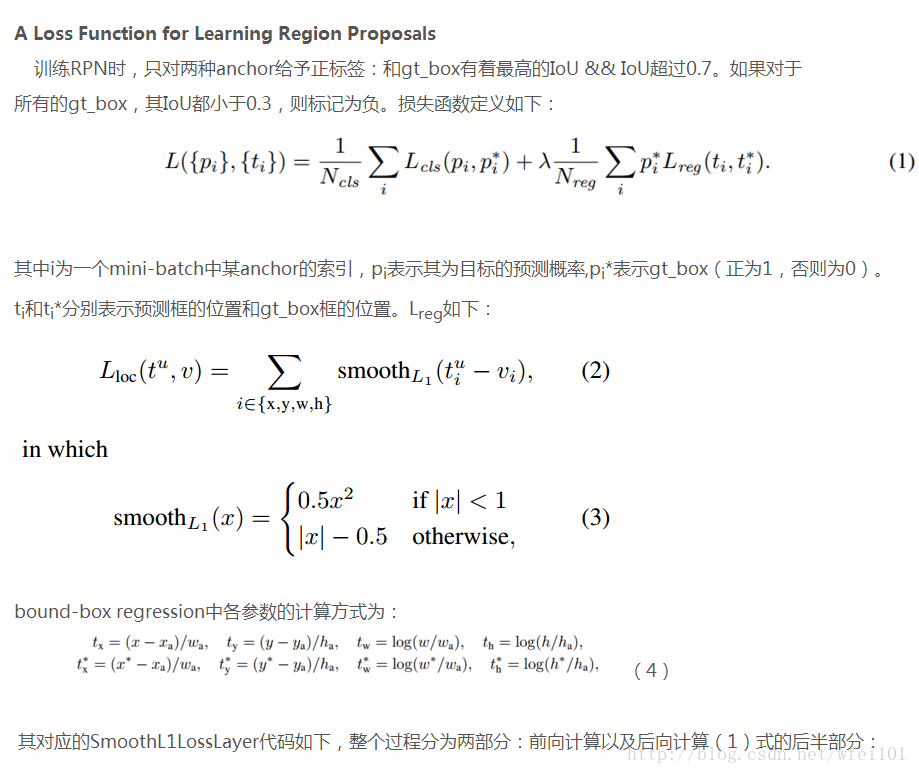
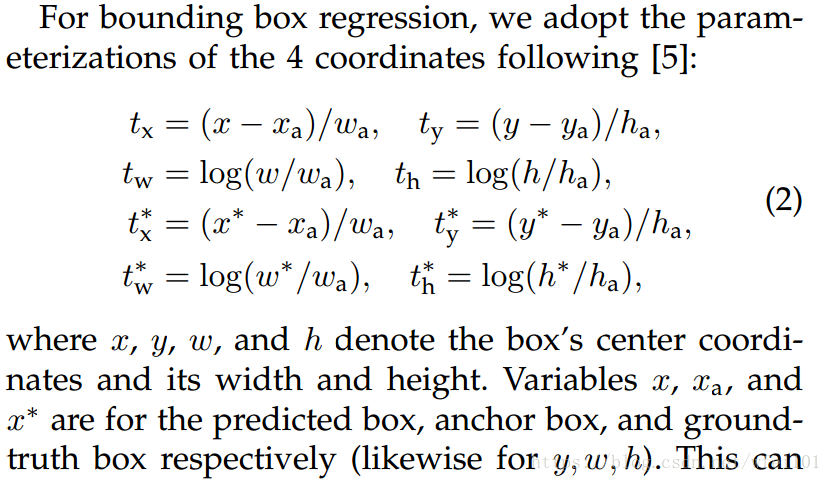
**各种损失函数**

**一、SmoothL1Loss**

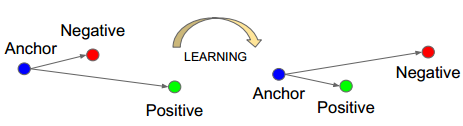




**二、Triplet Loss**

**1、简介**

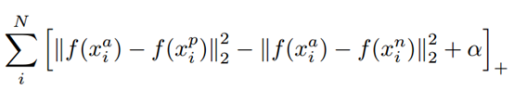
这里将Triplet Loss翻译为三元组损失，其中的三元也就是如下图的Anchor、Negative、Positive，如下图所示通过Triplet Loss的学习后使得Positive元和Anchor元之间的距离最小，而和Negative之间距离最大。其中Anchor为训练数据集中随机选取的一个样本，Positive为和Anchor属于同一类的样本，而Negative则为和Anchor不同类的样本。



这也就是说通过学习后，使得同类样本的positive样本更靠近Anchor，而不同类的样本Negative则远离Anchor。

**2、Triplet loss目标函数**

下面为Triplet loss的目标函数：



上式中的||\*||为欧式距离，所以：

https://img-blog.csdn.net/20170814123213586表示的是Positive元和Anchor之间的欧式距离度量。

https://img-blog.csdn.net/20170814123214129表示的是Negative和Anchor之间的欧式距离度量。

https://img-blog.csdn.net/20170814123214463是指x\_a与x\_n之间的距离和x\_a与x\_p之间的距离之间有一个最小的间隔。 另外这里距离用欧式距离度量，+表示[]内的值大于零的时候，取该值为损失，小于零的时候，损失为零。

由目标函数可以看出: 当x\_a与x\_n之间的距离 < x\_a与x\_p之间的距离加https://img-blog.csdn.net/20170814123214463时，[]内的值大于零，就会产生损失。 当x\_a与x\_n之间的距离 >= x\_a与x\_p之间的距离加https://img-blog.csdn.net/20170814123214463时，损失为零。

**3、内涵**

我们的目的就是使 loss 在训练迭代中下降的越小越好，也就是要使得 Anchor 与 Positive 越接近越好，Anchor 与 Negative 越远越好。基于上面这些，分析一下 margin 值的取值。

当 margin 值越小时，loss 也就较容易的趋近于 0，于是 Anchor 与 Positive 都不需要拉的太近，Anchor 与 Negative 不需要拉的太远，就能使得 loss 很快的趋近于 0。这样训练得到的结果，不能够很好的区分相似的图像。

当 Anchor 越大时，就需要使得网络参数要拼命地拉近 Anchor、Positive 之间的距离，拉远 Anchor、Negative 之间的距离。如果 margin 值设置的太大，很可能最后 loss 保持一个较大的值，难以趋近于 0 。

因此，设置一个合理的 margin 值很关键，这是衡量相似度的重要指标。简而言之，margin 值设置的越小，loss 很容易趋近于 0 ，但很难区分相似的图像。margin 值设置的越大，loss 值较难趋近于 0，甚至导致网络不收敛，但可以较有把握的区分较为相似的图像。