

基于改进损失函数的 YOLOv3 网络^①



吕 铄^{1,2,3}, 蔡 烜⁴, 冯 瑞^{1,2,3}

¹(复旦大学 计算机科学技术学院, 上海 201203)

²(上海视频技术与系统工程研究中心, 上海 201203)

³(复旦大学 智能信息处理实验室, 上海 201203)

⁴(公安部第三研究所物联网技术研发中心, 上海 201204)

摘 要: 为了提高卷积神经网络在目标检测的精度, 本文提出了一种基于改进损失函数的 YOLOv3 网络. 该网络模型应用一种新的损失函数 Tan-Squared Error (TSE), 将原有的平方和损失 (Sum Squared Error, SSE) 函数进行转化, 能更好地计算连续变量的损失; TSE 能有效减低 Sigmoid 函数梯度消失的影响, 使模型收敛更加快速. 在 VOC 数据集上的实验结果表明, 与原网络模型的表现相比, 利用 TSE 有效提高了检测精度, 且收敛更加快速.

关键词: 深度学习; 损失函数; 目标检测; 卷积神经网络

引用格式: 吕铄, 蔡烜, 冯瑞. 基于改进损失函数的 YOLOv3 网络. 计算机系统应用, 2019, 28(2): 1-7. <http://www.c-s-a.org.cn/1003-3254/6772.html>

YOLOv3 Network Based on Improved Loss Function

LYU Shuo^{1,2,3}, CAI Xuan⁴, FENG Rui^{1,2,3}

¹(School of Computer Science, Fudan University, Shanghai 201203, China)

²(Shanghai Engineering Research Center for Video Technology and System, Shanghai 201203, China)

³(Laboratory of Intelligent Information Processing, Fudan University, Shanghai 201203, China)

⁴(R & D Center of Internet of Things, The Third Research Institute of Ministry of Public Security, Shanghai 201204, China)

Abstract: To improve the object detect precision of Convolutional Neural Network (CNN), we present a YOLOv3 network which based on improved loss function. This network model uses a new loss function Tan-Squared Error (TSE) which transferred from primary Sum Squared Error (SSE), and works better on continuous variable error computing. Meanwhile, the properties of TSE could decrease the impact of vanishing gradient problem in sigmoid function, and speed up model converging. The experiment results in Pascal VOC dataset show that TSE improves the detect precision effectively compared with the performance of primary network model, and the convergence is accelerated.

Key words: deep learning; loss function; object detection; Convolutional Neural Network (CNN)

目标检测具有广阔的发展前景和巨大的商业价值, 已经成为国内外相关从业者的研究热点, 在智能安防、自动驾驶等领域具有广泛应用. 经典的目标检测方法有 Dalal 于 2005 年提出的基于 HOG 特征的检测方法^[1], Felzenswalb 等人于 2008 年提出的可变形组件模型 (Deformable Part Model, DPM) 检测方法^[2], 该方

法先利用梯度算子计算出目标物体的 HOG 特征并采用滑动窗口+SVM 的方法进行分类, 在目标检测问题中表现良好.

与经典方法相比, 神经网络提取特征能力强, 准确率高, 在计算机文本、图像分析等领域取得令人瞩目的成果, 受益于深度学习的快速发展, 基于卷积神

① 基金项目: 国家重点研发计划 (2017YFC0803700); 上海市科委项目 (17511101702); 复旦大学工程与应用技术研究院先导项目 (gyy2917-003)

Foundation item: National Key Research and Development Program of China (2017YFC0803700); Program of Science and Technology Commission of Shanghai Municipality (17511101702); Pilot Program of Academy for Engineering & Technology, Fudan University (gyy2917-003)

收稿时间: 2018-08-12; 修改时间: 2018-09-05; 采用时间: 2018-09-18; csa 在线出版时间: 2019-01-28

神经网络 (CNN) 的目标检测模型层出不穷, 检测效果不断提升, Girshick R 等人于 2014 年提出 R-CNN 网络^[3], 采用选择性搜索 (selective search) 方法代替传统的滑动窗口, 将 VOC2012 数据集上目标检测平均准确率 (mean Average Precision, mAP) 提高了 30%; Girshick R 和 Ren SQ 等人相继提出了 Fast R-CNN^[4]与 Faster R-CNN^[5]网络, Faster R-CNN 采用区域推荐网络 (Region Proposal Network, RPN) 生成候选框, 再对这些候选框进行分类和坐标回归, 检测精度大幅提升, 检测速度约 5 fps, 这些方法由于生成候选框和进行预测分成两个步骤进行, 所以称为 Two-Stage 方法, 同时进行这两种操作的网络称为 One-Stage 方法, 代表有 YOLO^[6], SSD^[7]等. 2016 年 Redmon J 等提出 YOLO 网络, 其特点是将生成候选框与分类回归合并成一个步骤, 预测时特征图被分成 $S \times S$ (S 为常数, 在 YOLOv1 中取 7) 个 cell, 对每个 cell 进行预测, 这就大大降低了计算复杂度, 加快了目标检测的速度, 帧率最高可达 45 fps, 之后, Redmon J 再次提出了 YOLOv2^[8], 与前代相比, 在 VOC2007 测试集上的 mAP 由 67.4% 提高到 76.8%, 然而由于一个 cell 只负责预测一个物体, 面对被遮挡目标的识别表现不够好. 2018 年 4 月, YOLO 发布第三个版本 YOLOv3^[9], 在 COCO 数据集上的 mAP-50 由 YOLOv2 的 44.0% 提高到 57.9%, 与 mAP61.1% 的 RetinaNet^[10]相比, RetinaNet 在输入尺寸 500×500 的情况下检测速度约 98 ms/帧, 而 YOLOv3 在输入尺寸 416×416 时检测速度可达 29 ms/帧, 在保证速度的前提下, 达到了很高的准确率.

为了实现实时分析的目标, 本文针对 One-Stage 方法的代表 YOLOv3 网络模型进行研究和改进, 根据坐标预测值和 Sigmoid 函数的特殊性质, 将原版的损失函数进行优化, 不仅减少了梯度消失的情况, 而且可以使网络收敛更加快速, 同时通过分析数据集中目标的分布情况修正 anchor box, 使得 anchor box 更加符合目标尺寸, 增强收敛效果, 在 Pascal VOC^[11]数据集上的实验表明, 在不影响检测速度的情况下准确率提高了 1 个百分点左右, 且收敛速度变快, 使目标检测能力进一步提升.

1 传统 YOLOv3 网络

YOLO 网络将目标检测问题转化为回归问题, 合并分类和定位任务到一个步骤, 直接预测物体的位置

及类别, 检测速度可以满足实时分析的要求. YOLOv3 包含了新的特征提取网络 Darknet-53, 以及三种尺度的 YOLO 层, 也就是预测层. 通过在三种尺度上进行预测的方法, 有效的增强了对不同大小物体及被遮挡物体的检测效果, 并引入跃层连接以强化收敛效果, 同时采用随机多尺度训练的方式增强了鲁棒性.

1.1 检测过程

YOLOv3 提出了新的提取图片特征的网络 Darknet53, 作为全卷积网络, darknet53 主要由卷积层、Batch Normalization 及跃层连接组成, 激活函数采用 LeakyRelu, 其网络结构如图 1 所示.

类别	通道数	卷积核大小步长	输出尺寸
卷积 + BN + leakyrelu	323×3		1
卷积 + BN + leakyrelu	643×3		2
1 ×	卷积 + BN + leakyrelu	321×3	1
	卷积 + BN + leakyrelu	643×3	1
跃层连接			
卷积 + BN + leakyrelu	1283×3		2
2 ×	卷积 + BN + leakyrelu	641×3	1
	卷积 + BN + leakyrelu	1283×3	1
跃层连接			
卷积 + BN + leakyrelu	256×3		2
8 ×	卷积 + BN + leakyrelu	1281×3	1
	卷积 + BN + leakyrelu	2563×3	1
跃层连接			
卷积 + BN + leakyrelu	5123×3		2
8 ×	卷积 + BN + leakyrelu	2563×3	1
	卷积 + BN + leakyrelu	5123×3	1
跃层连接			
卷积 + BN + leakyrelu	10243×3		2
4 ×	卷积 + BN + leakyrelu	5121×1	1
	卷积 + BN + leakyrelu	10243×3	1
跃层连接			
YOLO 卷积 (预测)	$3 \times (5 + \text{类别数})1 \times 1$		113×13
路由			
卷积 + BN + leakyrelu	2561 × 1		
上彩样			
Concat	768		
4 ×	卷积 + BN + leakyrelu	2561 × 1	1
	卷积 + BN + leakyrelu	5123 × 3	1
YOLO 卷积 (预测)	$3 \times (5 + \text{类别数})1 \times 1$		126×26
路由			
卷积 + BN + leakyrelu	1281 × 1		
上彩样			
Concat	384		
4 ×	卷积 + BN + leakyrelu	1281 × 1	1
	卷积 + BN + leakyrelu	2563 × 3	1
YOLO 卷积 (预测)	$3 \times (5 + \text{类别数})1 \times 1$		152×52

图 1 YOLOv3 的网络结构

YOLO 卷积层负责输出检测结果, 包括目标的中心位置 xy , 宽高 wh , 置信度, 以及类别, 这种检测分别在三个尺度上进行, 13×13 , 26×26 , 52×52 , 通道数为 3, 也就是每个 box 负责对三个 anchor box 进行回归, 取其中的一个作为最终检测结果, 共对 9 个 anchor box 进行回归, 所以对于一张输入图像, 最后的输出尺寸为 $1 \times (3 \times (13 \times 13 + 26 \times 26 + 52 \times 52)) \times (5 + k) =$

$1 \times 10647 \times (5+k)$, k 代表类别数, 在 COCO 数据集上为 80, VOC 数据集上为 20, 如图 2 所示。

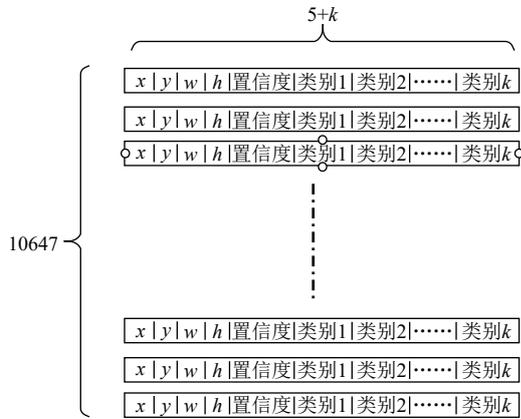


图 2 YOLOv3 网络的输出格式

在 YOLOv1 版本中, x, y, w, h 是直接预测物体实际值, 预测值的微小变化都会被放大到整个图像的范围, 导致坐标波动较大, 预测不准确. YOLOv2 对其进行了改进, 其公式为:

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w e^{t_w}$$

$$Loss = \frac{1}{2} \sum_{i=1}^{10647} \lambda_{obj} * \left((2 - truth_w * truth_h) * \sum_{r \in (x,y,w,h)} (truth_r - predict_r)^2 + \sum_{r=0}^{k-1} ((r == truth_{class}) ? 1 : 0 - predict_{class_r})^2 \right) + (truth_{conf} - predict_{conf})^2$$

其主要分为三大部分, 坐标损失, 置信度损失及分类损失, λ_{obj} 在该 cell 存在物体时为 1, 其他 cell 为 0, 且均采用 SSE 计算, 最终 Loss 采用和的形式而不是平均 Loss, 主要原因为预测的特殊机制, 造成正负样本比巨大, 尤其是置信度损失部分, 以一片包含一个目标为例, 置信度部分的正负样本比可以高达 1:10646, 如果采用平均损失, 会使损失趋近于 0, 网络预测变为全零, 失去预测能力. 并且, 根据作者描述, Lin TY 等提出的, 用于解决正负样本不均衡, 使得网络专注于困难样本计算的 Focal Loss 无法解决这个问题, 会导致 mAP 下降。

2 模型训练与改进

2.1 Anchor 参数设定

PASCAL VOC 数据集是目标分类、检测等常用

$$b_h = p_h e^{t_h}$$

其中, t_x, t_y 为网络预测值, 经过 Sigmoid 运算将其缩放到 0 和 1 之间, t_w, t_h 也为网络预测值, 无需 Sigmoid; c_x, c_y 为 cell 坐标, 也就是距离左上角的偏移量; p_w, p_h 代表该 cell 对应 anchor box 的宽高, 计算出 bounding box 的位置, 如图 3 所示. 通过对 confidence 这一项设定阈值, 过滤掉低分的预测框, 然后对剩下的预测框执行非极大值抑制 (Non Maximum Suppression, NMS) 处理, 得到网络最终的预测结果。

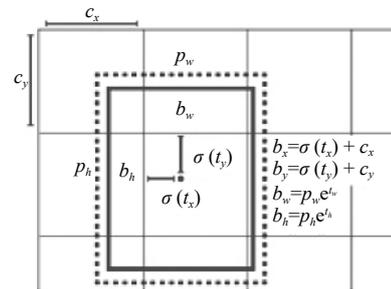


图 3 YOLOv3 的 bounding box 坐标转换公式

1.2 损失函数

对于 YOLOv3 的损失函数, Redmon J 在论文中并没有进行讲解, 本文作者根据对 Darknet, 也就是 Redmon J 实现 YOLOv3 网络的平台的源代码进行解读, 总结出 YOLOv3 网络的损失函数为:

的数据集, 通过模型在 VOC2007 test 部分的表现衡量目标检测模型的性能已经成为常用的验证方法, 在实际训练过程中, 一般使用 VOC2012 的全部样本及 VOC2007 的 train 及 val 部分样本作为训练集, VOC2007 test 部分作为测试集, 本文也是如此. 训练集部分包含图片 16 551 张, 物体 40 058 个, 测试集包含图片 4952 张, 物体 12 032 个. 根据标签数据, 对训练集的物体宽高进行分析, 其分布情况如图 4 所示, 可以发现 VOC 数据集中较小目标占比较大。

作为聚类算法的一种, K-means 具备简洁快速, 易于实现的优点, 应用非常广泛, 其基本思想是以空间中 K 个点作为形心, 将最靠近他们的点进行归类, 然后迭代更新这 K 个点的值, 直到 K 个值不再变化或达到最大迭代次数. 本文使用 K-means 算法对 VOC 数据集进

行分析,结果如图5,纵坐标表示平均畸变程度,越小说明类内距离越小,根据肘部法则选择K值为9,并确定相应的 anchor box.

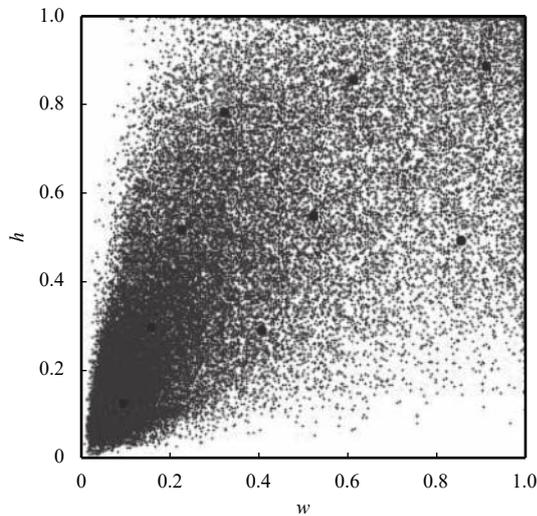


图4 VOC 物体宽高分布情况,大点为聚类结果

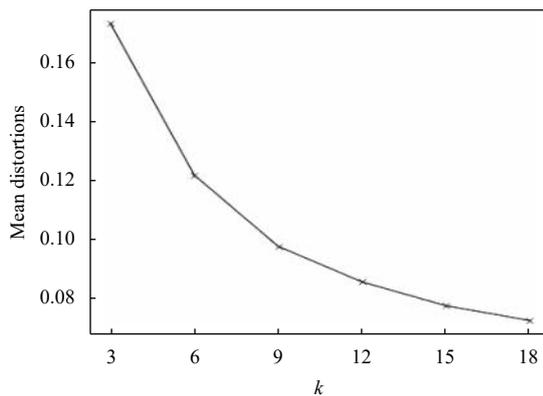


图5 K-means 聚类分析结果

2.2 损失函数分析

作为神经网络对误检测样本进行惩罚的根据,损失函数可以在很大程度上影响模型收敛的效果,如何设计更合适的损失函数以获得更好地预测效果也成为模型优化的重要方向,学界对此也进行了很多研究^[12,13].在YOLOv3网络的最终输出中, x, y , 物体置信度以及类别置信度部分均经过 Sigmoid 函数激活,然后采用 SSE 计算最终损失,从 Sigmoid 函数的导数图像(图6)可以看到,当神经网络的输出较大时, $\sigma'(x)$ 会变得非常小,此时使用平方误差得到的误差值很小,导致网络收敛很慢,出现误差越大收敛越慢,也就是梯度消失的情况,为了解决这个问题,当真实值只能取 0 或 1 时,一

个常用的方法是采用交叉熵 (cross-entropy) 损失函数,其形式为:

$$Loss = -\frac{1}{n} \sum_x (y * \ln(a) + (1 - y) * \ln(1 - a))$$

a 代表神经网络经过 Sigmoid 以后的输出值,也就是 $a = \sigma(\omega x + b)$, 令 $z = \omega x + b$, 可以计算出交叉熵函数的导数.

$$\frac{\partial L}{\partial \omega} = \frac{1}{n} \sum_x x(\sigma(z) - y), \quad \frac{\partial L}{\partial b} = \frac{1}{n} \sum_x (\sigma(z) - y),$$

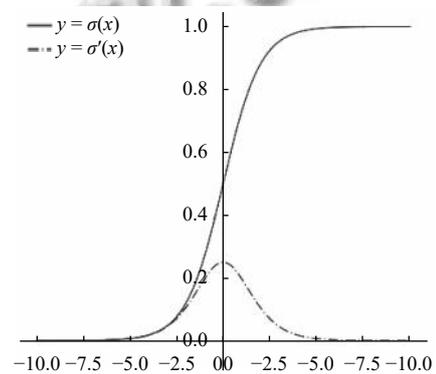


图6 Sigmoid 函数及其导数

这样就消除了 $\sigma'(z)$ 的影响,是目标检测网络常用的损失函数.但是,在YOLOv3网络中,如果直接采用交叉熵损失,由于输出值高达 $10647 \times (3+k)$ 个 ($pred_w, pred_h$, 没有采用 Sigmoid),又由于交叉熵在误差较大时梯度也非常大,与平方损失的梯度变化并不相同,难以对不同损失项之间的权重进行调整.另外,巨大的正负样本比也决定了采用求和而不是常见的求平均的方式计算损失值可以更好地体现预测误差.

特别的,对于 $pred_x$ 和 $pred_y$ 的输出,它们的真实值并不是 0 或 1,而是 0 和 1 之间的某个值,也就是说 $pred_x, pred_y$ 并不是离散变量,以 $truth_x = pred_x = 0.7$ 为例,交叉熵损失为 $-0.7 * \ln(0.7) - 0.3 * \ln(0.3) \approx 0.61$,不为 0,所以不能采用交叉熵损失函数.

通过以上分析可知,损失函数的导数形式 $f(x)$ 需要具备以下性质: ① $f(x)$ 的定义域为 $x \in (-1, 1)$, 值域 $f(x) \in (-1, 1)$; ② $f(x)$ 为单调递增函数,且 $f(0) = 0$; ③ $f(x)$ 的图像需要关于坐标原点对称; ④ $f'(x) \neq 0$. 因此,本文设计了一种新的损失函数,称为 tan 方损失 (Tan-Squared Error, TSE),导数形式为 $LossGrad = \tan(t - \sigma(z)) / \tan(1)$, 在YOLOv3中, $t - \sigma(z) \in (-1, 1)$, 由于 t 也是不

定项, 此差值与 $\sigma(z)$ 本身的取值无关, 令 $x = t - \sigma(z)$, 作出函数图像如图7.

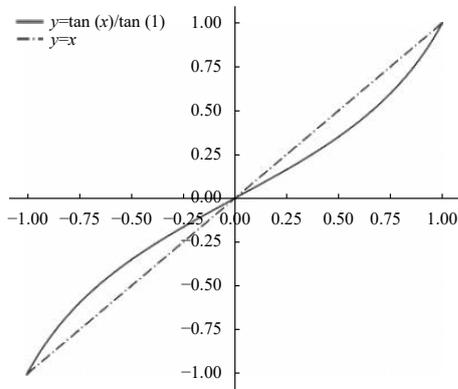


图7 TSE与SSE导数图像

在本网络的训练过程中, 由于极其巨大的正负样本比, 训练开始时会出现网络输出全零的情况, 此时负样本部分, 也就是真实值为0的部分, 其误差逼近于0, 而真实值为1的部分, 误差则非常的大, 对于 $pred_x$ 和 $pred_y$ 来说, 差值更是为(-1, 1), 而不是常见的(0, 1), 作者采用平方损失进行计算, 并且是定义了梯度值的计算方法, 也就是直接使用公式 $grad = t_x - pred_x$ 进行梯度计算, 这也正是平方损失函数的导数形式, 遵循同样的思路, 本文定义了TSE导数形式的计算公式, 通过TSE与SSE的图像(图7)可以直观地看出, TSE与SSE导数的值域均为(-1, 1), 并且TSE导数的绝对值要小于SSE, 也就是说对于较大的误差, 可以将它进行适当地缩小, 这样当梯度传播到Sigmoid函数时, 可以使 $\frac{\partial \sigma}{\partial w}$, $\frac{\partial \sigma}{\partial b}$ 小幅度地增大, 加快初始时的收敛速度, 减小梯度消失的影响; 同样的, 对于误差已经逼近0的情况, 由于TSE要略微小于SSE, 可以使输出层权重的调整幅度更小, 使模型得到更好收敛. 这种调整也应该控制在一个较小的幅度, 防止对输出层的权重作出过大的调整, 导致模型不收敛或者发散的情形. 对两函数求导, 得到他们的导数图像如图8.

可以看到, TSE导数的值域约为(-2.2, 2.2), 即使误差极大导数也在有限范围内, 并且最小值约为0.64, 不会出现梯度为0从而导致梯度消失. 不同于SSE的常数梯度值, TSE的梯度会随着误差的变化而变化, 具有误差大时梯度大, 误差小时梯度小的性质, 既能在误差较大时加大权重的调整, 也在误差减小以后, 以更小的变化率调整权重, 使网络模型可以收敛得更好, 这是

对于连续变量 $pred_x$ 和 $pred_y$ TSE所具有的优势.

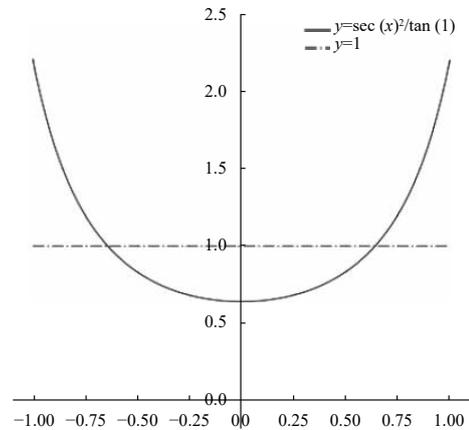


图8 TSE与SSE导数的梯度变化

2.3 调整幅度验证

对预测损失值的大小进行控制可以增强反向传播时卷积层权重调整的效果, TSE控制梯度变化的幅度是否合适也需要验证. 将TSE与梯度调整幅度更大的损失函数进行对比, 其导数形式为:

$$LossGrad = (\sec^2(x) * \tan(x)) / (\sec^2(1) * \tan(1))$$

从它与TSE的对比图像如图9.

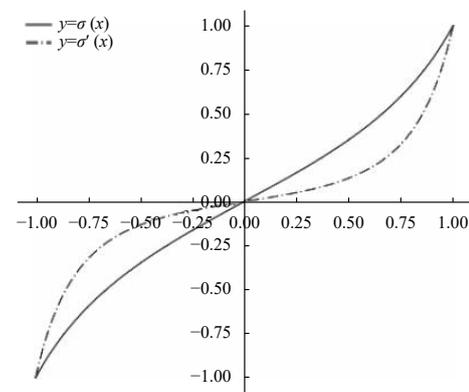


图9 TSE与另一函数对比

从图9可以看出: 该函数在训练的初期可以对损失进行更大幅度的调整, 理论上讲可以更大程度地减少梯度消失的情况, 使得预测层梯度保持在一个较大的数值. 在实际的实验过程中, 训练的初始阶段Loss下降得非常快, 正样本的置信度上升速度也加快了很多, 但是由于误差靠近0时梯度的过度减小, 负样本的置信度未能得到有效下降, 训练后期梯度消失的现象也导致正负样本没有被很好地区分开, 模型收敛

效果变差, 最终的 mAP 很低; 而且由于采用较大的学习率, 训练中也出现了预测值变为 Nan 的情况, 说明过大的调整幅度也可能导致梯度发散。

3 实验结果

在 Pascal VOC 数据集上进行实验, 使用两块 1080TI 的显卡, 显存为 2×12 GB, CPU 为 Intel Xeon E5-2620 V4@2.10 GHz, 软件环境为 cuda9.0 + cudnn7, 训练集为 VOC07+12, 测试集为 07 test-dev, 由于作者并未公布相关的 mAP 数据, 本文直接使用作者提供的训练配置文件进行训练, 总计训练了 50 000 个 batch, 除了更换损失函数和修改 anchor box 外不作其他改动, 结果如表 1 所示。

表 1 VOC 数据集训练结果

方法	训练集	测试集	mAP
SSE	07+12	07test	0.753
TSE	07+12	07test	0.764

由于本实验的主要目的是验证 TSE 的有效性, 作者并没有对训练时的超参数进行仔细调参, 不保证该 mAP 一定为 YOLOv3 网络模型在 VOC 数据集上可以达到的最高值。通过最终的 mAP 值可以表明, 只是改用 TSE 就可以得到 1 个百分点左右的提升, TSE 可以使网络获得更好地收敛效果。

为了验证 TSE 相比于 SSE 在收敛速度方面所获得的优势, 采取每 10 000 个 batch 保存一次模型的方式, 收集了 5 个阶段的 mAP 数据, 如图 9 所示, 可以看出, 在训练 10 000 个 batch 时, TSE 就可以比 SSE 取得更高的 mAP 值, 并且在 30 000 个 batch 时 mAP 就达到了 SSE 训练 40 000 个 batch 的水平, 说明使用 TSE 可以使网络收敛得更加快速。当训练 40 000 个 batch 以后, 学习率被减少为原来的 1/10, 模型进一步收敛, mAP 获得了比较大的提升, 并且受益于 TSE 的函数性质, 模型获得了 1 个百分点左右的 mAP 提升。

如图 10, 以 Other 项代指损失函数导数 $LossGrad = (sec^2(x) * \tan(x)) / (sec^2(1) * \tan(1))$, 实验结果表明, 对损失值进行较大幅度调整, 虽然收敛更加快速, 10 000 个 batch 的模型准确率要略高于其他方法, 但是最终的收敛效果要远差于 TSE。从训练过程中正负样本置信度的平均值来看, 采用 TSE 会使得正负样本的置信度均有小幅度的上升, 但是这种调整在不影

响正负样本区分的基础上, 增强了正样本的置信度, 从而获得了良好收敛效果。与 Other 项的对比结果说明了 TSE 能够以一个合适的幅度对梯度进行调整, 可以实现损失减少与置信度增加, 具有良好的函数性质。

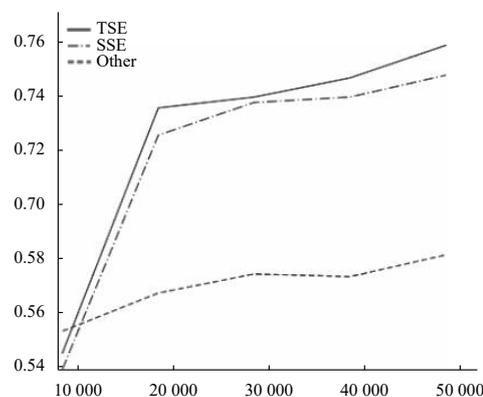


图 10 三种方法训练的 mAP

4 总结与展望

4.1 本文工作总结

本文分析了 YOLOv3 模型的结构, 并且对它实际的训练应用过程做出了有益尝试。通过对原版损失函数特点的分析, 提出了一种新的损失函数 TSE, 在 VOC 数据集上的实验结果表明: ① 与 SSE 相比, TSE 适用于连续变量的情况, 减少了梯度消失的情况, mAP 稳步提升; ② 模型的收敛速度得到提升, 能够使训练所需轮数减少, 节省训练所需时间; ③ 对比实验表明 TSE 进行梯度调整的幅度比较合适, 可以在保证正负样本区分度的基础上有效增强正样本的置信度。基于改进损失函数的网络模型取得 1 个百分点左右的 mAP 提升, 获得了更好的检测效果。

目前, YOLO 系列网络由于其检测快速的优点, 已经在实际生产中获得了大量应用。如郑志强等将 YOLO 网络应用于遥感图像的飞机识别^[14], 王福建等将 YOLO 网络应用于车辆信息的快速检测^[15], 蔡成涛等基于 YOLO 网络实现机场跑道目标的快速检测^[16], 都取得了非常好的效果, 本文提出的方法适用面广, 通用性强, 可以很好地应用于现有的 YOLO 检测系统中, 获得更为准确的检测效果, 使系统性能获得进一步增强。

4.2 未来工作展望

本文从损失函数的角度, 提出了现有网络的改进方法。在下一步的研究工作中, 将会在更多其他的网络模型中, 采用更大规模的数据集如 MSCOCO^[17]等, 用

以验证 TSE 的通用性与鲁棒性; 同时也对连续变量的梯度控制问题, 寻求更好的解决方案; 另外, 针对于网

络结构本身的改进, 如何设计出性能更强的网络来提取图像特征, 也需要更加细致地研究。

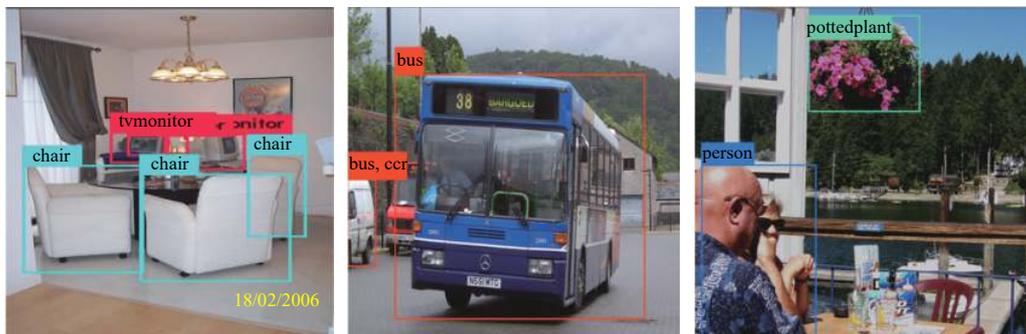


图 11 检测结果示例

参考文献

- Dalal N, Triggs B. Histograms of oriented gradients for human detection. Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, CA, USA. 2005. 886–893.
- Felzenszwalb P, McAllester D, Ramanan D. A discriminatively trained, multiscale, deformable part model. Proceedings of 2008 IEEE Conference on Computer Vision and Pattern Recognition. Anchorage, AK, USA. 2008. 1–8.
- Girshick R, Donahue J, Darrell T, et al. Region-based convolutional networks for accurate object detection and segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(1): 142–158. [doi: 10.1109/TPAMI.2015.2437384]
- Girshick R. Fast R-CNN. Proceedings of 2015 IEEE International Conference on Computer Vision. Santiago, Chile. 2015. 1440–1448.
- Ren SQ, He KM, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montréal, Canada. 2015. 91–99.
- Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 779–788.
- Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector. Proceedings of the 14th European Conference on Computer Vision. Amsterdam, the Netherlands. 2016. 21–37.
- Redmon J, Farhadi A. Yolo9000: Better, faster, stronger. Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA. 2017. 6517–6525.
- Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv preprint arXiv: 1804.02767, 2018.
- Lin TY, Goyal P, Girshick R, et al. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018. [doi: 10.1109/TPAMI.2018.2858826]
- Everingham M, Eslami SMA, Van Gool L, et al. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 2015, 111(1): 98–136. [doi: 10.1007/s11263-014-0733-5]
- 曾逸琪, 关胜晓. 一种基于隔离损失函数的人脸表情识别方法. 信息技术与网络安全, 2018, 37(6): 80–84.
- 龙鑫, 苏寒松, 刘高华, 等. 一种基于角度距离损失函数和卷积神经网络的人脸识别算法. 激光与光电子学进展, 2018, 55(12): 121505.
- 郑志强, 刘妍妍, 潘长城, 等. 改进 YOLO V3 遥感图像飞机识别应用. 电光与控制, 1–6. <http://kns.cnki.net/kcms/detail/41.1227.TN.20180823.1037.002.html>. [2018-08-25].
- 王福建, 张俊, 卢国权, 等. 基于 YOLO 的车辆信息检测和跟踪系统. 工业控制计算机, 2018, 31(7): 89–91. [doi: 10.3969/j.issn.1001-182X.2018.07.039]
- 蔡成涛, 吴科君, 严勇杰. 基于优化 YOLO 方法机场跑道目标检测. 指挥信息系统与技术, 2018, 9(3): 37–41.
- Chen XL, Fang H, Lin TY, et al. Microsoft COCO captions: Data collection and evaluation server. arXiv preprint arXiv: 1504.00325, 2015.