下面看具体实现。代码是同时确定gt box是分配在哪一层的哪一个或几个anchor上。具体的类为GridAssigner，其中输入参数为：Bboxes：所有的anchor。box_responsible_flags： gt 第一步分配的anchor flags，主要是记录在候选anchor中分配。 和gt_bboxes。该类遍历batch，维护一个assigned_gt_inds，类似mask的概念，元素值会被分配为-1：忽略样本，0：负样本，正整数：正样本，同时数字代表负责的gt box的索引。具体步骤如下：

第一步，将所有的assigned_gt_inds设置为-1，默认为忽略样本。

第二步，将所有iou小于一定值例如0.5（或者在一定区间的），设置为0，置为负样本。gt box和全部anchor计算iou，这里的boxes为anchor，是带有位置信息的。获取的overlaps 尺度为gt box个数*全部anchor个数（这里为300+1200+4800=6300）。

```
overlaps = self.iou_calculator(gt_bboxes, bboxes) # 获取全部iou，size为gt个数X6300
max_overlaps, argmax_overlaps = overlaps.max(dim=0) # 找和所有gtbox最大的iou，size为6300
assigned_gt_inds[(max_overlaps >= 0) & (max_overlaps <= self.neg_iou_thr)] = 0 #如果小
```

第三步，将全部iou中，非负责gt的（记录在box_responsible_flags，非中心点grid cell的anchor）置为-1，该步骤首先排除掉非中心点grid cell的anchor。因为排除掉的部分肯定不是正样本。

```
#获取和哪一个gt最大的iou，size为6300，和上一步类似，不过获取的都是负责gt box的grid cell里的
max_overlaps, argmax_overlaps = overlaps.max(dim=0)
# 获取的iou和一定阈值对比，例如0.5，大于该值，设置为正样本。
## 可见这一步是将gt box对应的grid cell 里面大于一定阈值的anchor设置为正样本，可能是多个anch
pos_inds = (max_overlaps > self.pos_iou_thr) & box_responsible_flags.type(torch.bool)
assigned_gt_inds[pos_inds] = argmax_overlaps[pos_inds] + 1
#--------------------------------------------------------------------------------#
#--------------------------------------------------------------------------------#
#获取全部gt和哪一个anchor最大的iou，尺度为gt的数目，例如有2个gt，那么size就是2
gt_max_overlaps, gt_argmax_overlaps = overlaps.max(dim=1)
# 遍历gt box，找到其最大的anchor，且在负责的grid cell中，设置为正样本。
# 因为上一步，有些gt box并找不到iou大于阈值的anchor，这部分也是要预测的，所以退而求其次，找最
for i in range(num_gts):
  if gt_max_overlaps[i] > self.min_pos_iou:
    if self.gt_max_assign_all:
      max_iou_inds = (overlaps[i, :] == gt_max_overlaps[i]) & \
          box_responsible_flags.type(torch.bool)
      assigned_gt_inds[max_iou_inds] = i + 1
```

至此，全部anchor全部分配完成，总结一下：

1. 全部anchor，和gt box的iou小于阈值的，设置为负样本；
2. 正样本来自两部分： 第一是gt box对应的grid cell里的anchor，iou大于阈值的。第二部分是gt box对应grid cell里的anchor，和gt box iou 最大的那一个；
3. 其余部分，设置为忽略样本；

可以看出，上面2中的第二部分的正样本是最后计算了，因此理论上所有gt box都会分配一个和自己iou最大的anchor。如果预先被2中第一部分分配了，有可能会被其他gt box挤走，也就是标签重写现象。这个以后可以重点分析一下。还可以看出，一个gt box 可以有多个anchor，但是一个anchor只能负责一个gt box。可以理解为，正负样本的分配在训练该样本之前已经做好了，和训练的好坏以及预测的结果并无关系。当然还有另外一种实现方式是： 忽略样本由训练过程中的真实预测的box和gt box算iou，较大的且没有被分配到的为忽略样本，是一种动态的分配方式。究竟哪一种方式好我没有去深入思考和测试，知道的小伙伴可以告诉我。