# YOLO检测模型蒸馏

作者：张松

# 知识蒸馏概述

知识蒸馏（Knowledge Distilling）是模型压缩的一种方法，是指利用已经训练的一个较复杂的Teacher模型，指导一个较轻量的Student模型训练，从而在减小模型大小和计算资源的同时，尽量保持原Teacher模型的准确率的方法。即用一个复杂网络（teacher network）学到的东西去辅助训练一个简单网络（student network）。

# 背景

引用知乎相关问题的回答粘贴如下，将KD方法的motivation讲的很清楚。

Knowledge Distill是一种简单弥补分类问题监督信号不足的办法。传统的分类问题，模型的目标是将输入的特征映射到输出空间的一个点上，例如在著名的Imagenet比赛中，就是要将所有可能的输入图片映射到输出空间的1000个点上。这么做的话这1000个点中的每一个点是一个one hot编码的类别信息。这样一个label能提供的监督信息只有log(class)这么多bit。然而在KD中，我们可以使用teacher model对于每个样本输出一个连续的label分布，这样可以利用的监督信息就远比one hot的多了。另外一个角度的理解，大家可以想象如果只有label这样的一个目标的话，那么这个模型的目标就是把训练样本中每一类的样本强制映射到同一个点上，这样其实对于训练很有帮助的类内variance和类间distance就损失掉了。然而使用teacher model的输出可以恢复出这方面的信息。具体的举例就像是paper中讲的， 猫和狗的距离比猫和桌子要近，同时如果一个动物确实长得像猫又像狗，那么它是可以给两类都提供监督。综上所述，KD的核心思想在于”打散”原来压缩到了一个点的监督信息，让student模型的输出尽量match teacher模型的输出分布。其实要达到这个目标其实不一定使用teacher model，在数据标注或者采集的时候本身保留的不确定信息也可以帮助模型的训练。

# 知识蒸馏方式

据迁移的方法不同可以简单分为基于目标蒸馏（也称为Soft-target蒸馏或Logits方法蒸馏）和基于特征蒸馏的算法两个大的方向

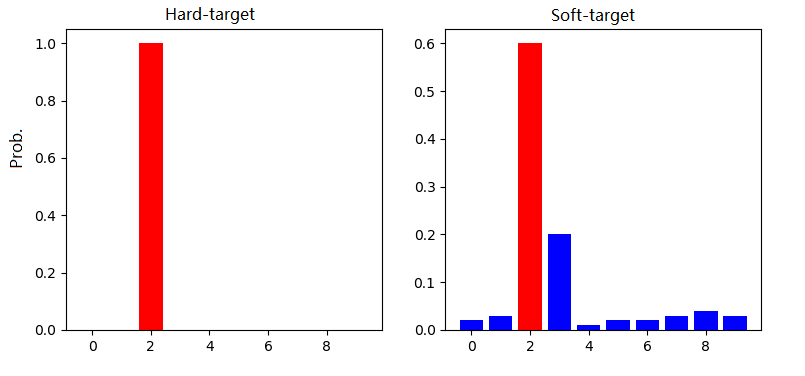
## 1 目标蒸馏-Logits方法

目标蒸馏方法中最经典的论文就是来自于2015年Hinton发表的一篇神作《Distilling the Knowledge in a Neural Network》。下面以这篇神作为例，讲讲目标蒸馏方法的原理。

在这篇论文中，Hinton将问题限定在分类问题下，分类问题的共同点是模型最后会有一个softmax层，其输出值对应了相应类别的概率值，使用softmax层输出的类别的概率来作为“Soft-target” 。

### 1.1 Hard-target 和 Soft-target

传统的神经网络训练方法是定义一个损失函数，目标是使预测值尽可能接近于真实值（**Hard- target**），损失函数就是使神经网络的损失值和尽可能小。这种训练过程是对ground truth求极大似然。在知识蒸馏中，是使用大模型的类别概率作为Soft-target的训练过程。

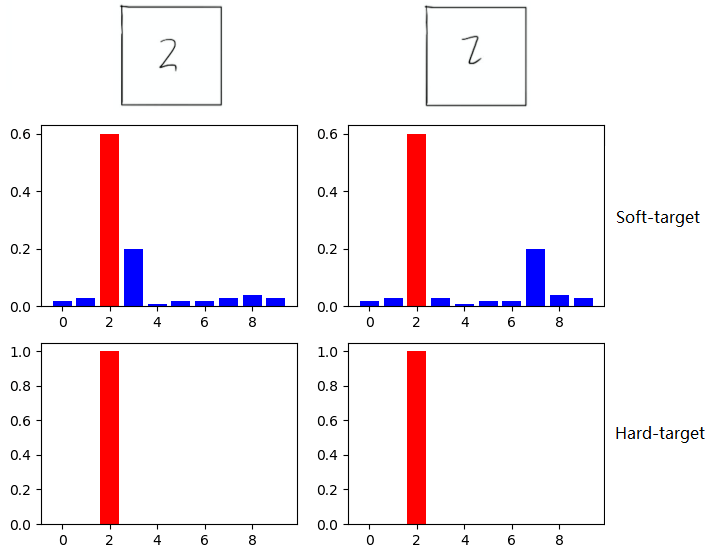


* **Hard-target**：原始数据集标注的 one-shot 标签，除了正标签为 1，其他负标签都是 0。

* **Soft-target**：Teacher模型softmax层输出的类别概率，每个类别都分配了概率，正标签的概率最高。

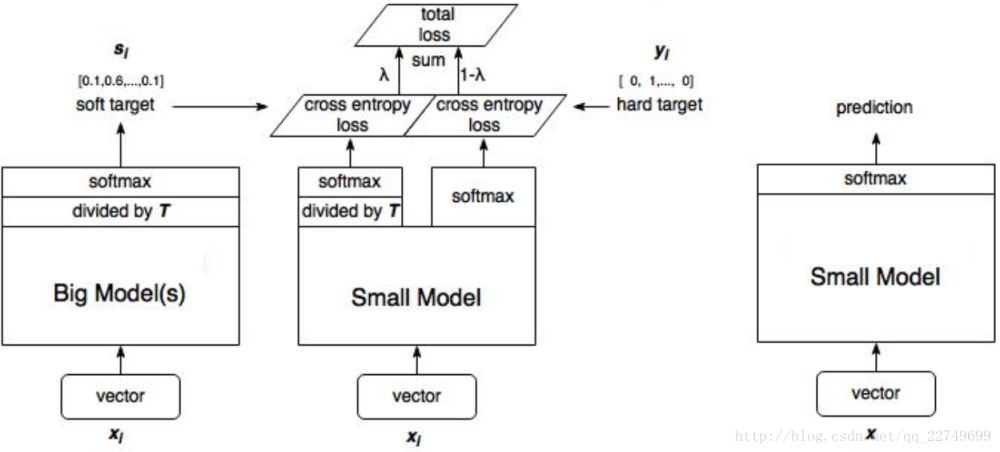
知识蒸馏用Teacher模型预测的 Soft-target 来辅助 Hard-target 训练 Student模型的方式为什么有效呢？softmax层的输出，除了正例之外，**负标签也带有Teacher模型归纳推理的大量信息**，比如某些负标签对应的概率远远大于其他负标签，则代表 Teacher模型在推理时认为该样本与该负标签有一定的相似性。而在传统的训练过程(Hard-target)中，所有负标签都被统一对待。也就是说，知识蒸馏的训练方式使得每个样本给Student模型带来的信息量大于传统的训练方式。

如在MNIST数据集中做手写体数字识别任务，假设某个输入的“2”更加形似"3"，softmax的输出值中"3"对应的概率会比其他负标签类别高；而另一个"2"更加形似"7"，则这个样本分配给"7"对应的概率会比其他负标签类别高。这两个"2"对应的Hard-target的值是相同的，但是它们的Soft-target却是不同的，由此我们可见Soft-target蕴含着比Hard-target更多的信息。



在使用 Soft-target 训练时，Student模型可以很快学习到 Teacher模型的推理过程；而传统的 Hard-target 的训练方式，所有的负标签都会被平等对待。因此，Soft-target 给 Student模型带来的信息量要大于 Hard-target，并且Soft-target分布的熵相对高时，其Soft-target蕴含的知识就更丰富。

### 1.2 具体方法



蒸馏过程的目标函数由distill loss(对应Soft-target)和Student loss(对应Hard-target)加权得到。如下所示：

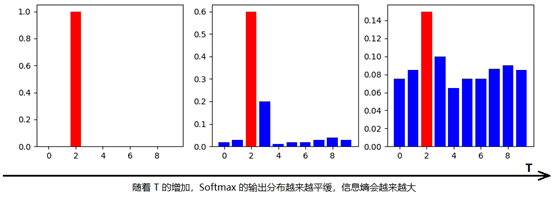


1.对于分类网络来说,原始的softmax函数：



蒸馏通过引入温度因子T来控制每个软目标的重要性， 

其中，当T=1时，就是普通的softmax变换。这里令T>1，就得到了软化的softmax。（这个很好理解，除以一个比1大的数，相当于被squash了，线性的sqush被指数放大，差距就不会这么大了）。得到的soft target，相比于one-hot的ground truth或softmax的prob输出，这个软化之后的target能够提供更多的类别间和类内信息。



hard loss(ground truth)是和真实标签的loss，一般是交叉熵loss。

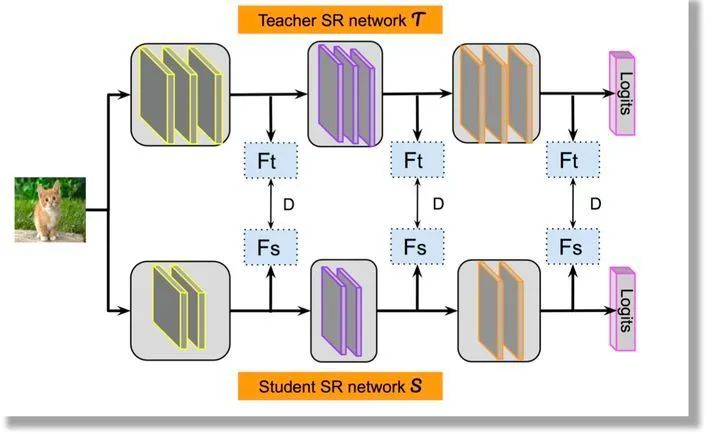
soft loss(teacher prediction)一般是kdloss，kdloss公式：



P代码teacher模型的概率分布，Q代表学生模型的概率分布，当P和Q越相近，KL损失值越趋于0。

## 2 特征蒸馏

另外一种知识蒸馏思路是特征蒸馏方法，Student学习Teacher网络结构中的中间层特征。最早采用这种模式的工作来自于论文《FITNETS：Hints for Thin Deep Nets》，它强迫Student某些中间层的网络响应，要去逼近Teacher对应的中间层的网络响应。这种情况下，Teacher中间特征层的响应，就是传递给Student的知识。一般的方法是计算两个feature map的MSEloss。



# YOLOv3检测蒸馏

## 差异：

由于YOLOv3检测框的位置输出为正无穷到负无穷的连续值，和上面将的分类离散kdloss不同，而且由于yolo是基于anchor的one stage模型，head out中99%都是背景预测。

Object detection at 200 Frames Per Second论文中指出，

直接在Yolo算法中引入distillation loss会有一些问题，因为目前的network distillation算法主要是针对RCNN系列的object detection算法（或者叫two stage系列）。对于two stage的object detection算法而言，其最后送给检测网络的ROI数量是很少的（默认是128个），而且大部分都是包含object的bbox，因此针对这些bbox引入distillation loss不会有太大问题。但是对于Yolo这类one stage算法而言，假设feature map大小是13\*13，每个grid cell预测5个bbox，那么一共就会生成13\*13\*5=845个bbox，而且大部分都是背景（background）。如果将大量的背景区域传递给student network，就会导致网络不断去回归这些背景区域的坐标以及对这些背景区域做分类，这样训练起来模型很难收敛。因此，作者利用Yolo网络输出的objectness对distillation loss做一定限定，换句话说，只有teacher network的输出objectness较高的bbox才会对student network的最终损失函数产生贡献，这就是objectness scaled distillation。

原来Yolo算法的损失函数，包含3个部分（公式1）：1、objectness loss，表示一个bbox是否包含object的损失；2、classification loss，表示一个bbox的分类损失；3、regression loss，表示一个bbox的坐标回归损失。

Yolo损失：回归损失+目标损失+分类损失，核心的算法如下图：

