



deeplearning.ai

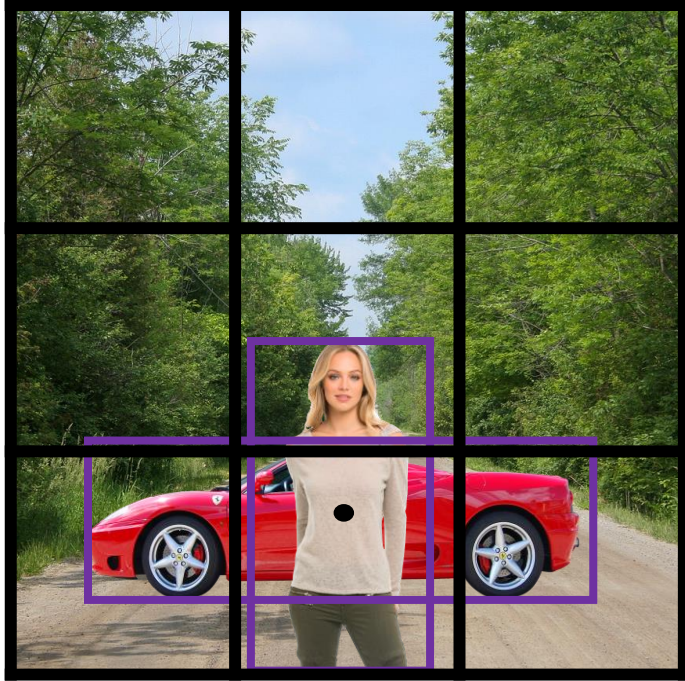
Object Detection

Anchor boxes

What if a grid cell wants to detect multiple objects

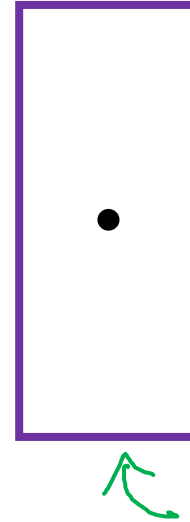
Overlapping objects:

In general you might use more anchor boxes maybe five or even more.

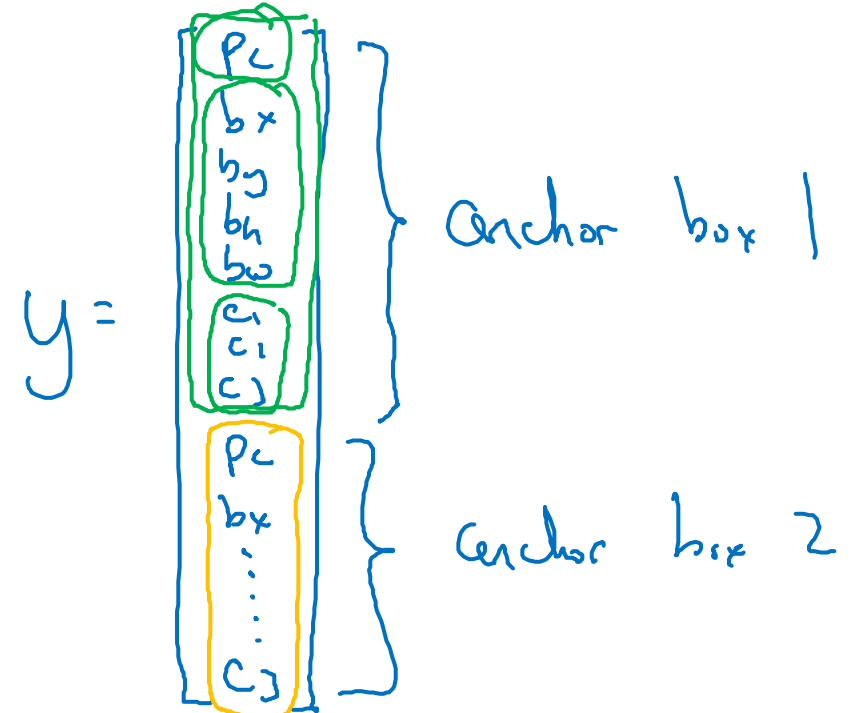
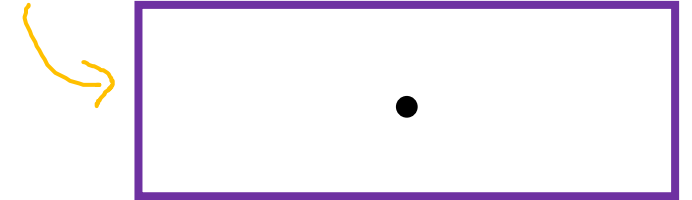


$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Anchor box 1:



Anchor box 2:

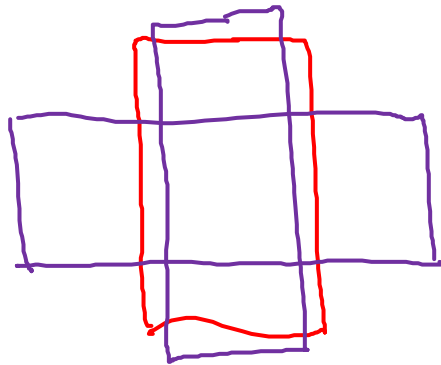


Anchor box algorithm

Previously:

Each object in training set image is assigned to grid cell that contains that object's midpoint.

Output y :
 $3 \times 3 \times 8$



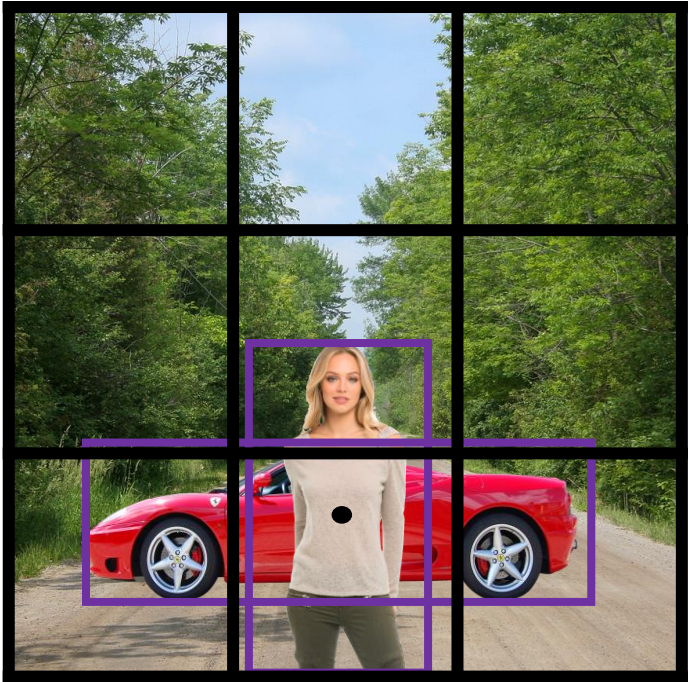
With two anchor boxes:

Each object in training image is assigned to grid cell that contains object's midpoint and anchor box for the grid cell with highest IoU.

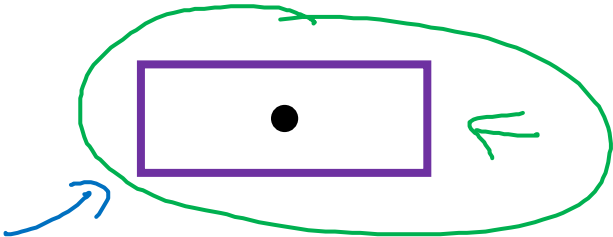
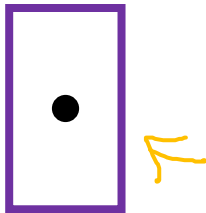
(grid cell, anchor box)

Output y :
 $3 \times 3 \times 16$
 $3 \times 3 \times 2 \times 8$

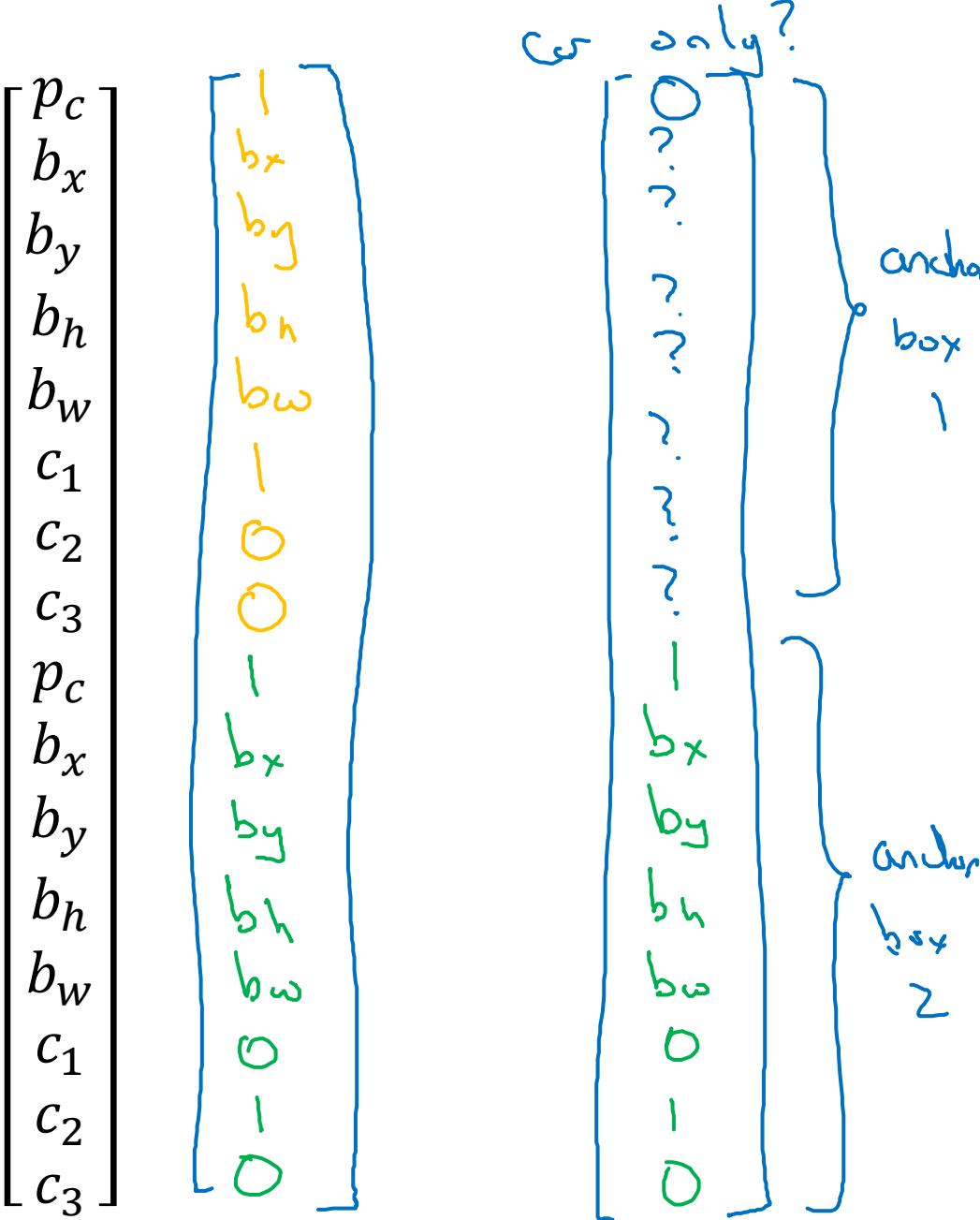
Anchor box example



Anchor box 1: Anchor box 2:



y =



Now just some additional details what if you have two anchor boxes but 3 objects in the same grid cell, that's one case that this algorithm doesn't handle it well, you know if you hopefully it won't happen but if it does this algorithm doesn't have a great way of handling it, we just implement some default tiebreaker for that case.

Or what if you have two objects associated the same grid cell but both of them have you know the same anchor box shape, again that's another case that this algorithm doesn't handle well. If you implement some default way of tiebreaking if that happens, hopefully this won't happen in your dataset, that won't happen much at all and so it shouldn't affect performance much.

Even though I've motivated anchor boxes as a way to deal with what happens if two objects appear in the same grid cell, in practice that happens quite rarely especially if you use a 19 by 19 rather than 3 by 3 grid, you know the chance of two object having the same midpoint out of these 361 cells. It happen but it doesn't happen that often.

The even better motivation or even better results that anchor boxes give you is it allows your learning algorithm to specialize better in particular if your data set has some tall skinny objects like pedestrians and some objects like cars then this allow your learning algorithm to specialize so that some of the outputs can specialize in detecting wide you know fat object like cars and some of the output units can specialize in detecting tall skinny like pedestrians.

People maybe just choose it by hand, you choose maybe 5 to 10 anchor box shapes that spans a variety of shapes that see to cover the types of objects you seem to detect.

As a much more advanced version just an advanced comment for those of you that have other knowledge of machine learning an even better way to do this in one of later YOLO research papers is to use a K-means algorithm to group together two types of object shapes you tend to get and if you use that to select a set of anchor boxes that this most stereotypically representative of the maybe multiple or there maybe dozens of object classes you're trying to detect but that's advanced way to automatically choose the anchor boxes.

And if you just choose by hand if a variety of shapes that reasonably span the set of object shapes you expect to detect some tall skinny ones some fat wide ones that should work easily as well.