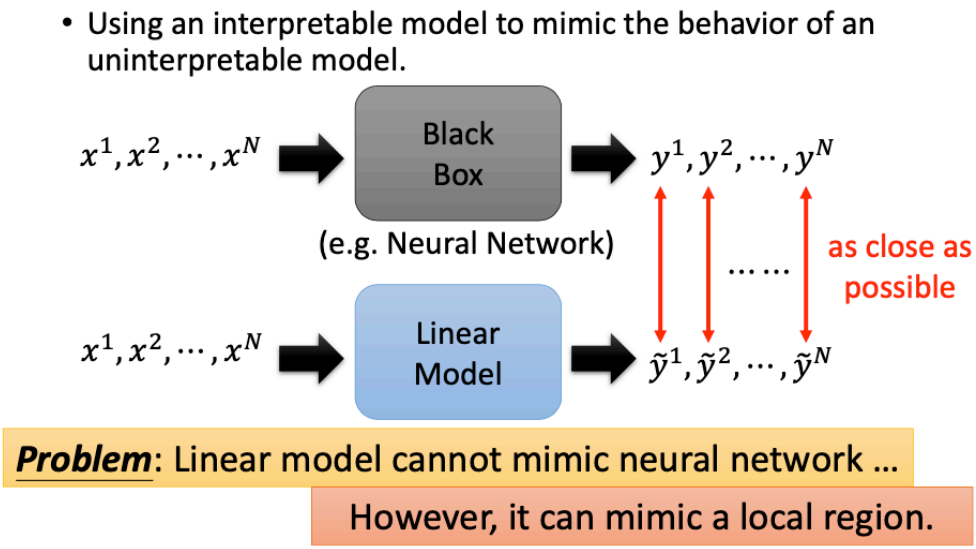


结果展示。现在你问机器蚂蚁长什么样子呢？机器就会给你画一堆蚂蚁的图片出来，再放到classifier里面，得出分类结果到底是火山还是蚂蚁



Using a model to explain another

现在我们使用一个interpretable model来模仿另外一个uninterpretable model；下图中的Black Box为uninterpretable model，比如Neural Network，蓝色方框是一个interpretable model，比如Linear model；现在我们的目标是使用相同的输入$x^1, x^2, \ldots, x^N$，使linear model和Neural Network有相近的输出



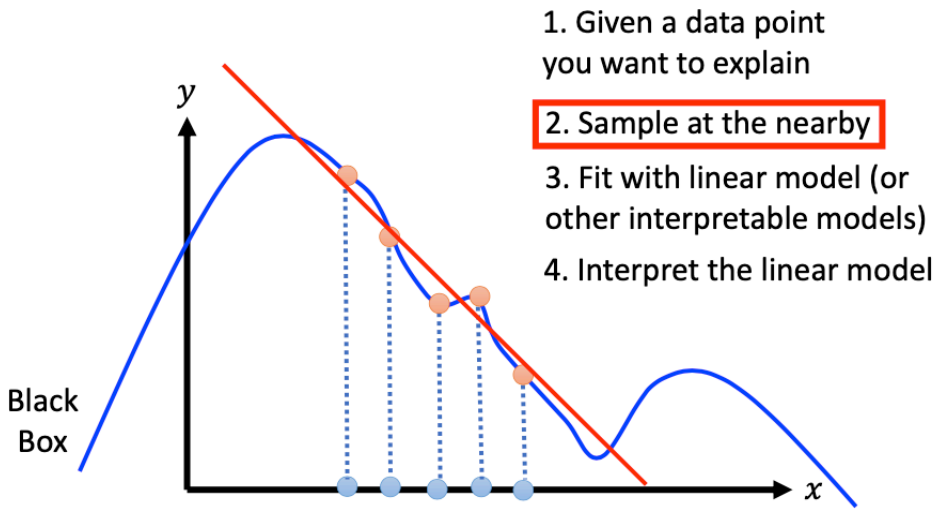
实际上并不能使用linear model来模拟整个neural network，但可以用来模拟其中一个local region

Local Interpretable Model-Agnostic Explanations (LIME)

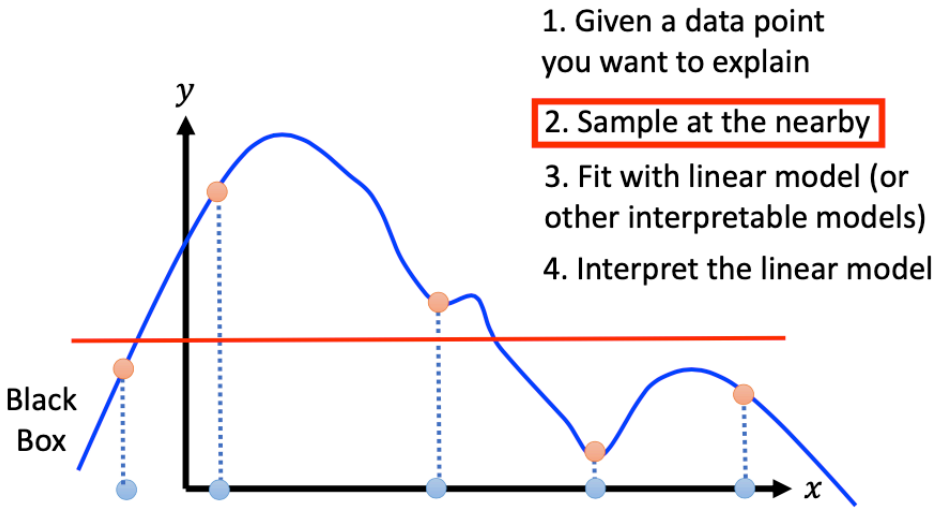
General

下图中input为x，output为y，都是一维的，表示Black Box中x和y的关系，由于我们并不能用linear model来模拟整个neural network，但可以用来模拟其中一个local region

1. 首先给出要explain的point，代入black box里面
2. 在第三个蓝色point（我们想要模拟的区域）周围sample附近的point，nearby的区域不同，结果也会不同
3. 使用linear model来模拟neural network在这个区域的行为
4. 得知了该区域的linear model之后，我们就可以知道在该区域x和y的关系，即x越大，y越小，也就interpret了原来的neural network在这部分区域的行为

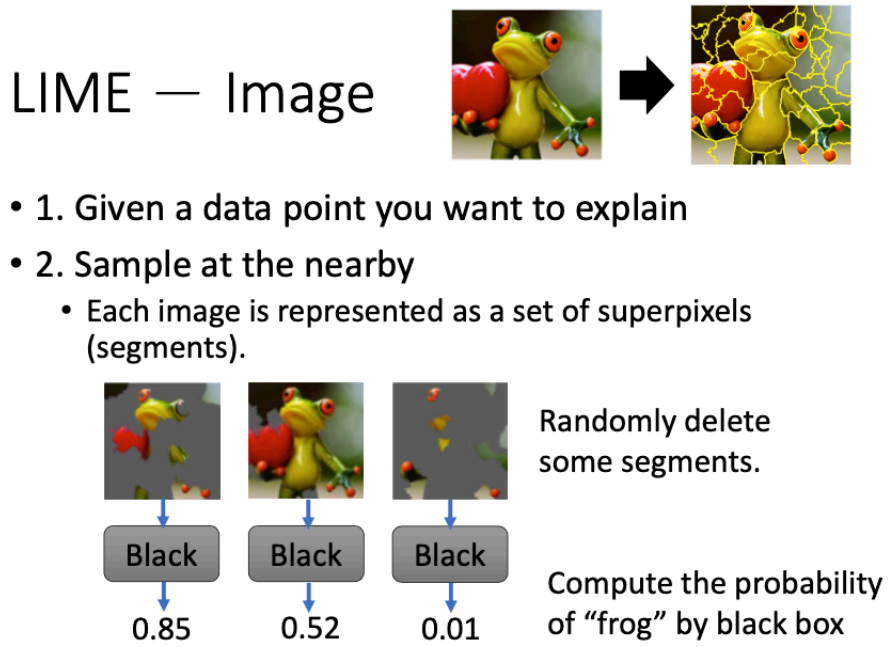


那么到底什么算是nearby呢？用不同的方法进行sample，结果不太一样。对于下图中的region，可以看到离第三个蓝色point的距离很远，取得的效果就非常不好了



LIME-Image

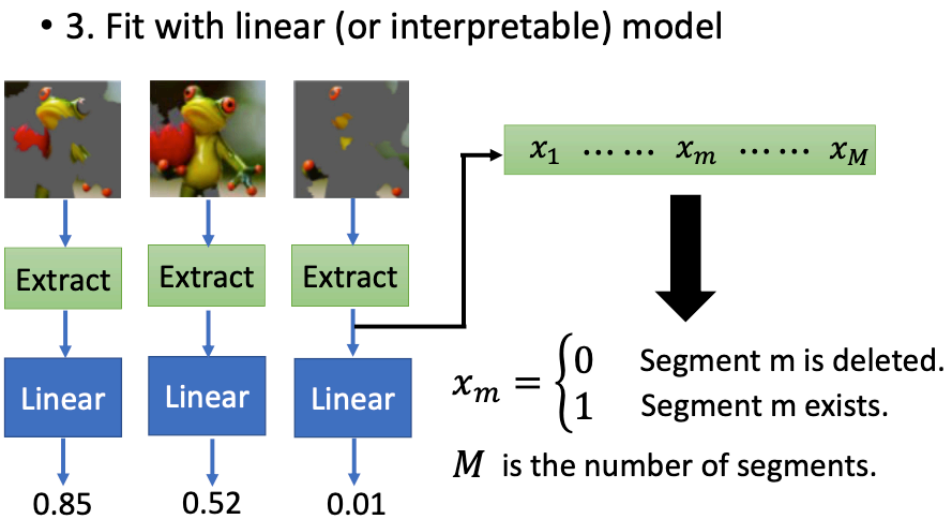
刚才说了general的情况，下面我们讲解LIME应用于image的情况



1. 首先需要一张需要解释的image；为什么这张图片可以被classify为树蛙?
2. sample at the nearby：首先把image分成多个segment，再随机去掉图中的一些segment，就得到了不同的新图片，这些新的图片就是sample的结果；再把这些新生成的图片输入black box，得到新图片是frog的可能性；
3. fit with linear model：即找到一个linear model来fit第3步输出的结果；先extract生成的新图片的特征，再把这些特征输入linear model；

Q：那么如何将image转化为一个vector呢?

A：这里我们将image中的每个segment使用x_i来表示，其中$i = 1, \ldots, m, \ldots, M$，M为segment的数量；$x_i$为1，表示当前segment被deleted，如果为0，表示exist；

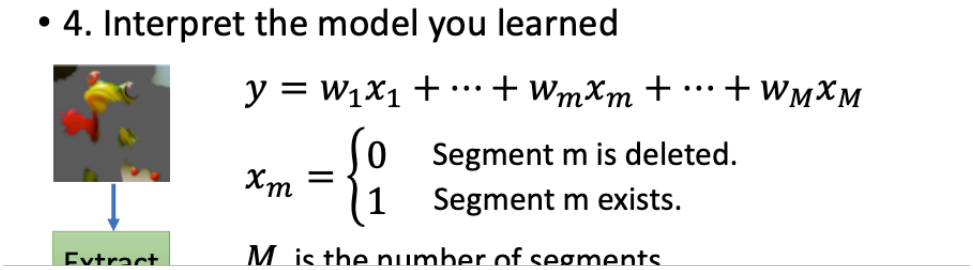


4. Interpret the model：对于学习出来的linear model，我们就可以对其进行interpret；首先需要将x_i和y的关系用一个公式表示出来，即

$$y = w_1 x_1 + .. + w_m x_m + ... + w_M x_M$$

对于w_m的值，有以下三种情况：

- $w_m \approx 0$，segment x_m被认为对分类为frog没有影响；
- $w_m > 0$，x_m对图片分类为frog是有正面的影响的；
- $w_m < 0$，看到这个segment，反而会让机器认为图片不是frog



Decision Tree

如果我们用不限制深度的decision tree，那么我们就可以使用decision tree来模拟black box（neural network），使两者的输出相近

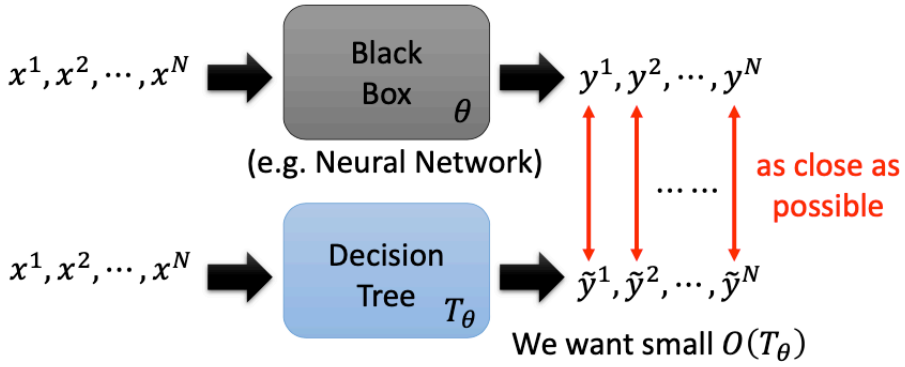
但decision tree的深度不可能是没有限制的。这里我们设neural network的参数为θ，decision tree的参数为T_θ，使用$O(T_\theta)$来表示T_θ的复杂度，复杂度可以用T_θ的深度来表示，也可以用neural的个数来表示；现在我们的目标不仅是使两者输出相近，还需要使$O(T_\theta)$的值最小化

Decision Tree

$O(T_\theta)$: how complex T_θ is

e.g. average depth of T_θ

- Using an interpretable model to mimic the behavior of an uninterpretable model.



Problem: We don't want the tree to be too large.

那么我们如何来实现使$O(T_\theta)$越小越好呢?

如下图所示，我们首先训练一个network，这个network可以很容易地被decision tree解释，使decision tree的复杂度没有那么高；这里我们加入了一个正则项$\lambda O(T_\theta)$，在训练network的同时，不仅要最小化loss function，还需要使$O(T_\theta)$的值尽量小，这时需要找到的network参数为θ^*，

$$\theta^* = arg\min L(\theta) + \lambda O(T_\theta)$$

Decision Tree – Tree regularization

https://arxiv.org/pdf/1711.06178.pdf

- Train a network that is easy to be interpreted by decision tree

T_θ : tree mimicking network with parameters θ

$O(T_\theta)$: how complex T_θ is