**Fast-RCNN**

论文：Fast R-CNN

Fast-RCNN是作者Ross继R-CNN后又一力作。同样使用vgg16作为网络的backbone，与R-CNN相比，训练时间快9被，测试推理时间快213倍，准确率从62提升到66(在Pascal Voc数据集上)

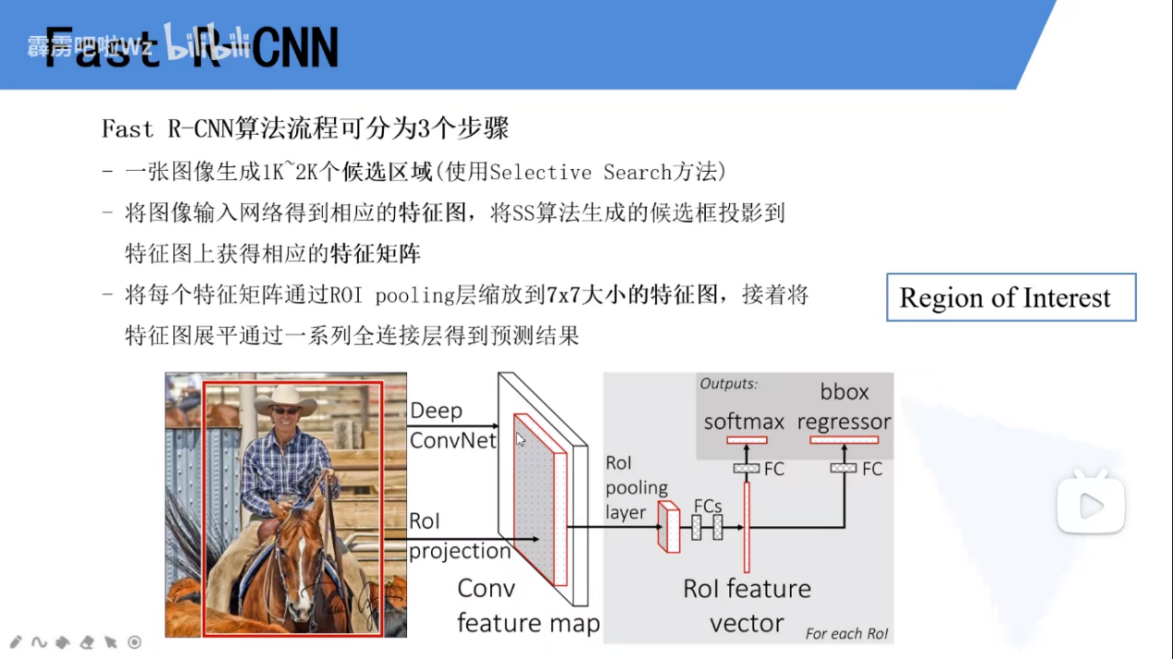
**Fast R-CNN步骤:**

1. 使用SS算法生成1k~2k个候选区域

2. 将原图像直接输入网络得到特征图，将SS算法生成的候选框(红色的框)投影到特征图上获得相应的特征矩阵(第二步与RCNN的第二部已经完全不同了)

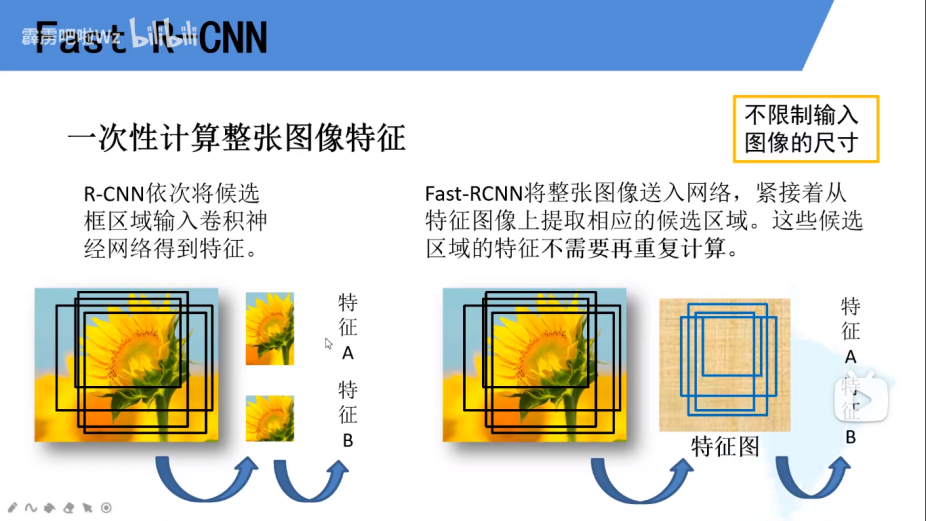
3. 将每个特征矩阵通过ROI pooling层缩放到7\*7大小的特征图，接着将特征图展平通过一系列全连接层得到预测结果

Fast R-CNN并不需要训练SVM分类器和bounding box回归器。



RCNN通过SS得到2000个候选框，需要2000次正向传播来提取特征，很多图像重叠的部分会产生冗余，产生相似的特征向量。

Fast R-CNN将整张图像送入网络中，从特征图上提取相应的候选区域(怎么提取，作者参考的别的论文)，这些候选框区域的特征不需要重复计算。或者可以说根据候选区域，得到相应候选区域的特征矩阵(反正二者存在映射关系)

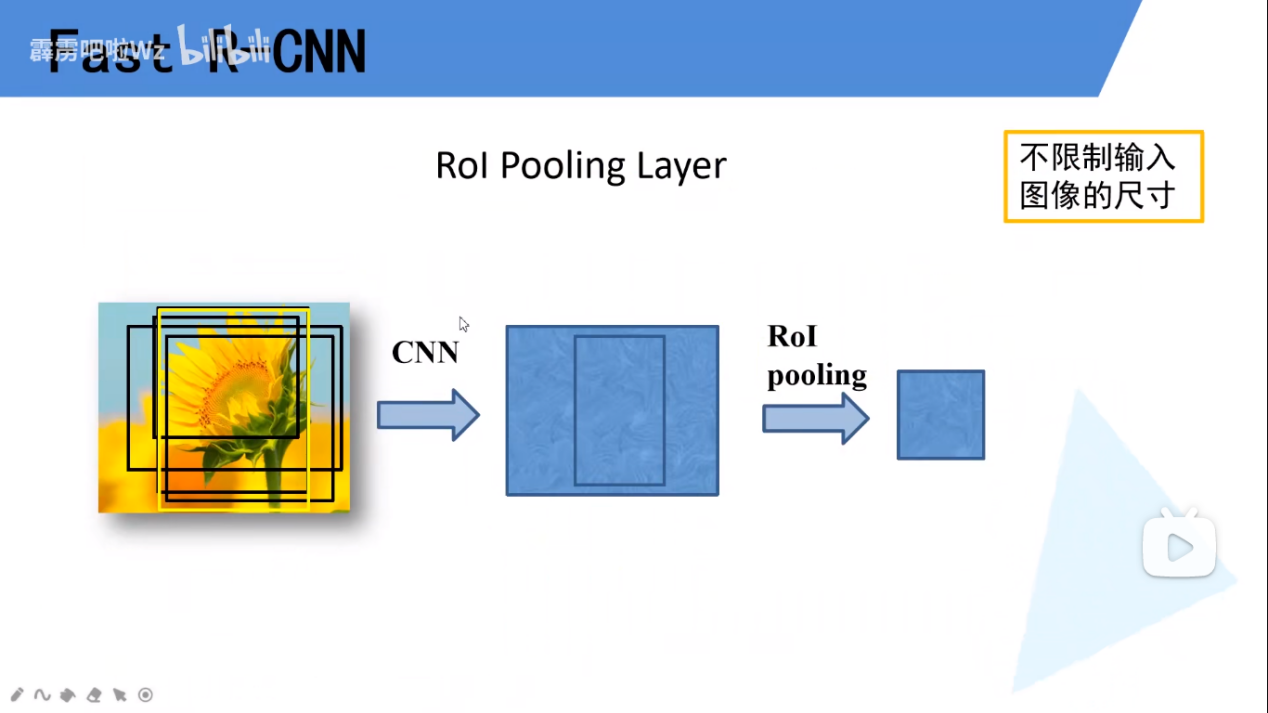


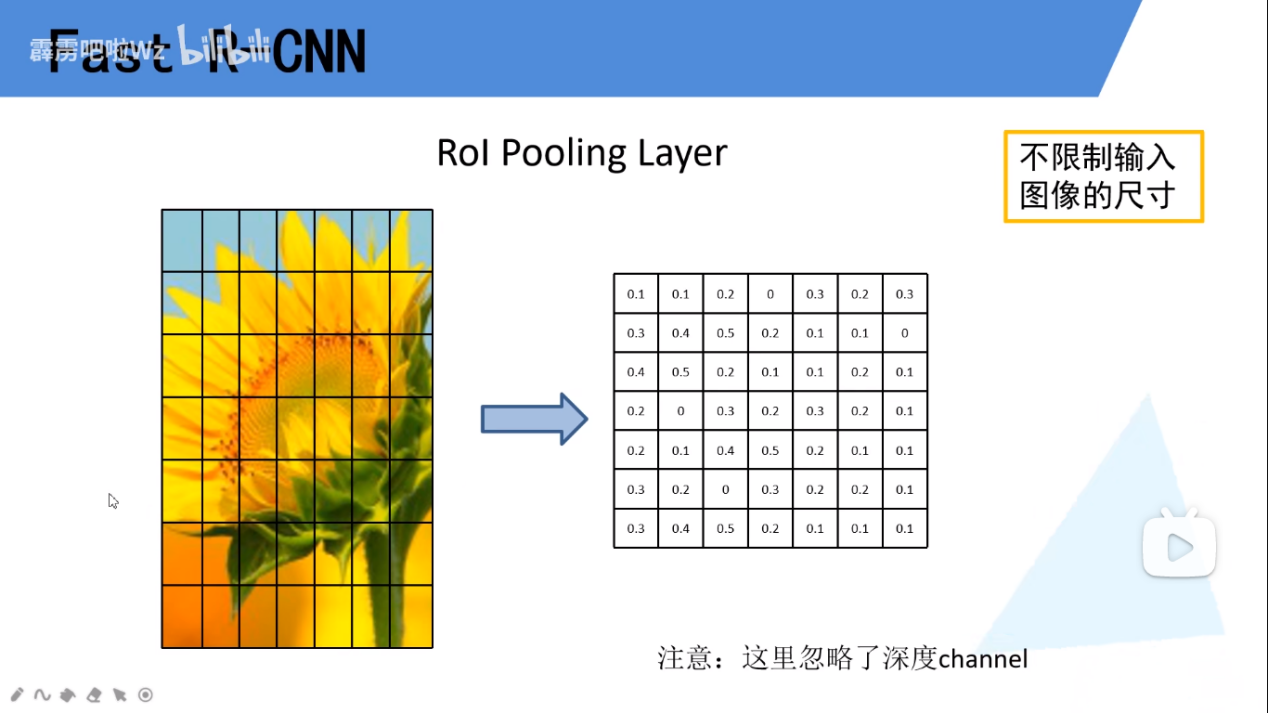
SS算法会产生2000个候选框，在训练过程中，只是抽取其中一小部分，这一部分当中还分正样本和负样本。正样本就是候选框中真实存在的物体，负样本可以理解为背景。

**正样本负样本：**

比如训练一个分类器，猫狗。如果猫的图片很多，狗的很少，则分类器分类时会倾向于猫。正负样本同理。如果正样本很多，则网络以很大的概率认为候选区域是我们检测的目标，可能框起来的是一个背景，它也认为这是我们需要检测的目标

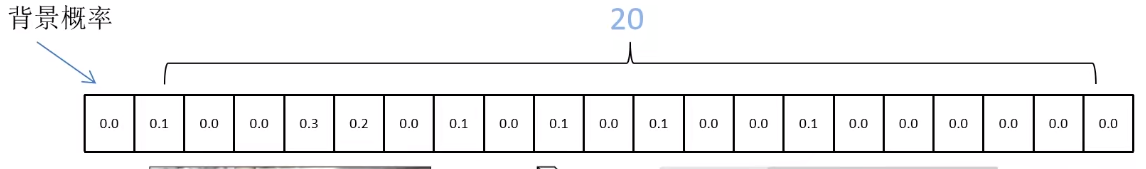
特征图经过ROI池化成特定的尺寸

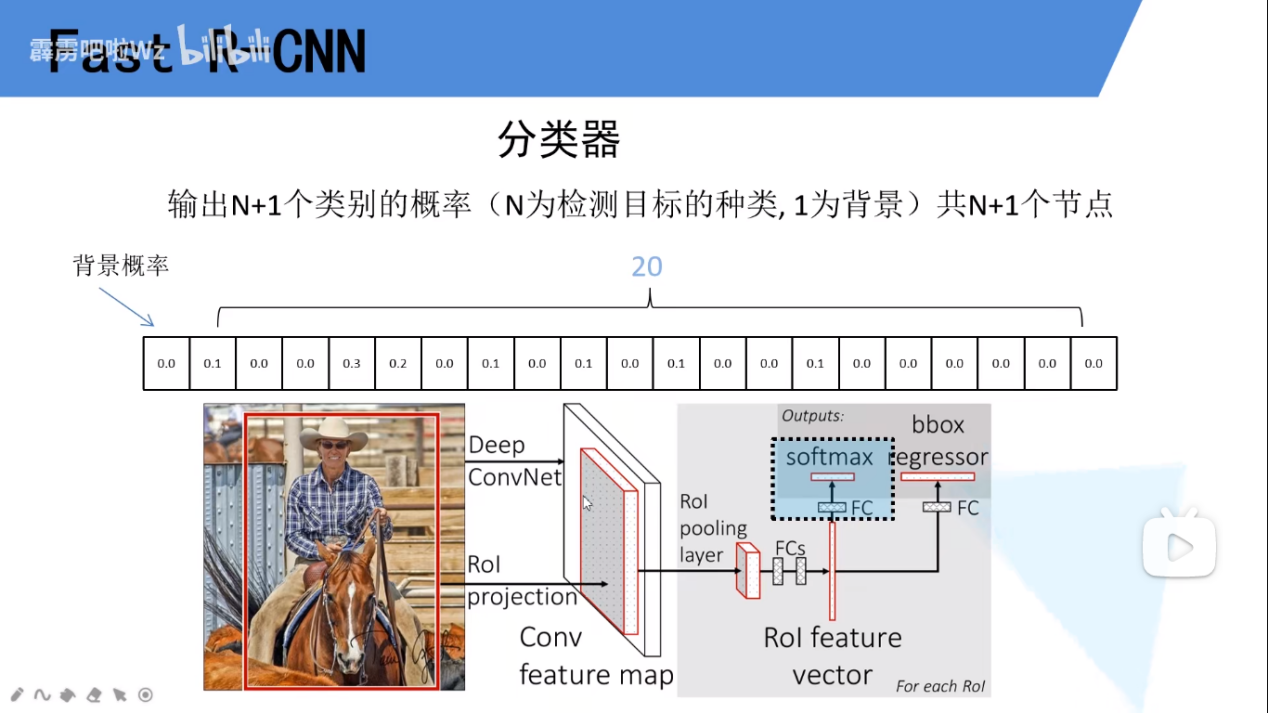




把整张图片放入cnn网络可以得到特征图，根据原图和特征图的映射关系，可以得到候选区域的特征矩阵，通过ROI pooling统一resize成统一尺寸，之后在经过两个全连接层，得到ROI特征向量。再并联两个全连接层，其中第一个全连接层用于目标概率的预测，另一个是bounding box regression(边界框回归)预测

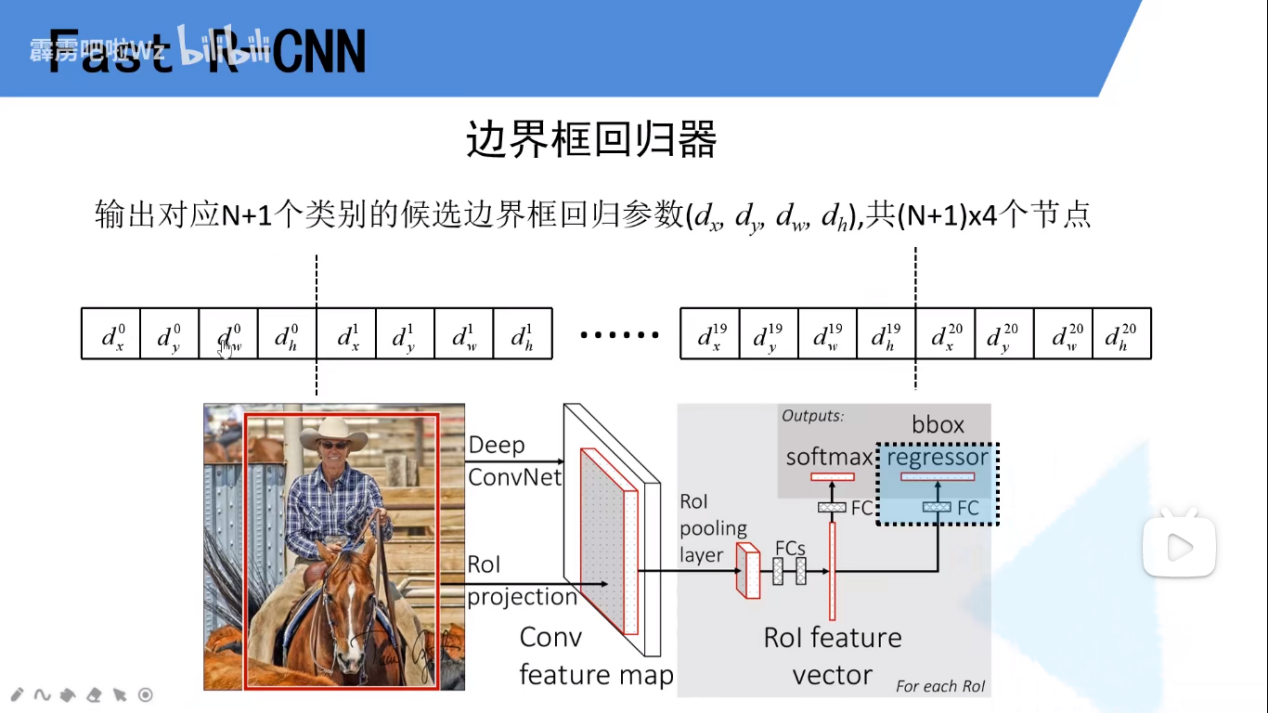
其中用来目标概率预测的全连接层，会输出n+1个类别概率，n是n个类别的目标，1是背景。以poscal voc数据集为例，有20个类别，所以会输出21个概率，第一个概率是背景的概率。下图概率框是经过softmax的，概率和为1





第二个全连接层是边界框回归器，会输出n+1个类别的候选框的回归参数(dx,dy,dw,dh)(d代表预测的偏移量)，共(n+1)\*4个节点。每4个一组



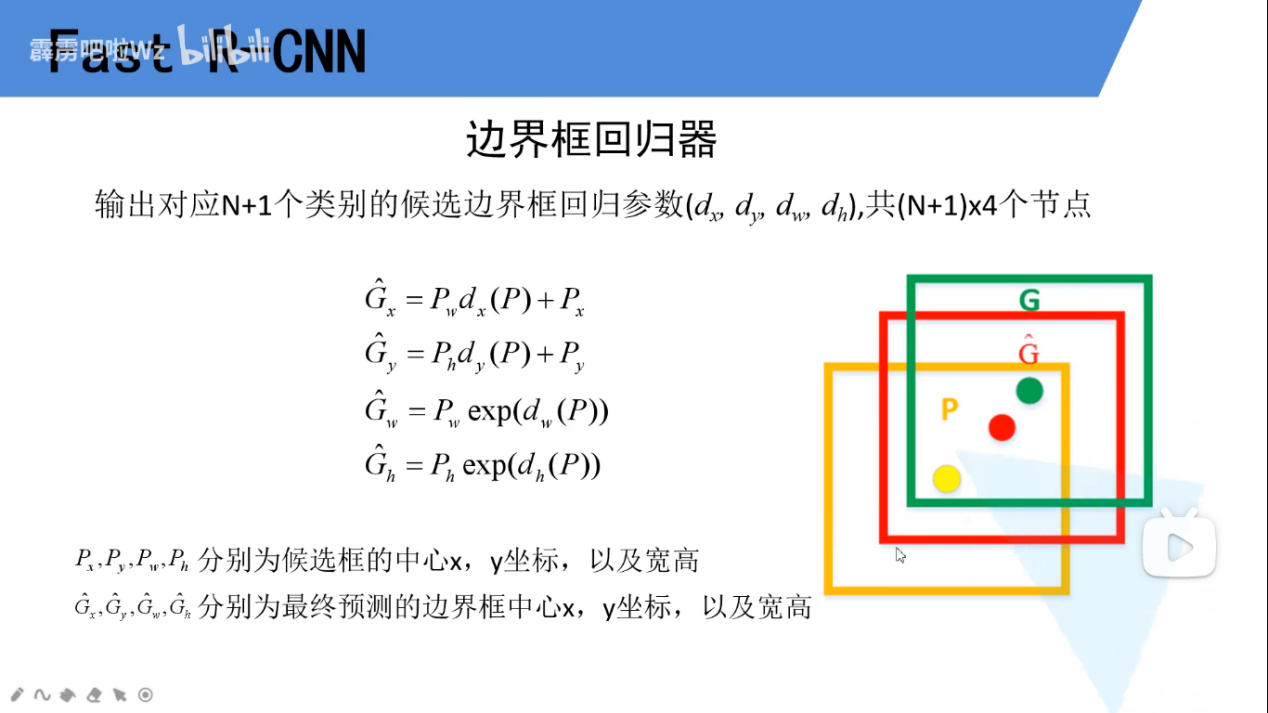


Px,Py,Pw,Ph：分别是黄色框中心点的x,y坐标，以及宽高。

同理G尖。G是ground truth，P应该是SS框出来的region proposal也叫候选框，红色的框是通过边界框回归器生成的。

这个红色的框vx=(Gx-Px)/Pw。同理vy。vw = ln(Gw/Pw)

G尖x的计算公式 = 候选框的中心坐标 \* 回归参数dx + 候选框中心坐标



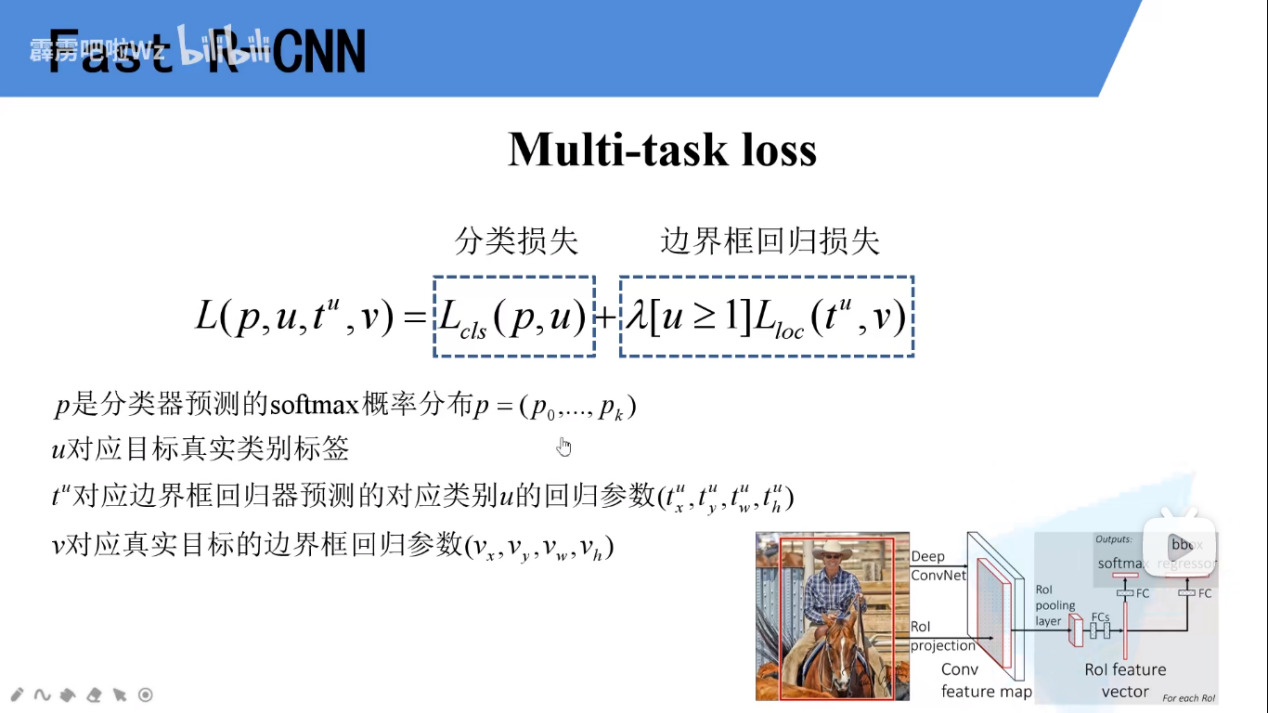
Fast R-CNN的损失函数

P：经过softmax后各个类别的概率分布(比如P0是为背景的概率)

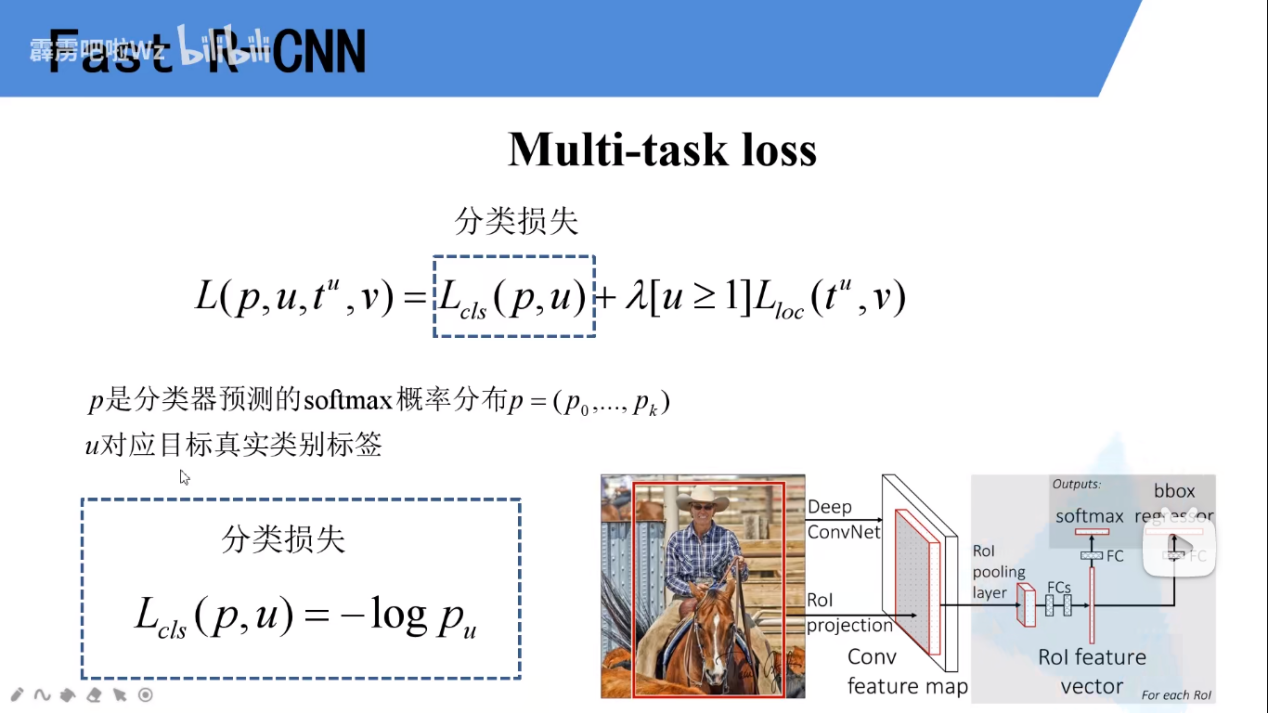
u：真实类别标签

v：真实目标边框的回归参数(vx,vy,vw,vh)是那个红色的框????(和G尖重了)

t^u：边界框回归器预测对应类别u的回归参数(.......)???????不懂

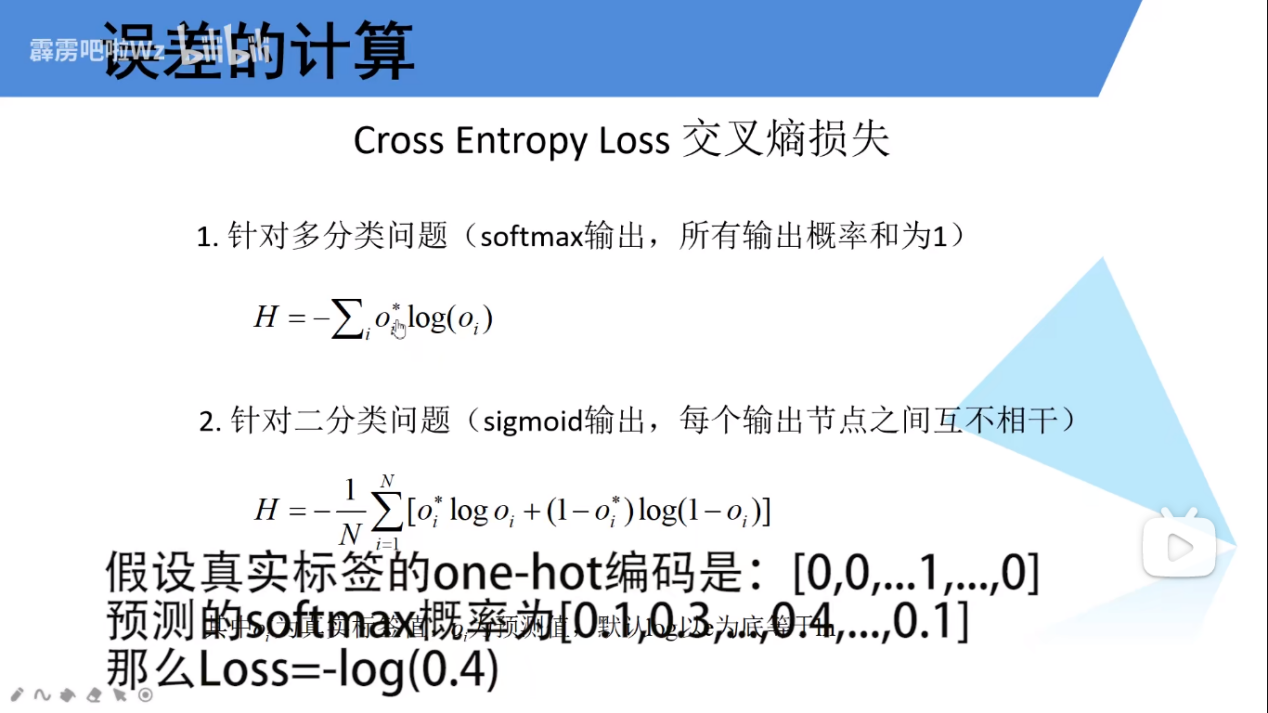


这个分类损失和CrossEntropyLoss损失函数一样



真实标签是转成one-hot编码的。

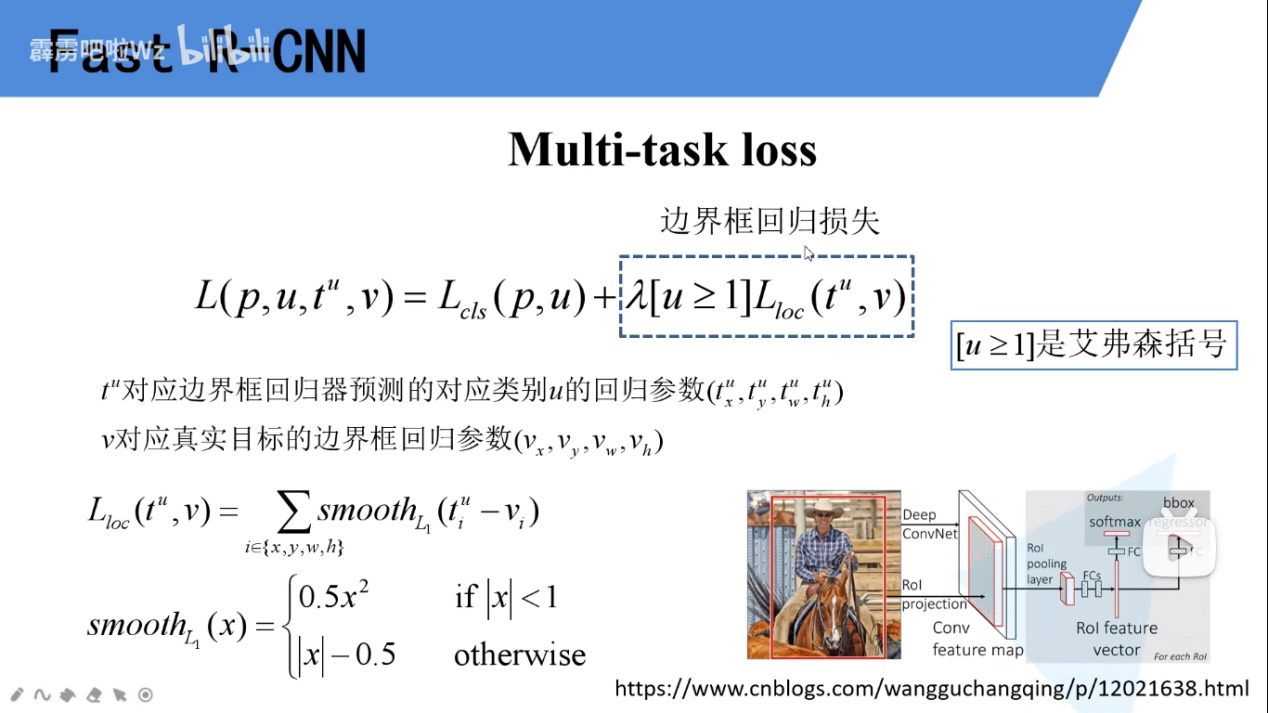
H = -(0\*log(0.1) +...+ 1\*log(0.4) + 0\*log(0.1)) = -log(0.4)



smooth是t∈(x,y,w,h)4部分累加而来，smooth函数在下面

λ是平衡系数，用来平衡分类损失和边界框回归损失

[u>=1]艾弗森括号：当u>=1时，这项为1，当u<0时，这项为0。这个u代表目标的真实标签。候选区域是我们要检测的某一类别时，u>=1(框起来的东西在分类的类别中)，也就是对应的正样本。当候选区域是背景时，u<0。既然是背景，就没有边界框回归损失这一项



SS算法在cpu上要2s，但是特征提取、分类、边界框回归只要零点几秒。所以SS算法比较费时

