### FCN-图像语义分割

### 图像语义分割，就是对一张图片上的所有像素点进行分类。以往的CNN都是对整张图片进行分类,不能识别图片中特定部分的物体，而全卷积网络是对一张图片中的每个像素进行分类，以此达到对图片特定部分进行分类的效果

与传统CNN在卷积层之后使用全连接层得到固定长度的特征向量进行分类（全连接层＋softmax）不同，FCN可以接受任意尺寸的输入图像，然后通过反卷积层对最后一个卷积层的feature map进行上采样, 使它恢复到输入图像相同的尺寸，从而可以对每个像素都产生了一个预测, 同时保留了原始输入图像中的空间信息, 最后在与输入图等大小的特征图上对每个像素进行分类，逐像素地用softmax分类计算损失,相当于每个像素对应一个训练样本。

因为语义分割需要输出整张图片的分割图，所以要求网络中的特征图至少是二维的，这样才能通过上采样还原到输入图片的同等大小。   
这就需要替换掉全连接层，改换为卷积层，而这就是卷积化。

**全连接层和卷积层的转化**   
全连接层和卷积层唯一的不同就是卷积层中的神经元只与输入数据中的一个局部区域连接，并且在卷积列中的神经元共享参数。   
但是这两类层的神经元都是计算点积，它们的函数形式是一样的。因此，将此两者相互转化是可能的。

对于任意一个卷积层，都存在一个能实现和它一样的前向传播函数的全连接层。权重矩阵是一个巨大的矩阵，除了某些特定块，其余部分都是零。而在其中大部分块中，元素都是相等的。

任何全连接层都可以被转化为卷积层。   
假设一个 numout=4096 的全连接层，输入数据体的尺寸是 7∗7∗512，这个全连接层可以被等效地看做一个 Kernel=7,P=0,S=1,numout=4096 的卷积层。   
也就是将滤波器的尺寸设置为和输入数据体的尺寸一致了，这个结果就和使用初始的那个全连接层一样了。

全连接层转化为卷积层：在两种变换中，将全连接层转化为卷积层在实际运用中更加有用。

假设卷积神经网络的输入是224x224x3的图像，一系列的卷积层和下采样层将图像数据变为尺寸为 7x7x512 的数据体。AlexNet使用了两个尺寸为4096的全连接层，最后一个有1000个神经元的全连接层用于计算分类评分。   
我们可以将这3个全连接层中的任意一个转化为卷积层：

第一个连接区域是[7x7x512]的全连接层，令其滤波器尺寸为Kernel=7，这样输出数据体就为[1x1x4096]了；   
第二个全连接层，令其滤波器尺寸为Kernel=1，这样输出数据体为[1x1x4096]；   
最后一个全连接层也做类似的，令其Kernel=1，最终输出为[1x1x1000]。

转化的意义   
这样的变换每次都需要把全连接层的权重WW重塑成卷积层的滤波器。   
如果想让卷积网络在一张更大的输入图片上滑动，得到多个输出，那么卷积层可以在单次前向传播中完成全连接层几次才能完成的操作。

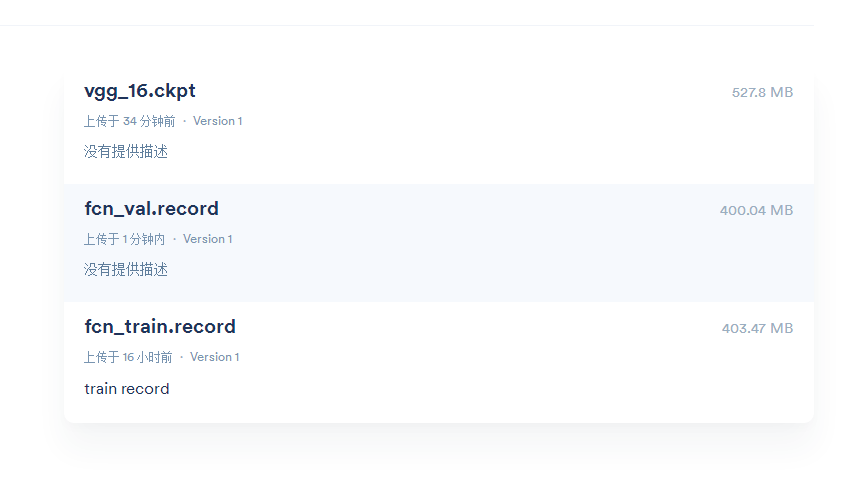
假设想让224×224的滑窗，以32的步长在384×384的图片上滑动，将每个位置的特征都提取到网络中，最后得到6×6个位置的类别得分。

如果224×224的输入图片经过卷积层和下采样层之后得到了[7x7x512]的特征图，那么，384×384的大图片直接经过同样的卷积层和下采样层之后会得到[12x12x512]的特征。然后再经过上面由3个全连接层转化得到的3个卷积层，最终得到[6x6x1000]的输出((12 – 7)/1 + 1 = 6)。这个结果正是滑窗在原图中6×6个位置的得分。

对于384×384的图像，让（含全连接层）的初始卷积神经网络以32像素的步长独立对图像中的224×224块进行多次评价，其效果和使用把全连接层变换为卷积层后的卷积神经网络进行一次前向传播是一样的。   
所以将全连接层转换成卷积层会更简便。

**结果：**

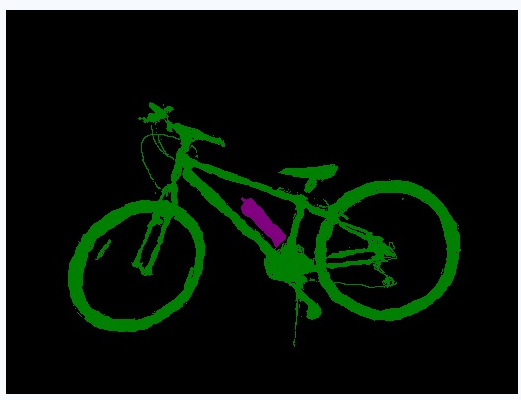
将代码上传到Tinymind上之后



Fcn\_16s

原图

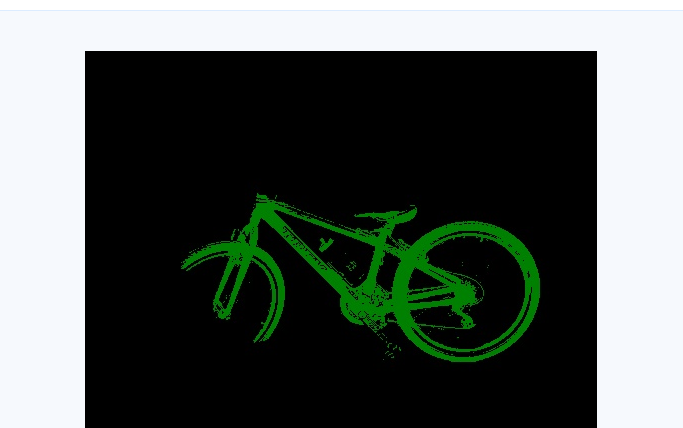
：

标签：

预测



Crf:



FCN\_8s代码实现：

fcn实现主要是全卷积网络来实现的，fcn32是在p5的时候将vgg16网络的全连接替换成卷积网络就成了fcn32，然后在此基础上 将p4进行2倍的反卷积之后与logits进行相加之后再用16倍的反卷积将图片恢复成原图大小 这就是fcn16s！同道理，在p4进行双倍的反卷积并与logits进行相加之后得到upsampled\_logits，再对p3进行与p4相同操作再与upsampled\_logits进行相加 然后再用8倍的反卷积将图像恢复成原图 这就是fcn8s了



