深度学习

模型优化

- 剪枝：将神经网络某此元余连接部的权重置为0，使得模型变更加具有稀疏化，从而提升模型性能。
  - 结构化剪枝：剪枝对块状的权重或卷神经元结构进行剪枝
  - 非结构化剪枝：对单元进行剪枝
  - 结构化剪枝：对filter/channel/layer进行剪枝

- 优化：
  1.在内存占用部署方面，大幅减缩型后小型集模型实现了更高的精度。
  2.经过剪枝之后网络能型变长了，同体积容纳能模型。
  3.更准有效的情况下，剪枝能比较有效的模型压缩网络。
  4.优化后模型可以使件能能实现在在向实施。

- soft loss 使用KL loss：KL散度：衡量两个概率分布的差异，KL散度越小，两个概率分布差异越小。KL散度=交叉熵-信息熵(常数)
- 应用场景：
  模型压缩
  优化训练，防止过拟合(强自信过到0和1)
  无源大，无监督数据集的数据增强
  少样本，零样本学习
  迁移学习和知识蒸馏
- 知识蒸馏：
  首先在一个已经训练好的网络(Teacher模型)，把原来数据input的数据网络，教师网络会给每个数据标签一个温度为T的软概率softlabels)
  同时把数据input输导至网络(student模)，也给学生网络一个温度T的softmax层的hardlabels，把softlabels和hard predictions的一个损失函数L(distillation loss)和L(hard loss)比例2两个损失函数相加，解释既要学生在模拟老师的预测结果，学生网络的这一个一个温度的softmax层和hard predictions的hardlabel网络一个损失函数L(student loss)的hardlabels，让他们两个损失函数达到最小。
  所以这个学生网络既要在温度为T的情况下能模拟老师网络的预测结果尽可能接近，又要在温度为1的情况下能模拟出真实数据尽可能接近。

激活函数

- 优化：输出值在0和之间，适用于预测二分类
- 缺点：输出均值不是0，输入较大时，导致小、容易梯度消失

Sigmoid

- 优点：输出均值为0，导致比sigmoid更好
- 缺点：存在梯度弥散

Tanh

输出为容和为的概率分布，一般做多分类问题输出

Softmax

- 优点：解决梯度弥散问题，减少神经元巧求计算，计算快
- 缺点：输出均值不为0，存在DeadReLU问题，可能会有梯度爆炸

Relu

- 优点：输入小于0时属于部分初期度，避免DeadReLU问题
- 缺点：小参数需要指定

Lakey Relu

- 优点：输出均值为0，导致比Lsigmoid更好
- 缺点：存在梯度弥散

Elu

- 优点：α是可学习的超参数，可以根据网络训练 → Prelu

Relu变种

激活函数旨在为网络引入非线性，使神经网络有更好的拟合能力，可以表示任意格式的输入数据

- 优点：各层输出会向标准正态分布靠拢，避免梯度消失和爆炸

SELU

- 解释最大输出值为1，可以避免数值过大 → Relu6

Relu6

- 优点：为Sigmoid弱线性近似，提交计算效率
- 缺点：梯度硬饱和，输出均值不为0

Hard-Sigmoid

- 优点：深层模型优于Relu
- 缺点：计算量

Swish

- YOLOv4中使用的激活函数，说函数在超参数设置上比Swish少0.4%与Mish少1.8TFLOPS使用提高。

Mish

其他变种

数据增强策略

- 几何变换（随机旋转、翻转、裁剪、变形、缩放）
- 颜色变换（亮度、对比度、色调、饱和度、噪声）

- 随机擦除(random erasing)：随机遮挡一个矩形区域，用均值像素填充
- CutOut：随机遮挡多个正方形区域，用0填充。
- Hide-And-Seek：将为图像遮挡图片的不同区域，使得模型不能得知区域特征，学习更加全面有能源的特征表示。
- GridMask：图片网格进行遮挡

- 遮挡增强（随机擦除、Cutout、Hide-And-Seek、GridMask）

- 构每部分单个神经元，加强全局注意力，适用于全连接
  - Dropout
- 对特征图上的随机连续区域置零，加强空间注意力，更多的子卷积网络
  - DropBlock

- 正则化增强（Dropout、DropBlock）

- Mixup：将随机的两张样本按比例混合，分类的结果按比例分配
- CutMix：将一部分区域cutout掉，随机填充训练集中的其他数据的区域像素值
- Mosaic：对四张图片进行拼接
- 相当于cutmix的优化版，增加数据多样性，使模型学习能异特征的能力，使模型可以拟合两种大小的目标，无需特别选择不来减少过拟合

多图增强（Mixup、CutMix、Mosaic）

学习率调整策略

- 每隔固定步长时，减少为原来的1/γ
  - 步长衰减：StepLR
- 在指定步骤时，衰减学习率为1/γ
  - 多步长衰减：MultiStepLR
- 每一步乘gamma
  - 指数衰减：ExponentialLR
- 按照余弦周期变换学习率
  - 余弦退火衰减：CosineAnnealingLR

优化器

- 批量梯度下降：优化算法的基础，通过损失梯度反向传动更新模型参数的值，使损失不断减少，每次更新都计算整个数据集的loss
- 梯度下降法
  - 更高阶于过程训练
  - 优点：可以探索新方向，找到更好的极值点
  - 缺点：更新频繁，梯度下降收敛
  - 随机梯度下降的SGD：每次随机选择一对样本，模型的损失函数会有较大变化
- 小批量梯度下降：以上两种方法的折中
- 动量法
  - 梯度可以为历史同向，步幅增大，加快下降速度，方向相反时，步幅减小
- 自适应梯度Adagrad
  - 自动适应参数的学习率，并在训练过程中对不同特征有较大的学习率，对于频繁出现的特征有较小的学习率。
  - 缺点：由于更新梯度平方的累加，学习率在训练后期会导致会不断减小，可能导致学习率过小，使得模型在后期训练不能以得到收敛。
- 自适应梯度RMSProp
  - 使用momentum和Adagrad的改进版，通过引入指数平方滑行指数加权来对学习率自适应地调整学习率，并解决学习率过小，不能收敛问题。
  - 优点：可理解Adagrad学习率下降较快的问题，并且引入动力项，可以减少震动，适合处理非平稳目标。对于RNN效果很好
  - 缺点：依赖全局学习率
- 动量+自适应Adam
  - 优化：
    1. 对初始偏差进行修正，使bias function可以接近初始状态
    2. 参数的更新不受梯度的伸缩变换影响
    3. 更新步长长期稳定小化天。只和alpha、beta1、beta2有关系，并且在迭代末期步的理论上接
    4. 更新的步长能够被限制在大致的范围内(初始学习率)
    5. 能自动调动偏置修正，能天然地实现步长从较大的初期到较小的
    6. 很适合应用于大规模的数据及参数的场景，不稳定的目标函数，梯度稀疏或梯度存在较大噪声的问题
  - Adam
  - 增加学习率上限
  - Adam的改进版本，通过将梯度衰减或添加到损失函数中，AdamW可以对模型的超参数和权重衰减的影响，避免权重更新过于强烈。
  - AdamW
  - 提升正则化效果：AdamW可以更好地控制正则化，提升模型的泛化能力。
  - 适用于大模型训练：AdamW在大规模深度学习训练中表现良好，有助于训练更深度复杂模型。

归一化

- 作用：
  1. 归一化的神经元能使一些不必要的数值消除。
  2. 为了防止梯度消失或爆炸。
  3. 统一量纲。
  4. 避免神经元饱和。

- 常用方法

卷积神经网络

- 全连接
  - 优点：
    - 表达能力强，拟合广泛
    - 容易训练
  - 缺点：
    - 参数量过多容易过拟合
    - 缺乏数据的空间信息
- 卷积
  - 卷积作用：
    - 浅层卷积：提取边缘特征
    - 中层卷积：提取局部特征
    - 深层卷积：提取全局特征

- 不同卷积核作用
  - 卷积核大小：
    - 空立了感的感受野，常见的尺寸有1、3、7等尺寸为的卷积核用于通道间的信息融合
  - 卷积参数(卷积核大小、步长、填充、膨胀)：
    - 步长(stride)：
      - 更换卷积层的参数步幅为，步长越大，卷积核与原一图像重叠少，那现的特征越稀疏；步长越小，重叠越多，提取到的特征更全局。
    - 填充(padding)：
      - 控制输出特征图的尺寸
    - 膨胀(dilation)：
      - 保持输出特征图的空间大小
      - 扩大卷积核的视野
      - 不可能保留边缘信息，减少信息损失
    - 分组(group)：
      - 将输入特征图的通道分为g个组，每个组内独立卷积，最后将结果拼接
      - 减少参数量，计算速度快，可以学习更多样的特征表示
      - 缺点：
        - 特征通道间信息交互减少，特征提取能力故单卷积容
  - 优势：同一卷积核可用于一个通道，这种权值共享会减少参数量
- 池化
  - MaxPool
    - 缺点：信息损失，丢弃精细特征
    - 应用场景：用于在网络浅层或之多卷的卷积层
  - AvgPool
    - 优点：平滑特征，避免信息丢失
    - 缺点：特征对比度下降，缺乏特征选择性
  - AdaptiveAvgPool
    - 生成固定大小的输出，适合在全连接层之前用来统一特征图的大小，减少模型大小，防止过拟合。
  - 优点：减少参数量，提取最明显的特征
  - 缺点：信息损失，丢弃精细特征
  - 应用场景：用于在网络浅层或之多卷的卷积层
- BatchNormal
  - 优点
    - 具有正则化效果，减少过拟合
    - 允许更大的学习率，加速模型收敛
    - 需要使用计算mean/var，增加模型时间
  - 缺点
    - 批次过小时效果差，训练不稳定
    - 某些情况时模型的表达能力
- Dropout
  - 优点
    - 防止过拟合，增强模型泛化能力。
    - 鼓励特征分布，防止特征共适应。
    - 训练时间变长，需要多步才达到收敛。
  - 缺点
    - 在推理时应需要重新四和反向传播。
    - 复杂网络需仔细调参，不能无脑叠加效果。

- 分组卷积
  - 优点：
    - 减少参数量，计算速度快
    - 可以学习更多样的特征表示
  - 缺点：
    - 特征通道间信息交互减少，特征提取能力故单卷积容
  - 特通道卷积(depth wise)：每个输入通道独立卷积，生成一个输出特征图。所以输出的特征图的通道数与输入通道数相同。
  - 逐点卷积(point wise)：使用1x1卷积核将depth wise的输出在通道维度上进行加权组合，生成新的特征图。
  - 深度可分卷积：输入特征图数量 = 分组数量
- 通道混洗：通道混洗的目的是将分组通道的特征，在不增加计算量的情况下，使通道间交换信。
  - 优点：加强不同通道之间的信息流动，提升特征提取能力
  - 缺点：过度混洗可能破坏原有的特征表达结构
- 注意力：在模型优化的目标函数中引入额外的项，通过为模型要学习的目标函数增加惩罚项，从而增强模型的泛化能力。
  - 优点
    - 防止过拟合 提高模型的泛化能力
    - 改善训练的条件数 加速模型收敛
    - 引入先验知识 增强模型鲁棒性
    - 进行特征选择 降低计算开销
    - 提高可解释性 帮助理解模型决策
    - 改善梯度流 稳定训练过程
  - 缺点
    - 选择注意力机制需要调参 否则可能削弱模型效果
    - 设置注意力参数模型结构的复杂度 增加了模型的过拟合风险和计算量
    - 增加之不当集成学习模型的复杂度
    - 仅依赖数据本身进行注意时可能不足
    - 一些注意力机制需要额外参数
- 残差结构
  - 残差结构

  - 通过引入残差项（residual block）来解决梯度消失和梯度爆炸等问题，残差块包含了两层卷积，即将输入通道改变的两条路的输出，形成残差连接。
  - 瓶颈结构

  - 瓶颈结构用于减少网络中的参数数量和计算量。同时提高网络的表示能力。1x1的卷积层起到了降维和升维的作用，3x3的卷积层起到了特征提取的作用。

经典网络

- 图像分割原理：图像中的物体打上标签，对图像中每一个像素进行分类(背景、人或车等)。从而进行区域划分
- UNet
  - UNet整体采用encoder-decoder结构，即半部分为特征提取，后半部分进上采样，最后把编码特征融合
  - 优势：图片影像表现好的，因为也能得到和高级语义特征图重新，所以高级的特征(with connection)和低级特征(没有connection)更好至上相连
- U2Net
  - U2 - Net 的架构是一个两级嵌套最初 U 结构。该设计具有以下优点：
  - 提出残差 U 块 (RSU) 中混合了不同大小的感受野，它能捕获不同的空间尺度的特征的能力
  - 这些 RSU 块中使用了膨胀卷积，它增加了整个架构的深度，而不会显著增加计算成本。
- LeNet-5
  - 用于识别手写数字和机器印刷字符的卷积神经网络
  - 使用一个3层的序列组合：卷积、下采样（池化）、非线性映射（LeNet-5最重要的特性，奠定了目前深度卷积网络的基础）
- AlexNet
  - 首个应用于图像分类的深层卷积神经网络
- VGGNet
  - 深层卷积网络结构，该网络所采用的 3x3 卷积核的感受野是所有并多模型的基础
- GoogLeNet
  - GoogLeNet中最核心的部分是其内部子网络结构Inception，该结构灵感来源于NiN
  - 采用不同大小的卷积核堆叠不同大小的感受野，最后拼接意味着不同尺度特征的融合
- Densenet
  - DenseNet提出了一个激进的密集连接机制：即互相连接所有的层，具体来说就是每个层都会接受其前面所有层的作为额外的输入。
  - 优点：
    - 由于密集连接方式，DenseNet提升了梯度的反向传播，使得网络更容易训练。
    - 参数更小且计算更高效，由于DenseNet提出了concat特征复用的思路，会至了特征重用。
    - 由于特征复用，最后的分类器使用了低维特征。
- MobileNetV2
  - 引入了残差结构和线性瓶颈结构 解决了普通网络ReLU不利精度的问题
  - 使用深度可分离卷积代替普通卷积 大大减少了计算和参数量。
  - 网络简单，适合在算力受限设备

图像分割

- 图像分割就是对图像中的物体打上标签，对图像中每一个像素进行分类(背景、人或车等)。从而进行区域划分

- UNet
  - UNet整体采用encoder-decoder结构，即半部分为特征提取，后半部分进上采样，最后把编码特征融合
  - 优势：图片影像表现好的，因为也能得到和高级语义特征图重新，所以高级的特征(with connection)和低级特征(没有connection)更好至上相连

- U2Net
  - U2 - Net 的架构是一个两级嵌套最初 U 结构。该设计具有以下优点：
  - 提出残差 U 块 (RSU) 中混合了不同大小的感受野，它能捕获不同的空间尺度的特征的能力
  - 这些 RSU 块中使用了膨胀卷积，它增加了整个架构的深度，而不会显著增加计算成本。

目标检测

- 两阶段目标检测
  - 第一个阶段：
    - 生成候选区域（Region Proposal）：通过启发式方法（如选择性搜索）或深度学习模型（如RPN）生成可能包含目标的区域。
    - 特征提取：对每个候选区域，使用卷积神经网络（CNN）提取特征。
    - 目的：减少搜索空间，提高效率，并为后续分类和定位提供高质量的候选区域。
  - 第二个阶段：
    - 分类：对候选区域的特征进行分类，判断其是否包含目标以及属于哪个类别。
    - 定位：对候选区域的边界进行微调，以更精确地定位目标。
    - 目的：利用第一阶段提取的特征，对候选区域进行精细的分类和定位，以提高检测的精度。
- 单阶段目标检测
  - 直接预测：
    - 单阶段目标检测方法直接从输入图像中预测目标的类别和边界框，无需生成候选区域。
    - 这种方法通常更快，因为它省去了生成候选区域的步骤，直接在图像上进行预测。
  - 特征图：
    - 单阶段目标检测方法通常使用卷积神经网络（CNN）来提取图像的特征，并将这些特征用于预测目标的类别和边界框。
    - 特征图通常是卷积神经网络的输出，它们包含了图像的空间信息和语义信息，用于目标的分类和定位。
  - 目的：
    - 通过直接预测，单阶段目标检测方法能够实现更快的检测速度，适合实时应用场景。
    - 这种方法在保证较高精度的同时，尽可能地减少计算资源的消耗，以提高效率。
- 两阶段目标检测
  - 优点：
    - 高精度：两阶段目标检测方法通常能够提供更高的检测精度，因为它们在第一阶段生成了高质量的候选区域，并在第二阶段对这些区域进行精细的分类和定位。
    - 适应性：这种方法能够适应各种复杂场景和不同大小的目标，因为它们能够在第一阶段生成多样化的候选区域，并在第二阶段对这些区域进行精细的处理。
  - 缺点：
    - 计算成本：两阶段目标检测方法通常需要较高的计算资源，因为它们需要在两个阶段中进行复杂的计算。
    - 实时性：由于计算成本较高，两阶段目标检测方法在实时应用场景中可能会面临挑战，因为它们可能无法满足实时处理的要求。
- 目标检测指标
  - 交并比（IoU）：
    - 定义：IoU是预测边界框与真实边界框之间的交集与并集的比值，用于评估目标检测的精度。
    - 计算：IoU = 交集面积 / 并集面积
    - 阈值：通常设定一个IoU阈值（如0.5），当预测边界框与真实边界框的IoU大于阈值时，认为预测是正确的。
    - 应用：IoU在目标检测中用于计算精度（如mAP）和评估模型的性能。
  - 平均精度（AP）：
    - 定义：AP是在不同IoU阈值下计算的平均精度，通常是在IoU从0.5到0.95之间的平均值。
    - 计算：AP通过计算不同IoU阈值下的精度-召回曲线（PR曲线）的面积来计算。
    - 应用：AP是评估目标检测模型性能的重要指标，用于比较不同模型的检测精度。
  - 平均精度均值（mAP）：
    - 定义：mAP是所有类别的AP的平均值，用于评估模型在多类别目标检测中的整体性能。
    - 计算：mAP通过计算每个类别的AP，然后取这些AP的平均值来计算。
    - 应用：mAP是评估多类别目标检测模型性能的关键指标，用于比较不同模型的检测精度。
- 非极大值抑制（NMS）
  - 定义：NMS是一种用于目标检测的后处理技术，用于从多个重叠的候选边界框中选择最佳的边界框，以减少冗余和提高检测精度。
  - 步骤：
    1. 按置信度排序：将所有候选边界框按照其置信度（或分类得分）从高到低排序。
    2. 选择最高置信度的边界框：选择置信度最高的边界框作为初始候选。
    3. 计算IoU：计算剩余边界框与当前选择的边界框的IoU。
    4. 移除重叠边界框：移除与当前选择的边界框IoU大于阈值的边界框。
    5. 重复：重复步骤2和3，直到所有边界框被处理完毕。
  - 目的：
    - 减少冗余：通过移除重叠的边界框，减少冗余的检测结果，提高检测的效率。
    - 提高精度：通过选择置信度最高的边界框，提高检测的精度。
  - 应用：NMS是目标检测模型中常用的后处理技术，用于优化检测结果，提高模型的实用性。

Loss计算

- 交叉熵损失函数
  - 二类别：
    - 二分类交叉熵损失函数（Binary Cross Entropy Loss）：用于二分类问题，计算预测概率与真实标签之间的差异。
    - 公式：$L = -[y \log(p) + (1-y)\log(1-p)]$
    - 应用：广泛用于二分类问题，如垃圾邮件检测、疾病诊断等。
  - 多类别：
    - 多类交叉熵损失函数（Multi-class Cross Entropy Loss）：用于多分类问题，计算预测概率与真实标签之间的差异。
    - 公式：$L = -\sum_{i=1}^{C} y_i \log(p_i)$
    - 应用：广泛用于多分类问题，如图像分类、文本分类等。
  - 实现：
    - 二类别：torch.nn.BCELoss()（要求输入已经经过sigmoid激活）；torch.nn.BCEWithLogitsLoss()（自带激活函数）
    - 多类别：torch.nn.CrossEntropyLoss()（自带激活函数）；torch.nn.NLLLoss()（不带激活函数）
- 焦点损失（Focal Loss）
  - 定义：焦点损失是一种用于解决类别不平衡问题的损失函数，通过调整交叉熵损失的权重，使模型更关注难分类的样本。
  - 公式：$FL(p_t) = -\alpha_t (1-p_t)^{\gamma} \log(p_t)$
  - 参数：
    - α：类别权重，用于调整不同类别的权重，通常设为类别频率的倒数。
    - γ：聚焦参数，用于调整难分类样本的权重，通常设为2。
  - 作用：
    - 减少易分类样本的权重：通过降低易分类样本的权重，减少模型对这些样本的关注，从而提高模型对难分类样本的关注。
    - 提高模型对难分类样本的关注：通过增加难分类样本的权重，提高模型对这些样本的关注，从而提高模型的整体性能。
  - 应用：
    - 目标检测：在目标检测任务中，Focal Loss常用于解决类别不平衡问题，特别是在背景类别占主导的情况下。
    - 图像分类：在图像分类任务中，Focal Loss也可以用于解决类别不平衡问题，特别是在某些类别样本较少的情况下。
  - 实现：
    - 二类别：torch.nn.BCEWithLogitsLoss()（自带激活函数）；torch.nn.BCELoss()（不带激活函数）
    - 多类别：torch.nn.CrossEntropyLoss()（自带激活函数）；torch.nn.NLLLoss()（不带激活函数）
- 多类别焦点损失（Multi-Class Focal Loss）
  - 定义：多类别焦点损失是焦点损失在多类别分类问题中的扩展，用于解决多类别分类问题中的类别不平衡问题。
  - 公式：$FL(p_t) = -\alpha_t (1-p_t)^{\gamma} \log(p_t)$
  - 参数：
    - α：类别权重，用于调整不同类别的权重，通常设为类别频率的倒数。
    - γ：聚焦参数，用于调整难分类样本的权重，通常设为2。
  - 作用：
    - 减少易分类样本的权重：通过降低易分类样本的权重，减少模型对这些样本的关注，从而提高模型对难分类样本的关注。
    - 提高模型对难分类样本的关注：通过增加难分类样本的权重，提高模型对这些样本的关注，从而提高模型的整体性能。
  - 应用：
    - 目标检测：在目标检测任务中，多类别焦点损失常用于解决类别不平衡问题，特别是在某些类别样本较少的情况下。
    - 图像分类：在图像分类任务中，多类别焦点损失也可以用于解决类别不平衡问题，特别是在某些类别样本较少的情况下。
  - 实现：
    - 二类别：torch.nn.BCEWithLogitsLoss()（自带激活函数）；torch.nn.BCELoss()（不带激活函数）
    - 多类别：torch.nn.CrossEntropyLoss()（自带激活函数）；torch.nn.NLLLoss()（不带激活函数）
- 均方误差损失函数
  - 定义：均方误差损失函数（Mean Squared Error Loss）是一种用于回归问题的损失函数，通过计算预测值与真实值之间的平方差来衡量模型的性能。
  - 公式：$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
  - 参数：
    - y_i：真实值
    - $\hat{y}_i$：预测值
    - n：样本数量
  - 作用：
    - 衡量预测值与真实值的差异：通过计算预测值与真实值的平方差，来衡量模型的预测精度。
    - 优化模型：通过最小化均方误差，来优化模型的参数，以提高模型的预测精度。
  - 应用：
    - 回归问题：均方误差损失函数常用于解决回归问题，如预测房价、股票价格、温度等。
    - 模型评估：均方误差损失函数也常用于评估模型的性能，以比较不同模型的预测精度。
  - 实现：
    - torch.nn.MSELoss()
- 平滑L1损失（Smooth L1）
  - 定义：平滑L1损失是一种用于回归问题的损失函数，结合了L1和L2损失的优点，以减少对异常值的敏感性。
  - 公式：$L_{smooth} = \begin{cases} 0.5(y - \hat{y})^2 / \beta, & \text{if } |y - \hat{y}| < \beta \\ |y - \hat{y}| - 0.5\beta, & \text{otherwise} \end{cases}$
  - 参数：
    - y：真实值
    - $\hat{y}$：预测值
    - β：阈值，通常设为1.0
  - 作用：
    - 减少对异常值的敏感性：通过结合L1和L2损失的优点，减少对异常值的敏感性，以提高模型的稳定性。
    - 提高模型的稳定性：通过减少对异常值的敏感性，提高模型的稳定性，以减少模型的波动。
  - 应用：
    - 回归问题：平滑L1损失常用于解决回归问题，如预测房价、股票价格、温度等。
    - 模型评估：平滑L1损失也常用于评估模型的性能，以比较不同模型的预测精度。
  - 实现：
    - torch.nn.SmoothL1Loss()
- IOU Loss
  - IOU
    - $IOU = \frac{|A \cap B|}{|A \cup B|}$
    - $IOU Loss = 1 - IOU$
    - 问题：
      - 无法直接优化没有重叠的部分，无法反映两个框如何相交的。

  - GIOU
    - $GIoU = IoU - \frac{|A_c - U|}{|A_c|}$
    - 先计算两个框的最小闭合区域面积Ac，再计算出IoU，再计算闭合区域中不属于两个框的区域占闭合区域的比重，最后用IoU减去这个比重得到GIoU。
    - 优点：
      - 对尺度不敏感
      - 不仅关注重叠区域，还关注其他的非重合区域，能更好地反映两者的重合度。
    - 问题：两个框属于包含关系时，GIOU会退化成IOU，无法区分其相对位置关系。

  - DIOU
    - $L_{DIOU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2}$
    - 其中，b^{gt}，分别代表了预测框和真实框的中心点，且 ρ 代表的计算两个中心点间的欧式距离，c代表的是能够同时包含预测框和真实框的最小闭包区域的对角线距离。
    - 优点：
      - DIOU与GIOU一样仅与目标相不重叠时，仍然可以为边界框提供移动方向。
      - DIOU可以直接最小化两个目标框的距离，因此比GIOU收敛的更快。
      - DIOU在包含两个框水平或者垂直方向上的情况下回归很快，而GIOU几乎退化为IOU。
      - DIoU可以替换非极大值抑制（NMS），在网络中NMS中，使用IoU作为预测结果是否删除的标准，会出现一些遮挡的目标被错误地删除的情况，而DIoU则不会出现这种情况。
  - CIOU
    - 为了使预测框与目标框之间的长宽比尽可能的接近，CIOU在DIOU的基础上增加了长宽比的惩罚项。
    - 其 ρ 用来衡量长宽比的惩罚项，定义为 $\rho = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2$
    - 完整的 CIoU 损失函数定义：
    - $\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}^{gt})}{c^2} + \alpha v$
    - 最后，CIoU的梯度实际上可以看成是 CIoU loss，但这整个 αv 项的梯度，在 w 和 h 上 $[0, 1]$ 的取不了，$w^2 + h^2$ 的值通常很小，会导致梯度爆炸，因此在 $\frac{1}{w^2+h^2}$ 前面加权重，α
  - EIOU
    - 在CIOU的损失基础上将长宽比的惯性子拆开分别计算目标和预测的长和宽，该损失函数包括三个部分：重叠面积、中心距离、宽高损失，原因是以往的CIOU中的方法，但是宽高损失直接使目标与预测的宽度和高度之差最小，使得收敛速度更快。
    - 优点
      - IOU Loss：
        - IOU算法是目标检测中最常用的指标，具有尺度不变性，满足非负性，同一性，交换性，三角等性质。
        - 缺点：
        - 1.如果两个框不相交，不能反映两个框距离的远近，因此没有梯度回传，无法进行学习。
        - 2.无法区分两个相交方式的不同。
      - GIOU Loss：
        - GIOU可看作是IOU的改进版，解决了IOU无法反映两个框相交方式的问题。
        - 缺点：
        - 1.当两个框一个在内部，一个在外部，且两个框的交集和并集一样时，GIOU的值不变，退化为IOU。
        - 2.当两个框相交时，在水平和垂直方向上的两条边的距离一样时，GIOU的值不变，退化为IOU。
      - DIOU Loss：
        - DIOU直接将两个框的中心点间的距离，引入到了损失函数中，能够解决IOU和GIOU遇到的问题。
        - 缺点：
        - 1.当两个框的中心点重合时，DIOU的值不变，退化为IOU。
        - 2.与GIOU一样，在水平和垂直方向上的两条边的距离一样时，DIOU的值不变，退化为IOU。
      - CIOU Loss：
        - CIOU是在DIOU的基础上增加了长宽比的惩罚项，增加了长宽比loss，这样预测框在重叠区域的回归会更加符合真实框。
      - EIOU Loss：
        - EIOU在CIOU的基础上增加了预测宽度和真实宽度的惩罚项，增加了长宽比loss，这样预测框在重叠区域的回归会更加符合真实框。
    - 缺点
      - 1.如果两个框不相交，不能反映两个框距离的远近，因此没有梯度回传，无法进行学习。
      - 2.无法区分两个相交方式的不同。
      - 1.当两个框的重叠区域相同，但两个框的宽高比不同时，CIOU会退化为DIOU。
      - 2.当两个框的宽高比一样时，CIOU会退化为DIOU。
      - 1.当两个框的宽高比一样时，EIOU会退化为DIOU。
      - 2.当两个框的宽高比一样时，EIOU会退化为DIOU。
      - 1.当两个框的宽高比一样时，EIOU会退化为DIOU。
      - 2.当两个框的宽高比一样时，EIOU会退化为DIOU。
      - 无
      - 无
- Center loss
  - 通过同同一类别的样本聚集在类别中心附近，减少类内差异。
  - $\mathcal{L} = \mathcal{L}_S + \lambda\mathcal{L}_C = -\sum_{i=1}^{m}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} + \frac{\lambda}{2}\sum_{i=1}^{m}\|x_i - c_{y_i}\|_2^2$
  - Arc-SoftmaxLoss + Arc-Softmax + NLLLoss
  - softmax 是通过角度分类的，Arc-Softmax 加度了角度间的分界线，从而实现对类内紧凑和类间分离。
- ArcFace Loss
  - $L = \frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{s\cos(\theta_{y_i,i} + m)}}{e^{s\cos(\theta_{y_i,i} + m)} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_{j,i}}}$
  - 一范数归一化之后的欧式距离，等价于余弦相似度

Presented with xmind