

tricks	应用	图神经网络
BN	部署	1.怎么表达特征 图片二维结构,可以定义CNN 文本语言一维结构,可以定义滑窗 图结构非欧几里得结构(图的结构可以任意变化,节点也可以以任意顺序标记) 通过图结构里的遍历,生成节点的序列,来创造一维度结构然后再使用Word2Vec的思想Node2Vec 周图节点来编码中心节点,相当于通过训练一个虚拟网络,把每个节点周围的结构信息储存在了这个虚拟网络里,而输入周图节点后这个网络输出的向量,正是这个中心节点的Embedding GNN
	解码	

评判标准	各种卷积	卡尔曼滤波	基本backbone			
<table><tr><td>TP(正确报)</td><td>FN(漏报)</td></tr><tr><td>FP(误报)</td><td>TN(正确不报)</td></tr></table> <div>准确率 (TP+TN)/(TP+FN+FP+TN) 精确率 P = TP/(TP+FP) 召回率 R = TP/(TP+FN) Fscore (P*R*2)/(P+R)</div> <div><div>Precision</div><div>Recall</div><div>IoU</div><div>Area of Overlap</div><div>Area of Union</div><div>Intersection</div></div> <div>召回率低就是漏报多,精确率低就是误报多 mAP</div> <div>AP:平均精确度(P-R曲线围起来面积是AP) mAP:是对多个验证集个体求平均AP值 目标检测,每一类都可以根据R和P绘制P-R曲线,AP是该曲线下的面积,mAP就是所有类AP的平均值</div> <div><div>Average Precision (AP):</div><div>AP at IoU=0.50 (primary challenge metric)</div><div>AP at IoU=0.50 (PASCAL VOC metric)</div><div>AP at IoU=0.75 (precision metric)</div><div>AP Across Scales:</div><div>AP for small objects: area < 32²</div><div>AP for medium objects: 32² < area < 96²</div><div>AP for large objects: area > 96²</div><div>Average Recall (AR):</div><div>AR given 1 detection per image</div><div>AR given 10 detections per image</div><div>AR given 100 detections per image</div><div>AR Across Scales:</div><div>AR for small objects: area < 32²</div><div>AR for medium objects: 32² < area < 96²</div><div>AR for large objects: area > 96²</div></div>	TP(正确报)	FN(漏报)	FP(误报)	TN(正确不报)	<div><div>空洞卷积</div><div>在标准的CNN中注入空洞,以此来增大感受野 和标准的CNN比较,空洞卷积会多一个超参数→空洞rate(指的是kernel间隔的数量) 提出要解决的问题: 语义分割要上采样,小的图片在pooling的过程之后没法上采样了(信息无法重建) 此处有个问题为啥语义分割要上采样重建信息(原始图片尺寸空间的预测) 不通过pooling也能看到较大的感受野 缺点: 损失信息的连续度 大目标有利,小目标有弊 设计: 锯齿状</div><div><div>全卷积</div><div>传统的CNN网络中,最后的卷积层后会连接上若干个全连接层,将特征图映射成一个固定长度的向量,FCN把CNN最后的全连接层换成卷积层,输出的是一张已经标记好的图不是一个概率值 FCN接受任意尺寸的输入图像,用反卷积层对最后一个卷积层的特征图进行上采样</div><div><div>反卷积(转置卷积)</div><div>右图所示,标准的CNN会将4*4的x经过卷积成2*2的y 矩阵表达就是y=Cx→其中C是4*16的矩阵 C的转置(16*4)*(4*1)→(16*1)←反卷积的操作 上采样包括三种方式 resize双线性插值 转置卷积 反池化</div><div><div>可分离卷积(mobilenet)</div><div>空间可分离卷积 空间可分离卷积能单独将卷积核划分为两个较小的卷积核 可在空间上分离的卷积是用于边缘检测的sobel卷积核 主要问题是并非所有卷积核都可以“分离”成两个较小的卷积核 深度可分离卷积 标准卷积12*12*3→5*5*3*256→8*8*256 深度卷积:12*12*3→5*5*1*3→8*8*3 逐点卷积:8*8*3→1*1*3*256→8*8*256 参数小,效果法将次数小,训练更快 缺点可能参数太少,就是效果不好</div><div><div>多尺度非线性卷积</div><div></div></div></div></div></div></div>	<div>Resnet</div>
TP(正确报)	FN(漏报)					
FP(误报)	TN(正确不报)					

损失函数	激活函数
<div><div>Focal-loss</div><div>在基本的交叉熵上给正负样本加权重,设定a的值来控制正负样本对总loss的共享权重,a取较小的值降低负样本(多的那类样本)的权重 没法控制容易分类和难分类样本的权重(通过减少易分类样本的权重,使得模型在训练时更专注于难分类的样本)</div><div><div>图</div><div><div>$CE(p_i, y) = -\log(p_i)$ $FL(p_i) = -(1 - p_i)^{\gamma} \log(p_i)$</div><div><div>well-classified misclassified</div></div></div></div><div><div>IOU很好的特性就是尺度不变性,两个框没有相交IoU=0(loss=0,没有梯度回传,无法进行学习训练),同时不能反映两者的距离大小 GIOU关注重叠区域,还关注其他的非重叠区域,能更反映重合度,两个框的最小包围区域面积Ac DioU中b,b^d分别是预测框和真实框的中心点,p是两个中心点间的欧式距离,c是同时包含预测框和真实框的最小包围区域对角线距离</div><div><div>$DIoU = IoU - \frac{p^2(b, b^d)}{c^2}$</div></div></div></div>	

经验总结	FAQ
<div>神经网络浅层信息更多的是细节特征(对小目标检测更有效),高层更多的是语义特征(对大目标检测更有利)</div> <div>提出问题,找方法解决问题 1.Anchor超参数多,参数影响很大,只有一部分框被看过,召回率低 解决方案-使用anchor free的办法,用做语义分割的思想基于pixel来做回归 2.基于pixel做回归的重叠问题 FPN层约定不同的回归值在不同的层去看 3.发现效果还是不太OK,实验出和多个质量不高的问题 使用center-ness多一个损失</div> <div>torch/csrc/stub.cpp链接libshm.so、libtorch_python.so、libcaffe2_gpu.so生成了_c.cpython-37m-x86_64-linux-gnu.so库 import torch会找torch package中的__init__.py文件from torch._C import *,其中torch._C就是_C.cpython-37m-x86_64-linux-gnu.so 默认的引擎都是Cython(CPython的C/C++扩展是一个共享库) torch/csrc/stub.cpp中一定实现了PyInit_C这个函数</div>	<div>1.语义分割的tricks是什么 2.为啥语义分割要上采样重建信息 3.棋盘效应 4.GNN网络 5.卡尔曼滤波 6.ResNet</div>