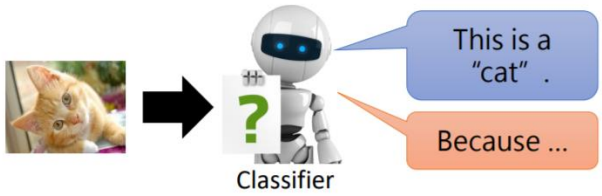
Explainable Machine Learning

Introduction

机器不仅要告诉我们 cat，还要告诉我们为什么



Local Explanation

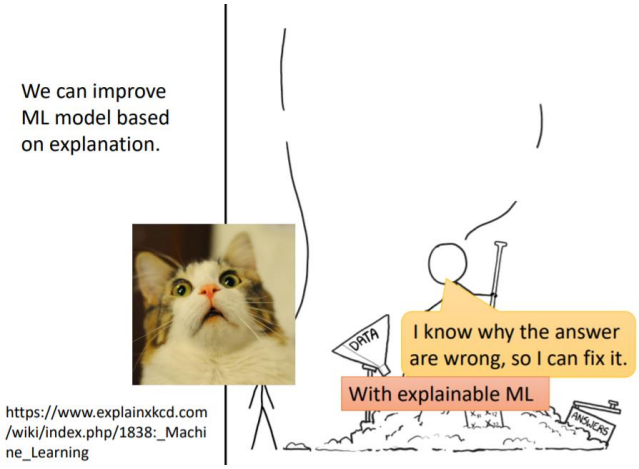
Why do you think *this image* is a cat?

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

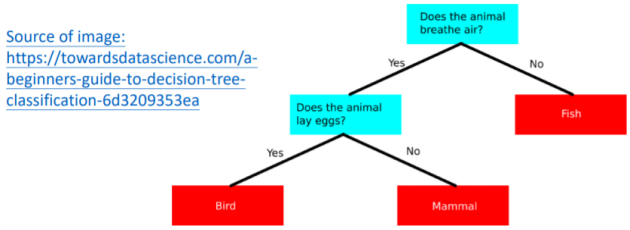
- Some models are intrinsically interpretable.
  - For example, linear model (from weights, you know the importance of features)
  - But ..... not very powerful.
- Deep network is difficult to interpretable.
  - Deep network is a black box.

    Because deep network is a black box, we don't use it.　削足適履 ☹
  - But it is more powerful than linear model ...

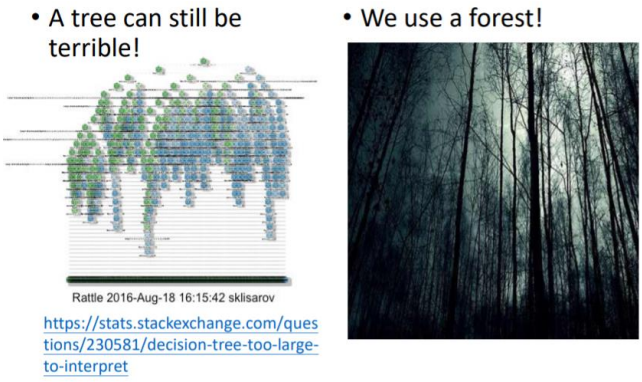
    Let's make deep network interpretable.

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



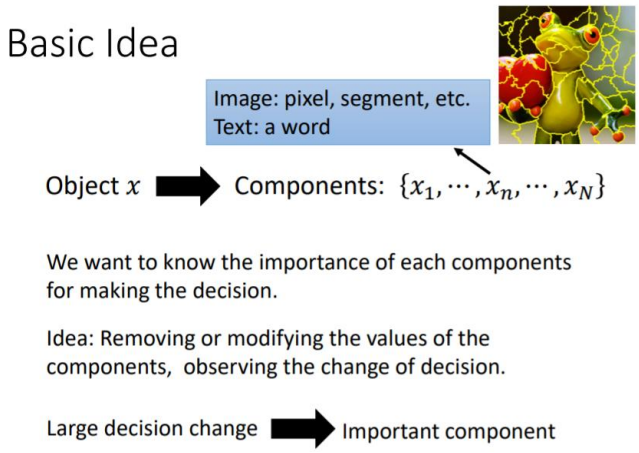
当分支特别多的时候，决策树的表现也会很差



Local Explanation

Basic Idea

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



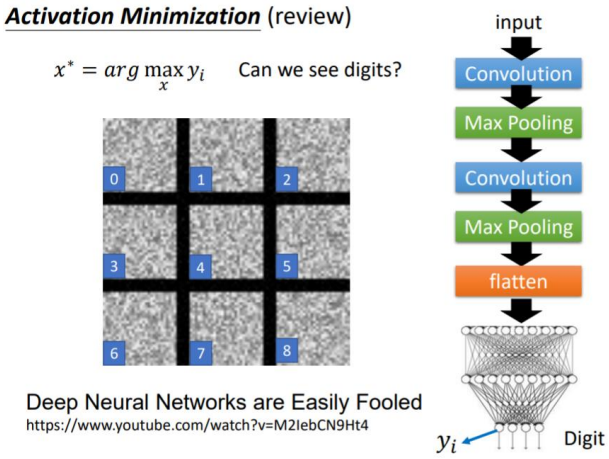
把灰色方块放到图像中，覆盖图像的一小部分；如果我们把灰色方块放到下图中的红色区域，那么对解释的结果影响不大，第一幅图还是一只狗

Global Explanation

Interpret the whole Model

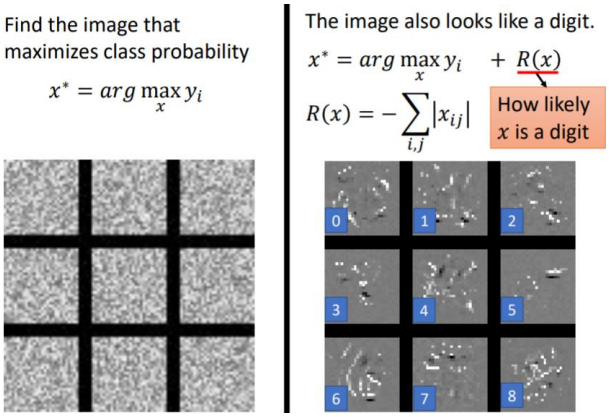
Activation Minimization(review)

让我们先 review 一下 activation minimization，现在我们的目标是找到一个x^*，是的输出的值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，是的输出的y达到最大值；现在我们的目标不仅是找到x是输出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



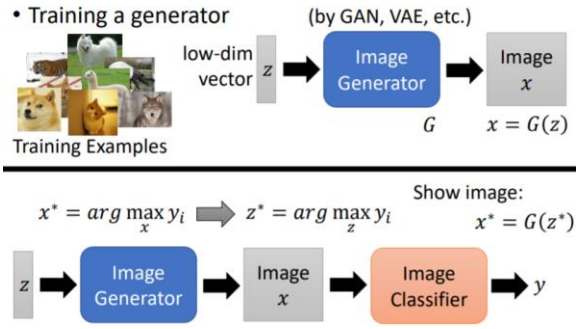
Constraint from Generator

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

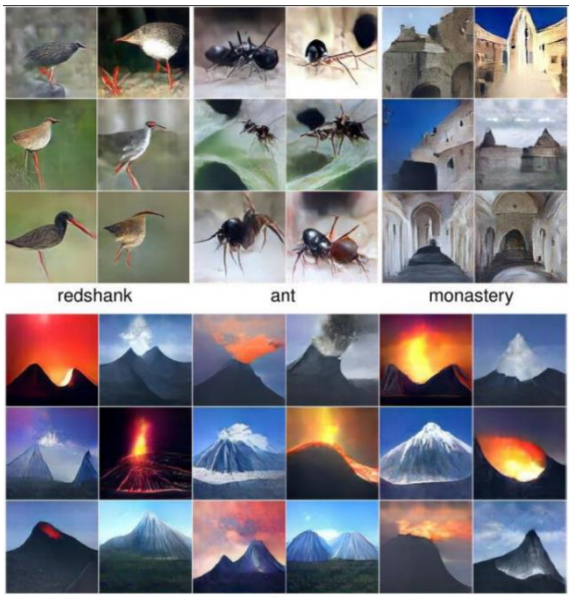
现在我们将生产的 Image 再输入 Image classifier，输出分类结果 y_i；那么我们现在的目标就是找到z^*，使得属于那个类别的可能性 y_i最大

$z^* = argmax\ y_i$

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

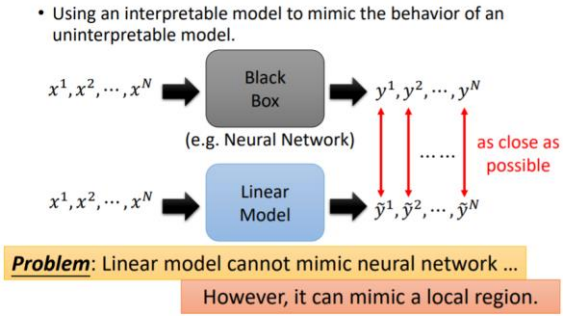


结果展示，现在你问机器蚂蚁长什么样子呢?机器就会给你画一堆蚂蚁的图片出来，再放到 classifier 里面，得出分类结果到底是火山还是蚂蚁



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, x^3 …… x^N,使 linear model 和 Neural Network 有相近输出



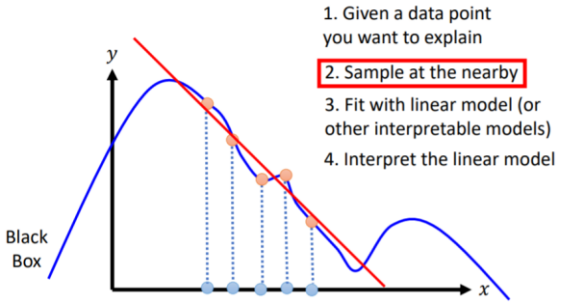
实际上并不能使用 linear model 来模拟整个 Neural network，但可以用来模拟其中一个 local region

Local interpretable Model-Agnostic Explanations (LIME)

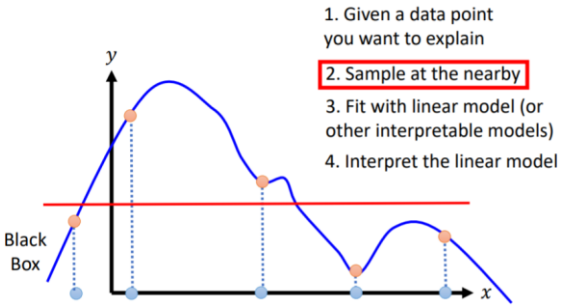
General

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

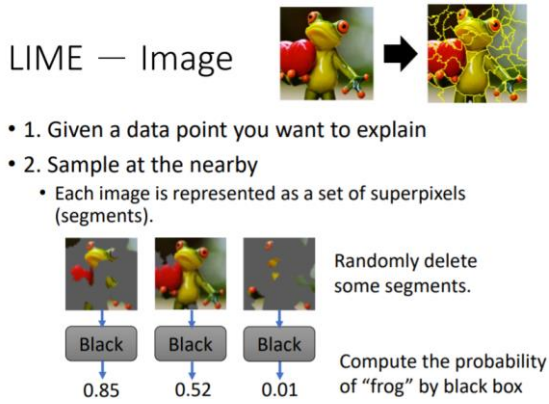


那么到底什么算是 nearby 呢?用不同的方法进行 sample，结果不太一样，对于下图中的 region，可以看到离第三个蓝色 point 的距离很远，取得的效果就非常不好了



LIME-Image

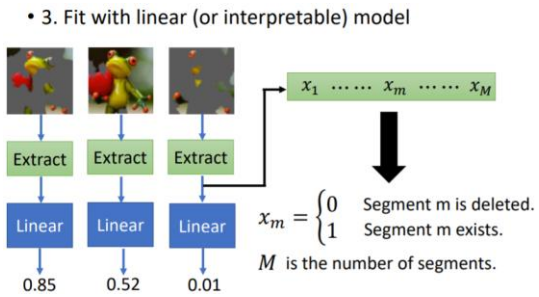
刚才说了 general 的情况，下面我们讲解 LIME 应用于 image 的情况



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$i = 1, \dots \dots, m, \dots, M$,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

$y = w_1 x_1 + \cdots + w_m x_m + \cdots + w_M x_M$

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

• 4. Interpret the model you learned

$$y = w_1 x_1 + \cdots + w_m x_m + \cdots + w_M x_M$$

$$x_m = \begin{cases} 0 & \text{Segment m is deleted.} \\ 1 & \text{Segment m exists.} \end{cases}$$

M is the number of segments.

If $w_m \approx 0$ ➡ segment m is not related to "frog"

If w_m is positive ➡ segment m indicates the image is "frog"

If w_m is negative ➡ segment m indicates the image is not "frog"

Decision Tree

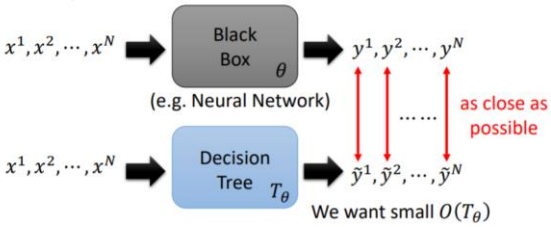
如果我们用不限制深度的 decision tree，那么我们就可以使用 decision tree 来模拟 black box(neural network),使两者的输出相近

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，还需要是$O(T_\theta)$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^*，

$\theta^* = argminL(\theta) + \lambda O(T_\theta)$

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

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

Original loss function for training network

Preference for network parameters

➡ Tree Regularization

Is the objective function with tree regularization differentiable? No! Check the reference for solution.