

trick1

导师：付

目录

1/
基础数据增强

2/
LabelSmoothing

3/
cutout, cutmix, mixup

4/
snapmix, Bi-Tempered Logistic Loss

1、基础数据增强



基础数据增强

数据增强也叫数据扩增，意思是在不实质性的增加数据的情况下，让有限的数据产生等价于更多数据的价值。

- 1 几何变换类
- 2 颜色变换类等

基础数据增强

1 几何变换类

几何变换类即对图像进行几何变换，包括翻转，旋转，裁剪，变形，缩放等各类操作



基础数据增强



深度之眼
deepshare.net

2 颜色变换类

常见的包括噪声、模糊、颜色变换、擦除、填充



AutoAugment: Learning Augmentation Policies from Data

一种自动搜索合适数据增强策略的方法 AutoAugment，该方法创建一个数据增强策略的搜索空间，利用搜索算法选取适合特定数据集的数据增强策略。此外，从一个数据集中学到的策略能够很好地迁移到其它相似的数据集上。

<https://arxiv.org/pdf/1805.09501.pdf>

2、LabelSmoothing

LabelSmoothin

标签平滑 (Label smoothing), 像L1、L2和dropout一样, 是机器学习领域的一种正则化方法, 通常用于分类问题, 目的是防止模型在训练时过于自信地预测标签, 改善泛化能力差的问题。

LabelSmoothin



深度之眼
deepshare.net

为什么需要label smoothing?

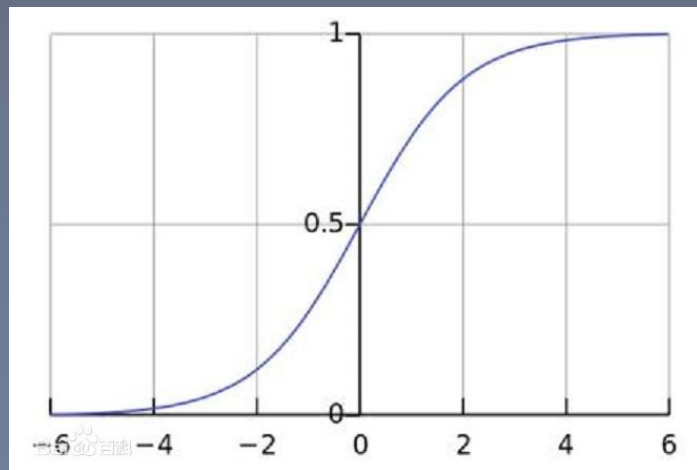
对于分类问题，我们通常认为训练数据中标签向量的目标类别概率应为1，非目标类别概率应为0。传统的one-hot编码的标签向量 y_i 为，

$$y_i = \begin{cases} 1, & i = \text{target} \\ 0, & i \neq \text{target} \end{cases}$$

在训练网络时，最小化损失函数 $H(y, p) = -\sum_i^K y_i \log p_i$ ，其中 p_i 由对模型倒数第二层输出的logits向量 z 应用Softmax函数计算得到，

$$p_i = \frac{\exp(z_i)}{\sum_j^K \exp(z_j)}$$

传统one-hot编码标签的网络学习过程中，鼓励模型预测为目标类别的概率趋近1，非目标类别的概率趋近0，即最终预测的logits向量（logits向量经过softmax后输出的就是预测的所有类别的概率分布）中目标类别 z_i 的值会趋于无穷大，使得模型向预测正确与错误标签的logit差值无限增大的方向学习，而过大的logit差值会使模型缺乏适应性，对它的预测过于自信。在训练数据不足以覆盖所有情况下，这就会导致网络过拟合，泛化能力差，而且实际上有些标注数据不一定准确，这时候使用交叉熵损失函数作为目标函数也不一定是最优的了。



label smoothing的数学定义

label smoothing结合了均匀分布，用更新的标签向量 \hat{y}_i 来替换传统的ont-hot编码的标签向量 y_{hot} :

$$\hat{y}_i = y_{hot}(1 - \alpha) + \alpha/K$$

其中K为多分类的类别总个数， α 是一个较小的超参数（一般取0.1），即

$$\hat{y}_i = \begin{cases} 1 - \alpha, & i = target \\ \alpha/K, & i \neq target \end{cases}$$

这样，标签平滑后的分布就相当于往真实分布中加入了噪声，避免模型对于正确标签过于自信，使得预测正负样本的输出值差别不那么大，从而避免过拟合，提高模型的泛化能力。





3、cutout cutmix mixup

cutout cutmix mixup

Mixup:将随机的两张样本按比例混合, 分类的结果按比例分配;

Cutout:随机的将样本中的部分区域cut掉, 并且填充0像素值, 分类的结果不变;

CutMix:就是将一部分区域cut掉但不填充0像素而是随机填充训练集中的其他数据的区域像素值, 分类结果按一定的比例分配

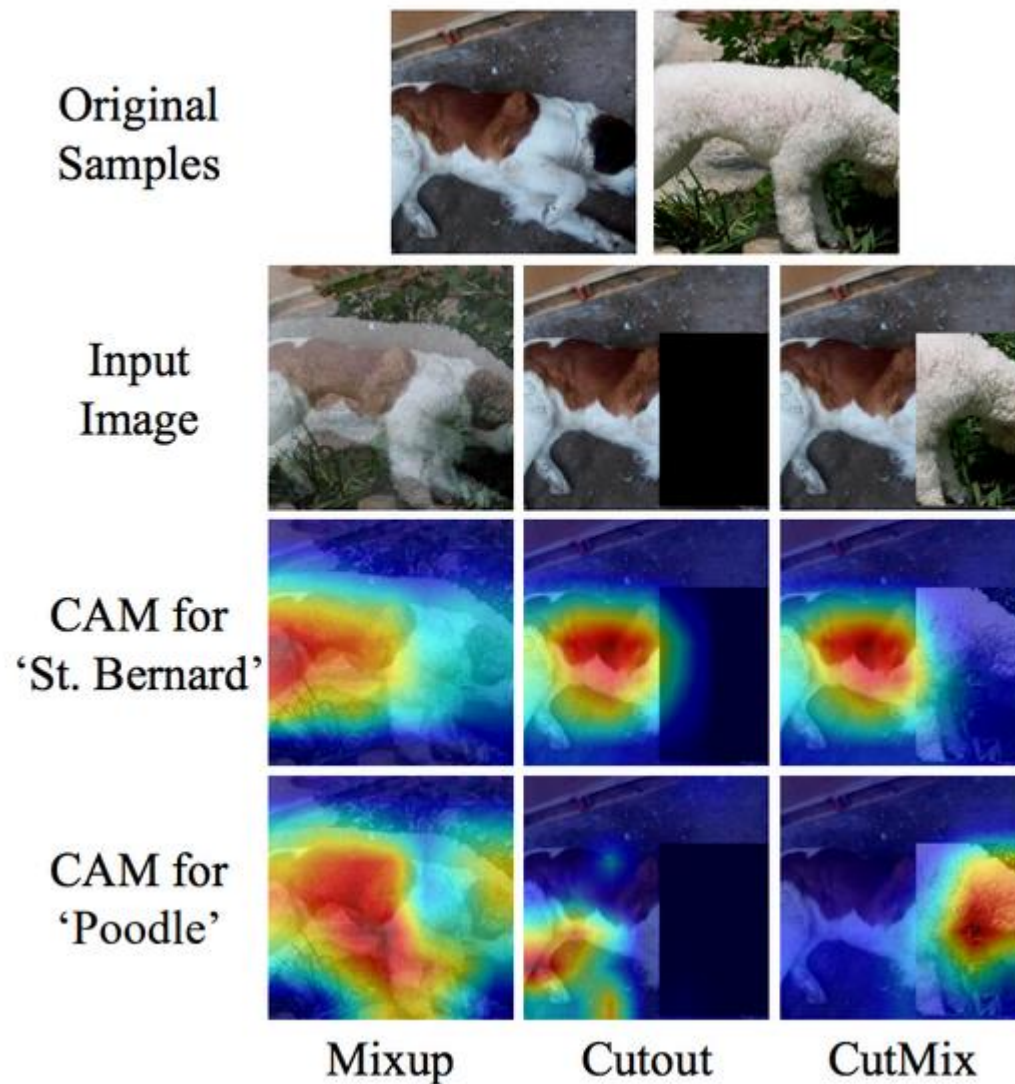
	ResNet-50	Mixup [48]	Cutout [3]	CutMix
Image				
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0	Dog 0.6 Cat 0.4

cutout cutmix mixup



深度之眼
deepshare.net

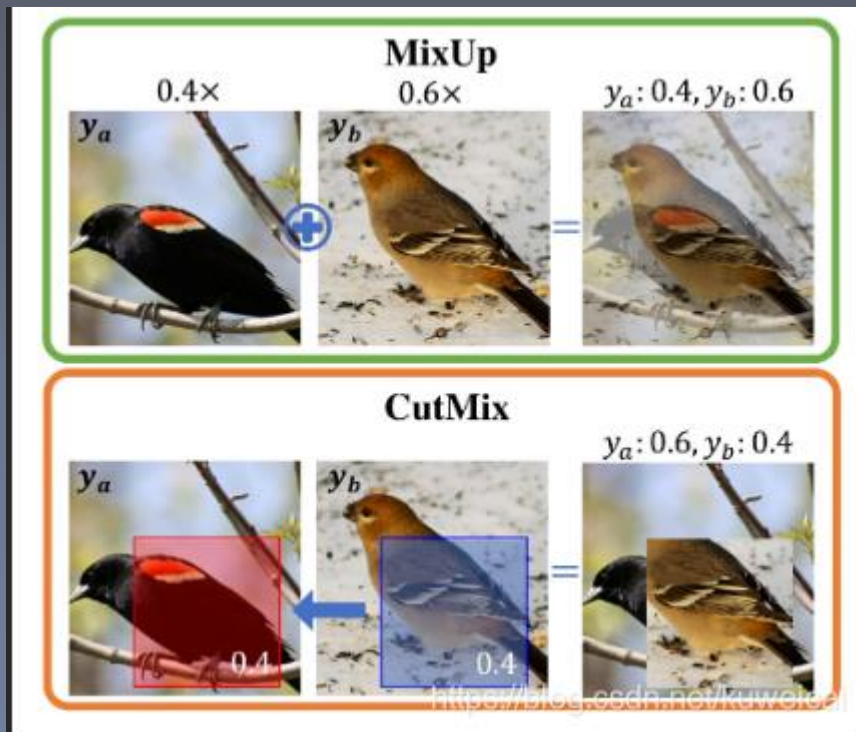
CutMix的操作使得模型能够从一幅图像上的局部视图上识别出两个目标，提高训练的效率。由图可以看出，Cutout能够使得模型专注于目标较难区分的区域（腹部），但是有一部分区域是没有任何信息的，会影响训练效率；Mixup的话会充分利用所有的像素信息，但是会引入一些非常不自然的伪像素信息



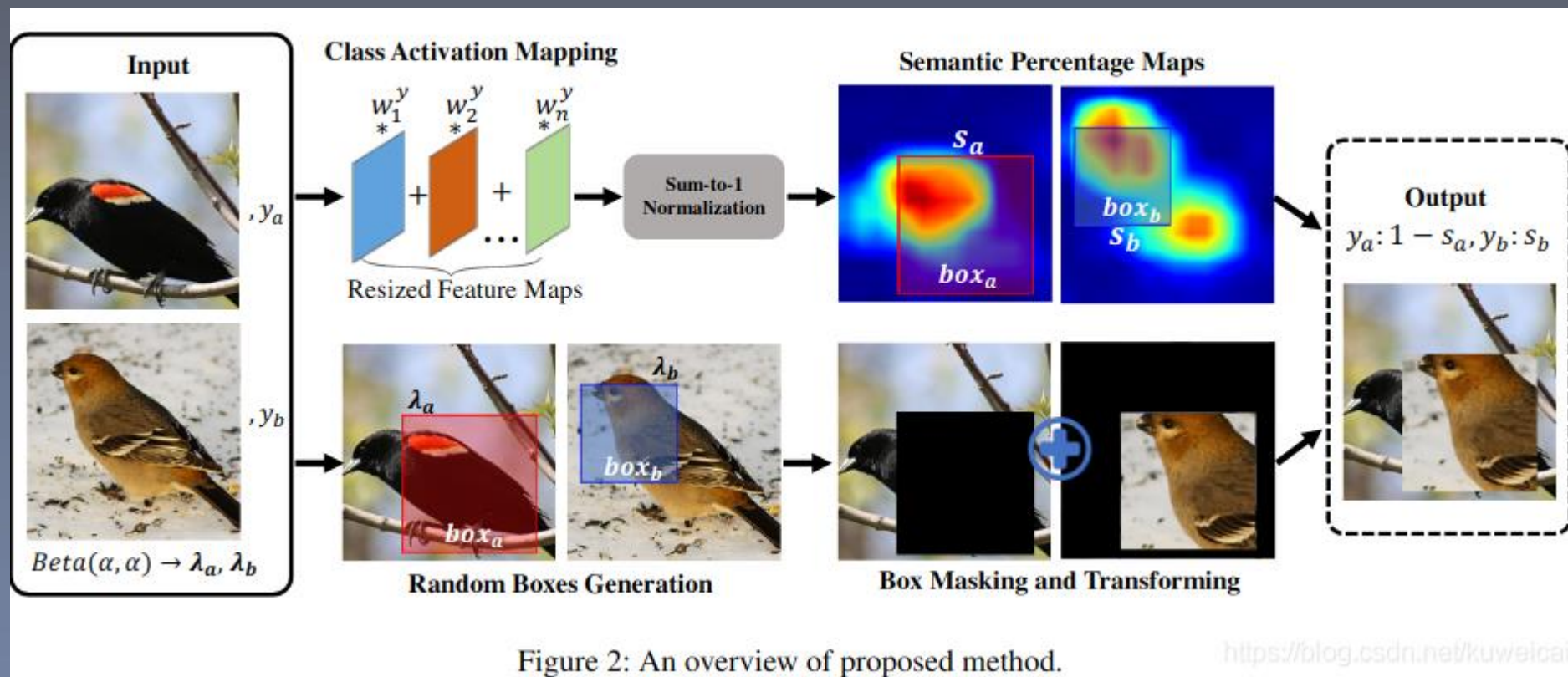
4、snapmix, Bi-Tempered Logistic Loss

usage method of NNI

snapmix



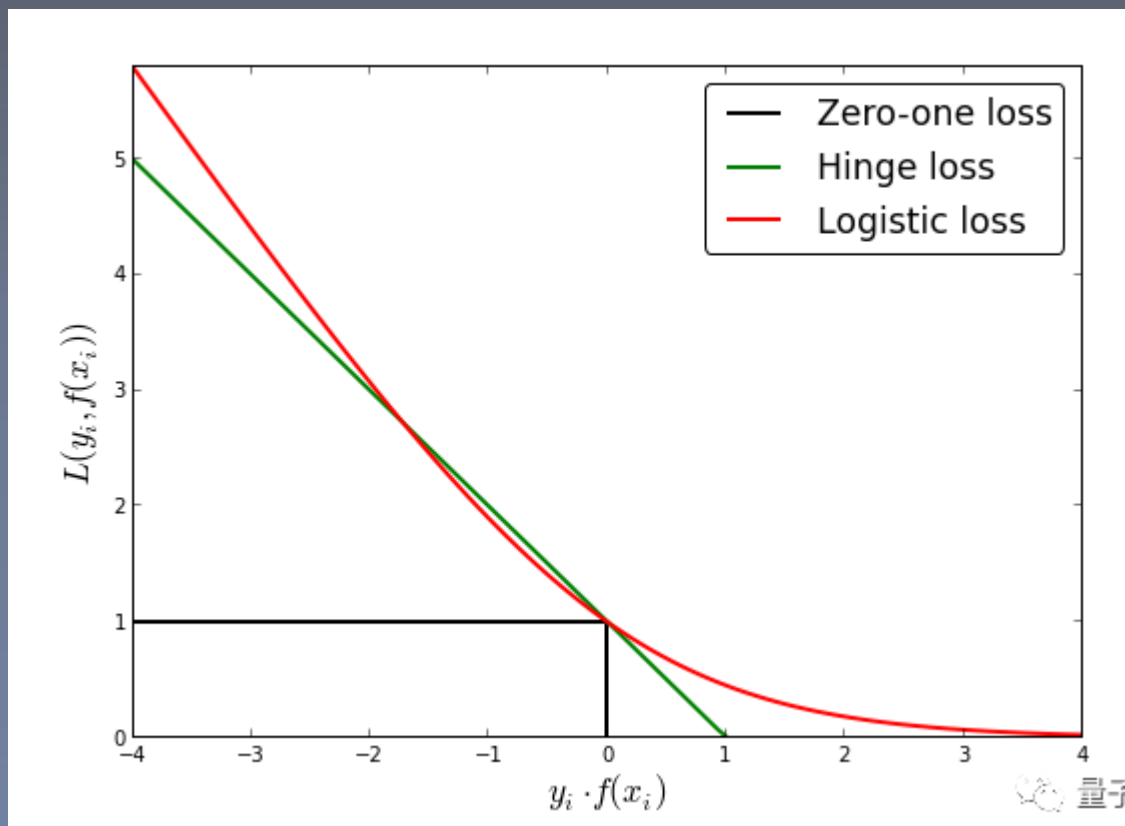
SnapMix 的思想很简单，既然在生成 label 的时候，直接在原图中以被 cut 掉的面积作为权重不合理，那作者就针对这一不合理作出了改进。



Bi-Tempered Logistic Loss

训练数据集里的标签通常不会都是正确的，比如图像分类，如果有人错误地把猫标记成狗，将会对训练结果造成不良的影响。

Bi-Tempered Logistic Loss



1、远离的异常值会支配总体的损失
逻辑损失函数对异常值非常敏感。这是因为损失函数的没有上界，而错误的标记数据往往远离决策边界。这就导致异常大的错误数值会拉伸决策边界，对训练的结果造成不良影响，并且可能会牺牲其他的正确样本。

2、错误的标签的影响会扩展到分类的边界上
神经网络的输出是一个矢量激活值，一般对于分类问题，我们使用的是softmax，将激活值表示为分别属于每个分类的概率。
由于逻辑损失的这种传递函数的尾部以指数方式快速衰减，因此训练过程将倾向于使边界更接近于错误标记的示例，以保证不属于该分类的函数值更接近于0。

Bi-Tempered Logistic Loss

在无噪声情况下，两种损失都能产生良好的决策边界，从而成功地将这两种类别分开。

小边距噪声，即噪声数据接近于决策边界。可以看出，由于softmax尾部快速衰减的原因，逻辑损失会将边界拉伸到更接近噪声点，以补偿它们的低概率。而双稳态损失函数有较重的尾部，保持边界远离噪声样本。

大边距噪声，即噪声数据远离决策边界。由于双稳态损失函数的有界性，可以防止这些远离边界的噪声点将决策边界拉开。

最后一个实验是随机噪声，噪声点随机分布在矢量空间中。逻辑损失受到噪声样本的高度干扰，无法收敛到一个良好的决策边界。而双稳态损失可以收敛到与无噪声情况几乎相同的结果上。

