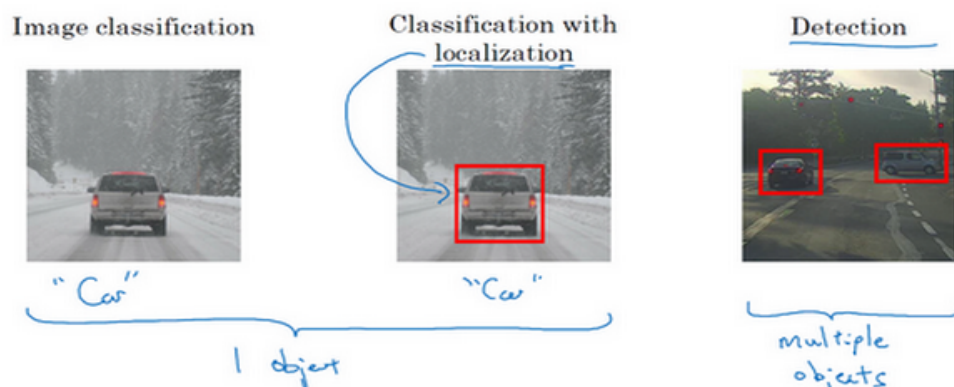


3.1 目标定位 (Object localization)

What are localization and detection?

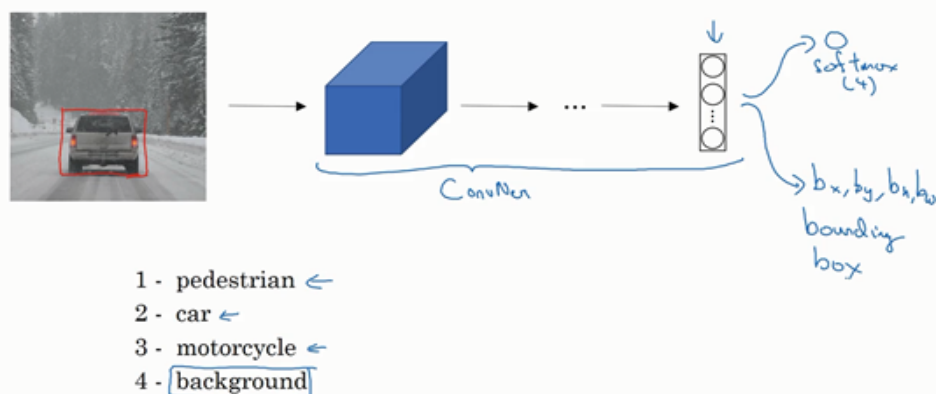


图片分类任务我们已经熟悉了，就是算法遍历图片，判断其中的对象是不是汽车，这就是图片分类。这节课我们要学习构建神经网络的另一个问题，即**定位分类**问题。这意味着，**我们不仅要用算法判断图片中是不是一辆汽车，还要在图片中标记出它的位置，用边框或红色方框把汽车圈起来，这就是定位分类问题。**其中“定位”的意思是判断汽车在图片中的具体位置。这周后面几天，我们再讲讲当图片中有多个对象时，应该如何检测它们，并确定出位置。比如，你正在做一个自动驾驶程序，程序不但要检测其它车辆，还要检测其它对象，如行人、摩托车等等，稍后我们再详细讲。

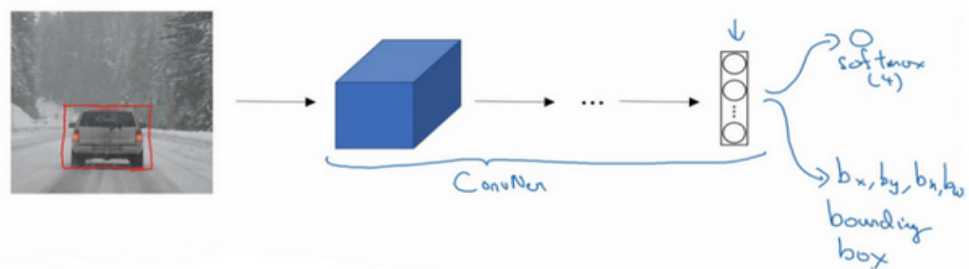
本周我们要研究的分类定位问题，通常只有一个较大的对象位于图片中间位置，我们要对它进行识别和定位。而在对象检测问题中，图片可以含有多个对象，甚至单张图片中会有多个不同分类的对象。因此，图片分类的思路可以帮助学习分类定位，而对象定位的思路又有助于学习对象检测，我们先从分类和定位开始讲起。

图片分类问题你已经并不陌生了，例如，输入一张图片到多层卷积神经网络。这就是卷积神经网络，它会输出一个特征向量，并反馈给 **softmax** 单元来预测图片类型。

Classification with localization



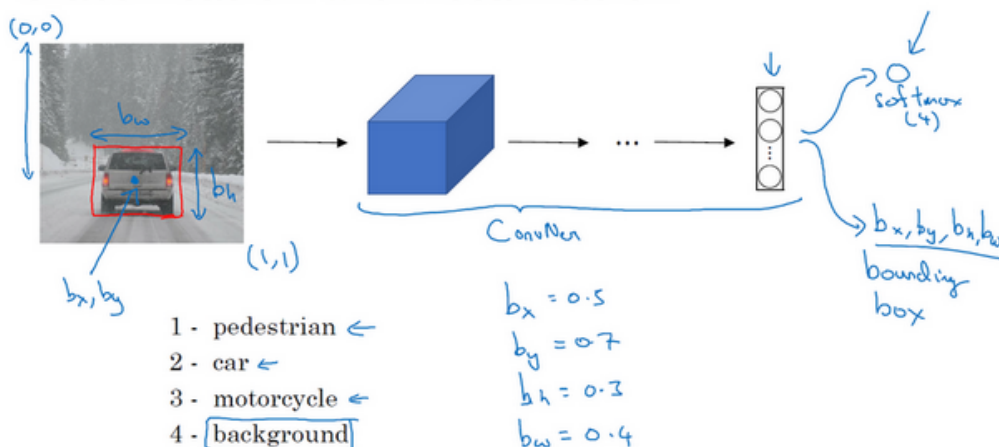
如果你正在构建汽车自动驾驶系统，那么对象可能包括以下几类：行人、汽车、摩托车和背景，这意味着图片中不含有前三种对象，也就是说图片中没有行人、汽车和摩托车，输出结果会是背景对象，这四个分类就是 softmax 函数可能输出的结果。



这就是标准的分类过程，如果你还想定位图片中汽车的位置，该怎么做呢？我们可以让神经网络多输出几个单元，**输出一个边界框**。具体说就是让神经网络再多输出 4 个数字，标记为 b_x, b_y, b_h 和 b_w ，这四个数字是被检测对象的边界框的参数化表示。

我们先来约定本周课程将使用的符号表示，**图片左上角的坐标为(0,0)，右下角标记为(1,1)**。要确定边界框的具体位置，需要指定红色方框的中心点，这个点表示为 (b_x, b_y) ，边界框的高度为 b_h ，宽度为 b_w 。因此训练集不仅包含神经网络要预测的对象分类标签，还要包含表示边界框的这四个数字，接着采用监督学习算法，输出一个分类标签，还有四个参数值，从而给出检测对象的边框位置。此例中， b_x 的理想值是 0.5，因为它表示汽车位于图片水平方向的中间位置； b_y 大约是 0.7，表示汽车位于距离图片底部 $\frac{3}{10}$ 的位置； b_h 约为 0.3，因为红色方框的高度是图片高度的 0.3 倍； b_w 约为 0.4，红色方框的宽度是图片宽度的 0.4 倍。

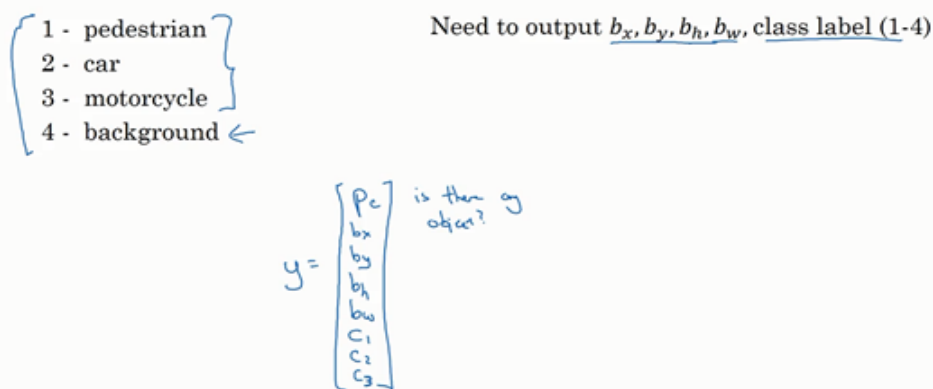
Classification with localization



Andrew Ng

下面我再具体讲讲如何为监督学习任务定义目标标签 y 。

Defining the target label y



请注意，这里有四个分类，神经网络输出的是这四个数字和一个分类标签，或分类标签出

现的概率。目标标签 y 的定义如下：

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

它是一个向量，第一个组件 p_c 表示是否含有对象，如果对象属于前三类（行人、汽车、摩托车），则 $p_c = 1$ ，如果是背景，则图片中没有要检测的对象，则 $p_c = 0$ 。我们可以这样理解 p_c ，它表示被检测对象属于某一分类的概率，背景分类除外。


如果检测到对象，就输出被检测对象的边界框参数 b_x 、 b_y 、 b_h 和 b_w 。最后，如果存在某个对象，那么 $p_c = 1$ ，同时输出 c_1 、 c_2 和 c_3 ，表示该对象属于 1-3 类中的哪一类，是行人，汽车还是摩托车。鉴于我们所要处理的问题，我们假设图片中只含有一个对象，所以针对这个

分类定位问题，图片最多只会出现其中一个对象。

Defining the target label y

Need to output b_x, b_y, b_h, b_w , class label (1-4)

1 - pedestrian
2 - car
3 - motorcycle
4 - background

$x =$ 

$y =$ $\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$ is there an object?



$\begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$

我们再看几个样本，假如这是一张训练集图片，标记为 x ，即上图的汽车图片。而在 y 当中，第一个元素 $p_c = 1$ ，因为图中有一辆车， b_x 、 b_y 、 b_h 和 b_w 会指明边界框的位置，所以标签训练集需要标签的边界框。图片中是一辆车，所以结果属于分类2，因为定位目标不是行人或摩托车，而是汽车，所以 $c_1 = 0$ ， $c_2 = 1$ ， $c_3 = 0$ ， c_1 、 c_2 和 c_3 中最多只有一个等于1。

这是图片中只有一个检测对象的情况，如果图片中没有检测对象呢？如果训练样本是这样一张图片呢？

Need to output b_x, b_y, b_h, b_w , class label (1-4)

1 - pedestrian
2 - car
3 - motorcycle
4 - background

$x =$  

$y =$ $\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$ is there an object?

$\begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$ $\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix} \leftarrow \text{"don't care"}$

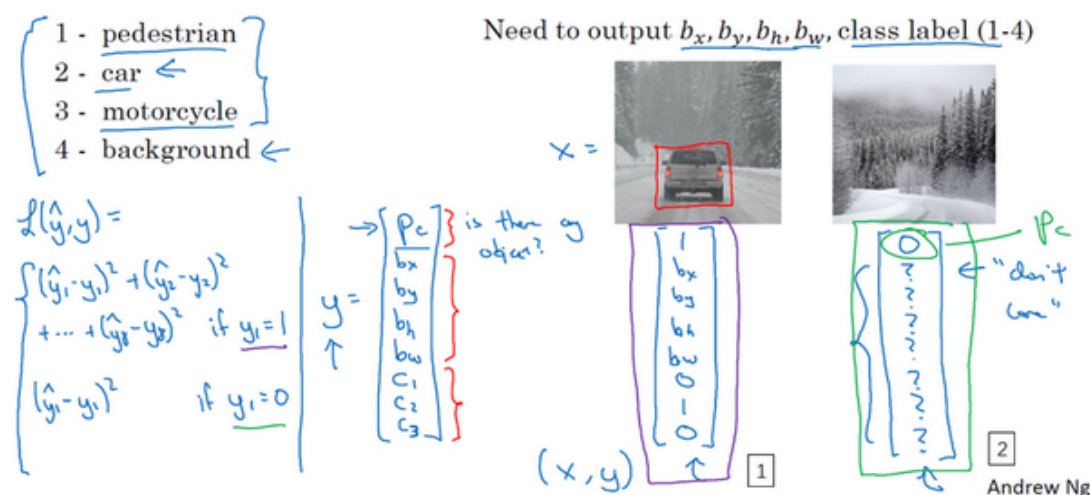
(x, y)

Andrew Ng

这种情况下， $p_c = 0$ ， y 的其它参数将变得毫无意义，这里我全部写成问号，表示“毫无意义”的参数，因为图片中不存在检测对象，所以不用考虑网络输出中边界框的大小，也不用考虑图片中的对象是属于 c_1 、 c_2 和 c_3 中的哪一类。针对给定的被标记的训练样本，不论图片中是否含有定位对象，构建输入图片 x 和分类标签 y 的具体过程都是如此。这些数据最终定义了训练集。

最后，我们介绍一下神经网络的损失函数，其参数为类别 y 和网络输出 \hat{y} ，如果采用平方误差策略，则 $L(\hat{y}, y) = (\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \dots + (\hat{y}_8 - y_8)^2$ ，损失值等于每个元素相应差值的平方和。

Defining the target label y



如果图片中存在定位对象，那么 $y_1 = 1$ ，所以 $y_1 = p_c$ ，同样地，如果图片中存在定位对象， $p_c = 1$ ，损失值就是不同元素的平方和。

另一种情况是， $y_1 = 0$ ，也就是 $p_c = 0$ ，损失值是 $(\hat{y}_1 - y_1)^2$ ，因为对于这种情况，我们不用考虑其它元素，只需要关注神经网络输出 p_c 的准确度。

回顾一下，当 $y_1 = 1$ 时，也就是这种情况（编号 1），平方误差策略可以减少这 8 个元素预测值和实际输出结果之间差值的平方。如果 $y_1 = 0$ ， y 矩阵中的后 7 个元素都不用考虑（编号 2），只需要考虑神经网络评估 y_1 （即 p_c ）的准确度。

为了让大家了解对象定位的细节，这里我用平方误差简化了描述过程。实际应用中，你可以不对 c_1 、 c_2 、 c_3 和 **softmax** 激活函数应用对数损失函数，并输出其中一个元素值，通常做法是对边界框坐标应用平方差或类似方法，对 p_c 应用逻辑回归函数，甚至采用平方预测误差也是可以的。

以上就是利用神经网络解决对象分类和定位问题的详细过程，结果证明，利用神经网络输出批量实数来识别图片中的对象是个非常有用的算法。下节课，我想和大家分享另一种思路，就是把神经网络输出的实数集作为一个回归任务，这个思想也被应用于计算机视觉的其它领域，也是非常有效的，所以下节课见。