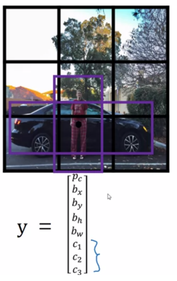
11 November 2020 Wednesday 09:13

Anchor Boxes

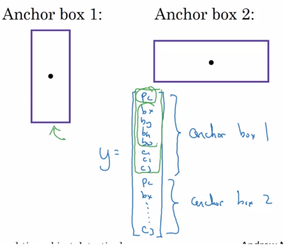
Grid



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

↙ only one object in each grid cell

← if there multiple objects whose centers rely on the same cell, then we have a problem!



↙ multiple obj in one cell

↙ anchor count is a hyperparam

↳ Each obj in training img is assigned to a grid cell containing objects midpoint and an anchor box w/ max IoU.

→ what if 2 obj appear in same grid cell?

→ How to choose anchor boxes?:

↳ YOLO uses k-means algo.

→ We need our net's predictors to be able to tell whether it is their job to predict apple or pear.

1. Create thousands of "anchor boxes" or "prior boxes" for each predictor that represent the ideal location, shape and size of the object it specializes in predicting.
2. For each anchor box, calculate which object's bounding box has the highest overlap divided by non-overlap. This is called Intersection Over Union or IOU.
3. If the highest IOU is greater than 50%, tell the anchor box that it should detect the object that gave the highest IOU.
4. Otherwise if the IOU is greater than 40%, tell the neural network that the true detection is ambiguous and not to learn from that example.
5. If the highest IOU is less than 40%, then the anchor box should predict that there is no object.

→ In RetinaNet, smallest anchor box size is 32x32.

As a general rule, you should ask yourself the following questions about your dataset before diving into training your model:

1. What is the smallest size box I want to be able to detect?
2. What is the largest size box I want to be able to detect?
3. What are the shapes the box can take? For example, a car detector might have short and wide anchor boxes as long as there is no chance of the car or the camera being turned on its side.

You can get a rough estimate of these by actually calculating the most extreme sizes and aspect ratios in the dataset. YOLO v3, another object detector, uses K-means to estimate the ideal bounding boxes. Another option is to learn the anchor box configuration.

Labeling Training Set Anchor Boxes

↳ need to assign class & bbox to each anchor box

↳ offset of gt bbox relative to anchor box.

How do we assign ground-truth bounding boxes to anchor boxes similar to them?

Assume that the anchor boxes in the image are $A_1, A_2, \ldots, A_{n_a}$ and the ground-truth bounding boxes are $B_1, B_2, \ldots, B_{n_b}$ and $n_a \geq n_b$. Define matrix $\mathbf{X} \in \mathbb{R}^{n_a \times n_b}$, where element x_{ij} in the i^{th} row and j^{th} column is the IoU of the anchor box A_i to the ground-truth bounding box B_j. First, we find the largest element in the matrix $\mathbf{X}$ and record the row index and column index of the element as i_1, j_1. We assign the ground-truth bounding box B_{j_1} to the anchor box A_{i_1}. Obviously, anchor box A_{i_1} and ground-truth bounding box B_{j_1} have the highest similarity among all the "anchor box–ground-truth bounding box" pairings. Next, discard all elements in the i_1th row and the j_1th column in the matrix $\mathbf{X}$. Find the largest remaining element in the matrix $\mathbf{X}$ and record the row index and column index of the element as i_2, j_2. We assign ground-truth bounding box B_{j_2} to anchor box A_{i_2} and then discard all elements in the i_2th row and the j_2th column in the matrix $\mathbf{X}$. At this point, elements in two rows and two columns in the matrix $\mathbf{X}$ have been discarded.

We proceed until all elements in the n_b column in the matrix $\mathbf{X}$ are discarded. At this time, we have assigned a ground-truth bounding box to each of the n_b anchor boxes. Next, we only traverse the remaining $n_a - n_b$ anchor boxes. Given anchor box A_i, find the bounding box B_j with the largest IoU with A_i according to the i^{th} row of the matrix $\mathbf{X}$, and only assign ground-truth bounding box B_j to anchor box A_i when the IoU is greater than the predetermined threshold.

↳ Background category for classification