YOLO 系列综述-Part 3：

YOLOv4——大神接棒，传奇再续

目录

1. 简介

实时物体检测已经成为众多应用中的一个重要组成部分，横跨自主车辆、机器人、视频监控和增强现实等各个领域。在各种物体检测算法中，YOLO（You Only Look Once）框架因其在速度和准确性方面的显著平衡而脱颖而出，能够快速、可靠地识别图像中的物体。自成立以来，YOLO 系列已经经历了多次迭代，每次都是在以前的版本基础上解决局限性并提高性能（见表格 1）。

表格 1：历代 YOLO 发布时间、主要改动及作者，由笔者同学、四川大学硕士贺宇劼整理。相关文章链接：
https://blog.csdn.net/qq_54478153/article/details/139477271

时间	版本	主要改动	作者
2015	v1		Joseph Redmon & Ali Farhadi
2016	v2	引入Batch Normalization 和锚框(anchor boxes)	Joseph Redmon & Ali Farhadi
2018	v3	更高效的骨干网络、引入特征金字塔和PAN	Joseph Redmon & Ali Farhadi
2020	v4	引入SPP模块、CSP模块、Mish激活函数、CmBN归一化	Alexey Bochkovskiy
2020	v5	引入Focus结构	Ultralytics
2021	x	引入解耦头 (Decoupled Head)、不再预设锚框	旷视
2022	v6	引入RepConv，量化相关内容，简化解耦头	美团
2022	v7	Efficient Layer Aggregation Network (ELAN)模块，MP模块，Rep	台湾中央研究院
	v8	C2f模块	Ultralytics
2024	v9	Generalized Efficient Layer Aggregation Network (GELAN)	台湾中央研究院
2024	v10	整体高效的网络设计、空间-通道解耦下采样	清华

在本系列文章中，笔者将回顾 YOLO 框架的发展历程，从开创性的 YOLOv1 到最新的 YOLOv10，逐一剖析每个版本的核心创新、主要差异和关键改进。本系列不仅会探讨 YOLO 各版本的技术进步，还将重点讨论在追求检测速度与准确性之间的平衡。

作为系列的第三章，本文将深入分析 YOLOv4 模型。继上一篇对 YOLOv2 和 YOLOv3 的全面探讨之后，本章将延续这一研究工作，进一步分析在 YOLO 创始人退出后，第四代 YOLO 模型的发展方向和所面临的挑战。希望通过对比分析，在为读者提供一个清晰的视角，以理解 YOLO 框架的演进及其在物体检测领域的深远影响。

文章攥写时间短，未能仔细校对。可能存在笔误、描述不准确、重要内容缺失等问题。希望大家能指出，笔者会随时修改补充。

注：该系列文章中，所有黑色加粗并配有下划线的关键词，均配有交叉引用。

曹倬瑄

2024/6/25 于惠州

2. YOLOv4：基本概念与架构

2.1. 引言&背景

自 2016 年 YOLOv1 诞生以来，YOLO 系列以其在目标检测领域的革命性速度和准确性，引领了技术革新的潮流。YOLOv1 开创性地将目标检测任务转化为回归问题，赋予每个网格单元预测其区域内物体的职责。继任者 YOLOv2 在此基础上引入批量归一化和锚框技术，显著提升了检测性能。YOLOv3 进一步采用多尺度预测，大幅提升了对小尺寸目标的检测能力。

尽管 YOLOv3 的原作者 Joseph Redmon 因对技术潜在的军事应用和隐私侵犯的担忧而退出了项目，YOLO 系列的研究并未因此止步。在人们普遍认为 YOLO 系列可能终结之际，YOLOv4 的问世再次证明了其创新潜力。尽管研发团队经历了重大变更，YOLOv4 依旧继承了系列的优良传统：在保持高 mAP 的同时，显著降低了计算量，成为学术成果在工业应用中的典范。

总体来看，YOLOv4 并没有像前几代那样提出划时代的创新，而是汇集了近年来目标检测领域的最新技术和成果。通过大量的人工试验，YOLOv4 可以视为一种**人工神经架构搜索（NAS）**，通过实验筛选出一系列新方法，对 YOLOv3 的网络结构、训练策略和数据增强等多个方面进行了全面增强，实现了性能的进一步提升。

2.2. 主要改进点

YOLOv4 相比于 v3，主要改进点如下：

- **跨阶段部分连接（Cross-Stage-Partial-connections, CSP）**：CSP 减少了计算量，同时保持了特征的丰富性。
- **SSP 池化（Spatial Pyramid Pooling）**：SSP 使得 YOLOv4 能够在不同尺度上提取特征，这不仅有助于网络更好地识别不同大小的对象，还有助于提高模型的运行效率。
- **跨小批量归一化（Cross mini-Batch Normalization, CmBN）**：CmBN 改进了模型的归一化策略，有助于提高训练稳定性和性能。
- **自对抗训练（Self-adversarial-training, SAT）**：SAT 作为一种新的数据增强技术，增强了模型对输入扰动的鲁棒性。
- **Mish 激活函数**：YOLOv4 采用了 Mish 激活函数，替代了传统的 ReLU 激活函数，进一步提升了模型的非线性表达能力。

YOLOv4 在 MS COCO 数据集上达到了 43.5% AP 的检测性能，同时保持了高帧率的实时检测能力。但是尽管 YOLOv4 在速度和准确性上取得了显著的成果，但仍存在一些挑战，如小目标检测和类别不平衡问题。

3. YOLOv4 核心技术点分析

3.1. 基本结构

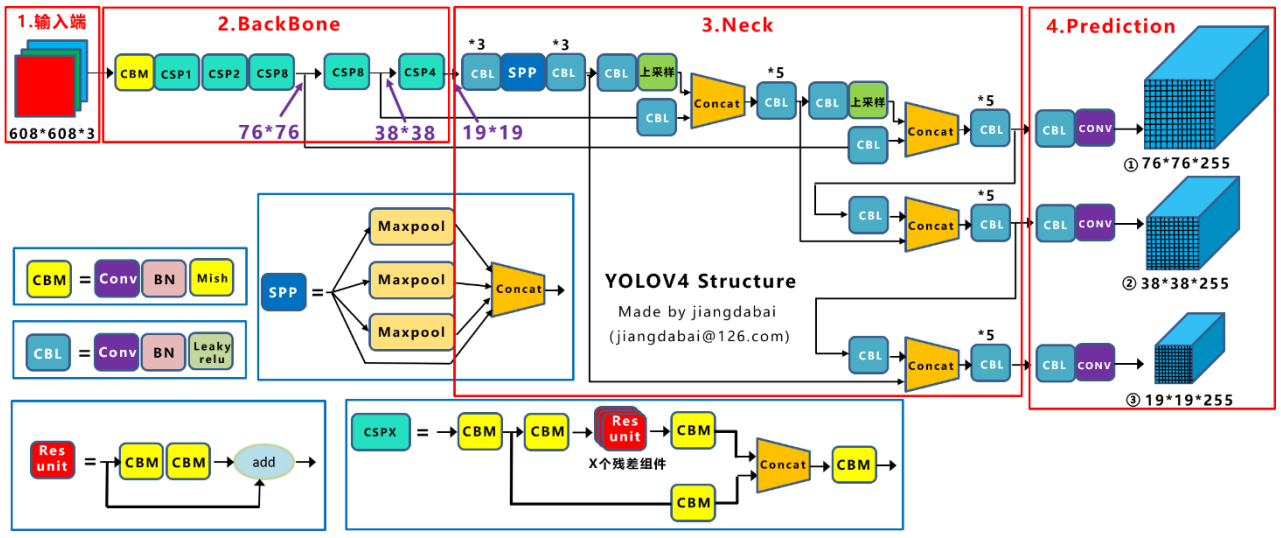
在现代计算机视觉领域，典型的目标检测器由几个关键部分组成。首先，检测器的**主干网络（Backbone）**负责从输入图像中提取特征。对于在 GPU 平台上运行的检测器，可以选择 VGG、ResNet、ResNeXt 或 DenseNet 等在 ImageNet 上预先训练的网络。而在 CPU 平台上，为了优化计算效率，可能更倾向于使用 SqueezeNet、MobileNet 或 ShuffleNet 等轻量级网络。

接下来是检测器的**头部网络（Head）**，它对主干网络提取的特征进行进一步的分析和预测，生成边界框和类别标签。头部网络可以是单阶段的，如 YOLO、SSD 和 RetinaNet，它们直接在单个网络中完成特征提取和目标预测；也可以是两阶段的，如 R-CNN 系列，它们先通过区域提议网络（Region Proposal Network，RPN）生成候选区域，然后再进行精确的目标检测。

近年来，为了进一步提升性能，许多目标检测器在主干和头部之间引入了**颈部网络（Neck）**。颈部网络通过自下而上的路径和自上而下的路径混合和组合图像特征，以实现更丰富的特征表示。常见的颈部网络结构包括特征金字塔网络（FPN）、路径汇聚网络（PAN）、BiFPN 和 NAS-FPN 等。

综上所述，**一个完整的目标检测器通常由特征输入、骨干网络、颈部和头部四部分组成**，它们共同构成了一个强大的系统，用于精确地识别和定位图像中的对象。

下图为 YOLOv4 算法的网络框架示意图：



YOLOv4 的五个基本组件：

- **CBM**：Yolov4 网络结构中的最小组件，由 Conv+Bn+Mish 激活函数三者组成。
- **CBL**：由 Conv+Bn+Leaky_relu 激活函数三者组成。
- **Res unit**：借鉴 Resnet 网络中的残差结构，让网络可以构建的更深。
- **CSPX**：借鉴 CSPNet 网络结构，由三个卷积层和 X 个 Res unint 模块 Concate 组成。
- **SPP**：采用 1×1，5×5，9×9，13×13 的最大池化的方式，进行多尺度融合。

以下是对 YOLOv4 网络结构改进：

- **输入端**：这里指的创新主要是训练时对输入端的改进，主要包括 Mosaic 数据增强、cmBN、SAT 自对抗训练
- **主干网络(Backbone)-主干特征提取网络**：将各种新的方式结合起来，包括：CSPDarknet53、Mish 激活函数、Dropblock
- **颈部网络(Neck)-加强特征提取网络**：目标检测网络在 BackBone 和最后的输出层之间往往会插入一些层，比如 Yolov4 中的 SPP 模块、FPN+PAN 结构
- **头部网络（Head）-用来预测**：输出层的锚框机制和 Yolov3 相同，主要改进的是训练时的损失函数 CIOU_Loss，以及预测框筛选的 nms 变为 DIOU_nms

3.2. 输入端创新

3.2.1. Mosaic Data Augmentation（Mosaic 数据增强）

YOLOv4 在数据增强领域带来了创新，特别是通过引入 Mosaic 增强方法，这一技术通过将四幅训练图像组合成单一图像，显著增加了上下文的多样性。这种先进的方法不仅增强了模型对多变场景的适应性，还有效减少了对大规模数据集的依赖。

在深入 Mosaic 方法之前，我们先回顾一下传统的数据增强技术：Mixup、Cutout 和 CutMix：

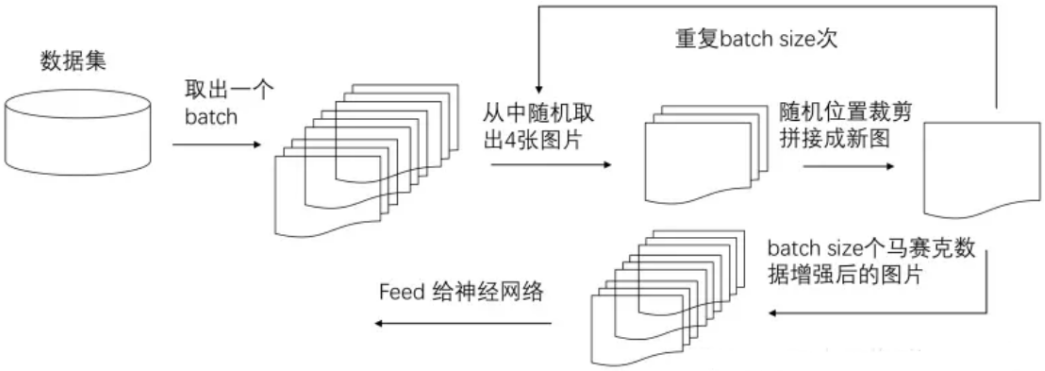
- **Mixup**: 通过按比例混合两张随机样本，相应地分配分类结果。
- **Cutout**: 随机切除样本中的一部分区域，并用零值填充，保持分类结果不变。
- **CutMix**: 切除一部分区域，但不填充零值，而是随机替换为训练集中其他数据的区域像素值，按比例分配分类结果

这三种方法的主要区别在于：

- Cutout 和 CutMix 主要在于填充策略的不同。
- Mixup 与 CutMix 在样本混合方式上有所区别：Mixup 通过图像间的比例插值混合样本，而 CutMix 通过切除和补丁的方式混合图像，避免了混合后可能出现的不自然现象。

YOLOv4 的 Mosaic 数据增强在 CutMix 的基础上进行了扩展，理论上具有相似之处，但 Mosaic 利用了四张图像。Mosaic 数据增强的实现方法可以分为以下几个步骤：

1. 随机选择四张图片：从训练集中随机选取四张图片。
2. 随机缩放和裁剪：对每一张图片进行随机缩放和裁剪操作，使其适应模型的输入尺寸。
3. 随机排列：将四张图片随机排列成一个矩形区域，形成一个新的训练样本。
4. 数据扩充：通过翻转、旋转等操作进一步扩充数据集。
5. 模型训练：将生成的新样本输入到模型中进行训练。



其具体代码实现方案可参考：https://blog.csdn.net/dgvv4/article/details/123988282

根据论文描述，Mosaic 的优点在于它不仅丰富了物体检测的背景，而且在进行批量归一化（BN）计算时，可以同时处理四张图像的数据。这种方法减少了对大 mini-batch 大小的需求，使得即使只使用一个 GPU，也能实现高效的训练效果。上图是一个实现的例子。

在平时项目训练时，小目标的 AP 一般比中目标和大目标低很多。Coco 数据集中包含大量的小目标，但比较麻烦的是小目标的分布并不均匀。Mosaic 数据增强使大、中、小目标分配更加均匀。

3.2.2. Self-Adversarial Training，SAT（自对抗训练）

YOLOv4 采用了自对抗训练，通过在训练过程中对原始图像进行扰动，模拟对抗性攻击，增加数据的多样性和复杂性，迫使模型学习更加鲁棒的特征。它灵感来源于**对抗性训练（Adversarial Training）**，但专注于在数据层面进行操作，而不对模型参数进行操作。其操作步骤为：

1. **原始图像修改**：在训练的前向传播阶段，神经网络在接收到原始图像之前，对其进行一系列随机的、可逆的变换，例如添加噪声、旋转、缩放、裁剪等。
2. **对抗性扰动**：这些变换可以设计得更具对抗性，例如，通过优化技术寻找能够最大程度降低模型性能的扰动。
3. **模型预测**：修改后的图像输入到模型中，模型进行正常的前向传播和预测。
4. **反向传播**：在损失函数计算后，通过标准的反向传播过程更新模型的权重。
5. **自适应性**：通过这种方式，模型被迫学习如何从经过扰动的图像中提取有效信息，从而提高其泛化能力。

3.2.3. CmBN

在深度学习领域，归一化技术对于提高神经网络性能至关重要。在上一篇中我们已经讨论过批量归一化（Batch Normalization，BN）通过规范化层输入，加速了训练过程并提高了模型的稳定性。然而，随着研究的深入，BN 的一些局限性也逐渐显现，尤其是在小批量尺寸的情况下，归一化统计的不稳定性可能会影响模型性能。

为了解决这一问题，YOLOv4 引入了一种创新的归一化技术——Cross mini-Batch Normalization（CmBN）。CmBN 的核心思想是在单个大批量中，跨多个小批量（mini-batches）进行特征的归一化处理。这种方法允许模型在每个训练迭代中收集更全面的激活分布统计信息，从而提高了归一化层的准确性和稳定性。CmBN 的操作流程如下：

1. 将一个大批量分割为多个小批量。
2. 对每个小批量独立进行归一化计算，获取均值和方差。
3. 将所有小批量的统计信息汇总，计算全局均值和方差。
4. 使用这些全局统计信息对每个小批量的数据进行归一化。

CmBN 的优势在于：

- **提高泛化能力**：通过跨多个小批量进行归一化，CmBN 减少了因批量大小不同而导致的性能波动。
- **增强稳定性**：CmBN 提供了更稳定的归一化统计信息，有助于模型在训练过程中的稳定性。
- **保持灵活性**：与 BN 相比，CmBN 在不同硬件和不同批量大小下都能保持一致的性能。

3.2.4. Label Smoothing 类标签平滑

在模型训练过程中，对预测结果持有 100%的信心可能暗示着模型仅仅在复现其训练数据，而非真正地学习到数据背后的模式。这种情况下，如果训练样本中混入了错误标记的样本，而模型又过于依赖这些样本，它将倾向于调整参数以紧密拟合这些错误，从而放大了这些错误样本的负面影响。

标签平滑技术通过设定一个低于 100%的上限值来调整预测目标，例如将上限设为 0.9。这意味着在计算损失时，即使模型对某个类别的预测非常有信心，损失函数也只会针对这个上限值（而非完美的 1.0）来评估。这种方法有助于缓解模型的过拟合现象。

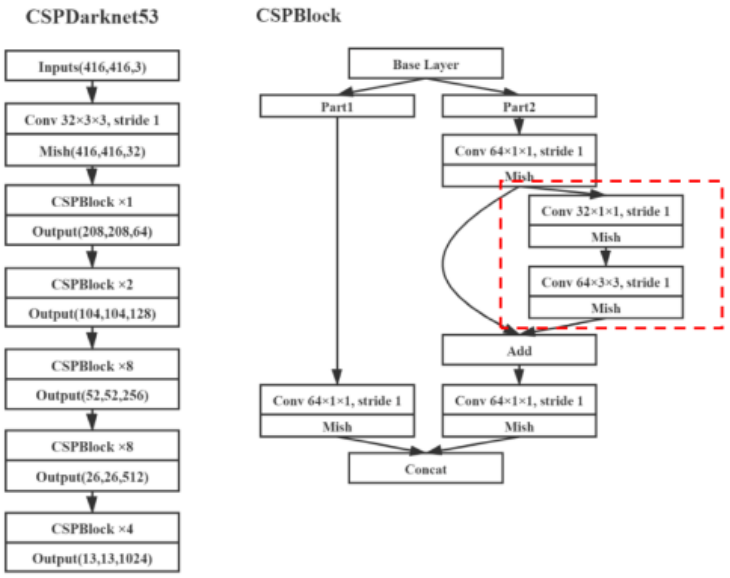
简而言之，标签平滑通过减少标签极值之间的差距，降低了模型对任何单一类别的过度自信，从而促进了模型的泛化能力。通过适度地调整标签值，使其不完全趋于极端（0 或 1），模型能够更好地捕捉数据的分布特性，增强了其在未知数据上的泛化性能。

3.3. 主干网络(Backbone)创新

3.3.1. CSPDarknet53

CSPDarknet53 是在 Yolov3 主干网络 Darknet53 的基础上，借鉴 2019 年 CSPNet 的经验，产生的 Backbone 结构，其中包含了 5 个 CSP 模块。

与 Darknet53 相比的第一点小改动，残差层之前的 2 层 conv+bn+Relu 变成了一层的 conv+bn+Mish，主要原因是把 Darknet53 中残差层之上用来下采样的 3x3 卷积放入了 Resblock_body 中。



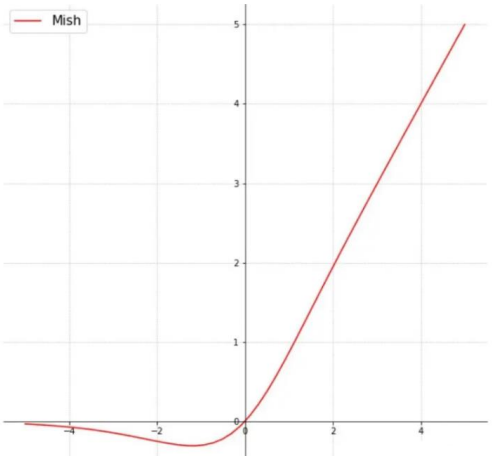
为什么采用 CSP 模块？

CSPNet 全称是 Cross Stage Paritial Network，主要从网络结构设计的角度解决推理中从计算量很大的问题。CSPNet 的作者认为推理计算过高的问题是由于网络优化中的梯度信息重复导致的。因此采用 CSP 模块先将基础层的特征映射划分为两部分，然后通过跨阶段层次结构将它们合并，在减少了计算量的同时可以保证准确率。我们将在该系列综述的 **Part4** 详细分析 CSPNet 网络结构。

3.3.2. Mish 激活函数

YOLOv4 实验了多种激活函数后，选择了 Mish 激活函数。Mish 因在各种任务中展现出的优异性能而被选用，它结合了 ReLU 和 Swish 的特点，同时解决了 ReLU 的梯度消失和 Swish 计算复杂的问题。

$$Mish(x) = x \times tanh(ln(1 + e^x))$$



从图中可以看出该激活函数，在负值时并不是完全截断，而允许比较小的负梯度流入从而保证了信息的流动。同时 Mish 激活函数无边界，这让他避免了饱和（有下界，无上界）且每一点连续平滑且非单调性，从而使得梯度下降更好。

Yolov4 的 Backbone 中都使用了 Mish 激活函数，而后面的网络则还是使用 leaky_relu 函数。

3.3.3. DropBlock 正则化

正则化技术是数据科学领域中用于规避过拟合的关键策略。过拟合是模型在训练数据上表现过于完美，以至于丧失了泛化能力的现象。为应对这一挑战，研究者们已经开发了多种正则化方法，包括 L1 和 L2 正则化、Dropout、Early Stopping 以及数据增强等。

在 YOLOv4 中，采用了 DropBlock 正则化方法，它是一种结构化的 Dropout 变体。DropBlock 通过随机地丢弃网络中的整个特征块，促进了网络学习更加分散和鲁棒的特征表示，进而增强了模型的泛化能力。DropBlock 的引入是为了解决传统 Dropout 在处理卷积神经网络中的局限性。尽管 Dropout 在全连接层中被证明是一种有效的正则化策略，但在特征空间高度相关的卷积层中，其效果并不理想。与随机丢弃单个神经元的 Dropout 不同，DropBlock 专注于在特征图的相邻区域中进行特征丢弃，这不仅有助于简化模型，还在每次训练迭代中促进了局部网络权重的概念学习，并通过补偿机制减少了过拟合的风险。如下图所示：

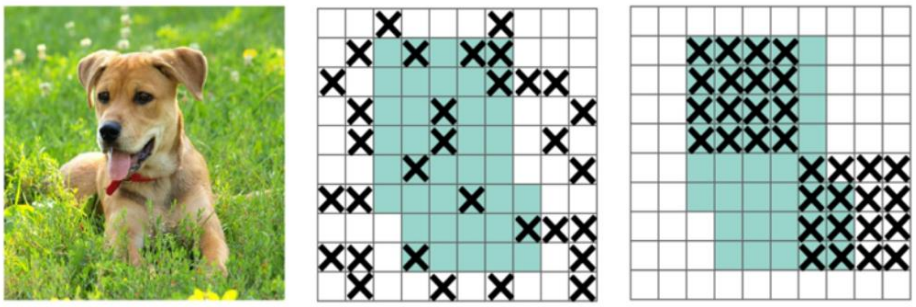


图 a 展示了卷积神经网络处理的输入图像。在**图 b** 和**图 c** 中，以绿色高亮显示的区域代表了网络中激活的单元，这些单元捕获了输入图像中的关键语义信息。这些区域是理解图像内容的核心部分。

然而，在应用 Dropout 时，随机地丢弃这些激活单元并不总是有效，因为它们在空间上是连续的，并且包含了紧密相关的信息。因此，即使部分单元被丢弃，剩余的单元仍然能够提供足够的语义信息，这限制了 Dropout 在去除特定特征时的效果。（**图 b**）

DropBlock 正则化的策略则不同。如**图 c** 的描述，DropBlock 不是随机地丢弃单个单元，而是选择性地丢弃连续的区域。例如，它可能在一个特征图上删除整个头部或脚部的区域。这种操作有效地移除了特定的语义信息，迫使网络中的其他部分学习到更加泛化的特征表示，以补偿丢失的信息。

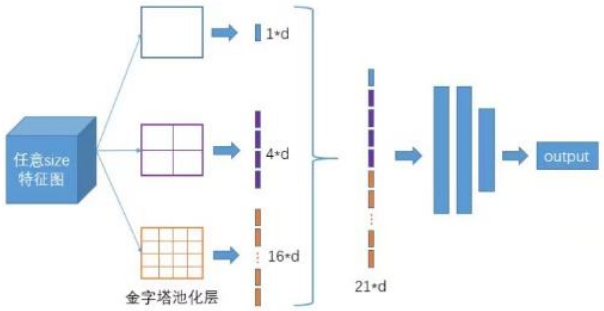
通过这种方式，DropBlock 正则化鼓励网络学习到更加鲁棒的特征，提高了模型对输入变化的适应能力，从而在图像分类等任务中提高了性能。

3.4. 颈部网络(Neck)创新

Yolov4 的 Neck 结构主要采用了 SPP 模块、FPN+PAN 的方式。

3.4.1. SPP 模块

SSP（Spatial Pyramid Pooling）池化是一种在 CNN 中用于特征提取的技术，是何恺明大佬提出的，它允许网络在不同尺度上捕获图像特征。这种池化方法通过在多个不同大小的窗口上应用池化操作来实现，从而生成一个多尺度的特征表示，这有助于网络更好地理解图像中的不同对象。如下图所示，下图中对任意尺寸的特征图直接进行固定尺寸的池化，来得到固定数量的特征。



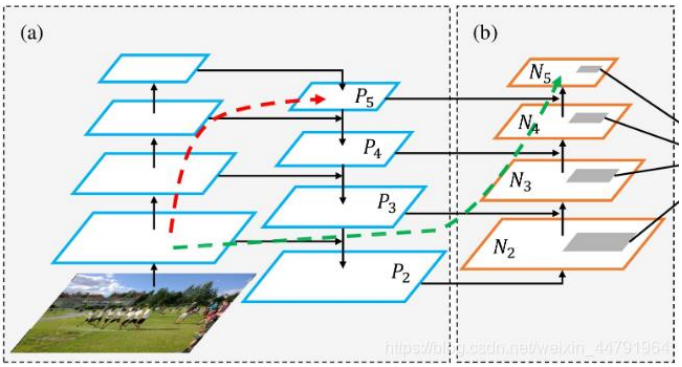
在 YOLOv4 的网络架构中，SSP 池化被用于主干网络的最后几层，以生成最终的特征图。这些特征图随后被送入检测头部进行目标的定位和分类。SSP 池化的应用使得 YOLOv4 能够在不同尺度上提取特征，这不仅有助于网络更好地识别不同大小的对象，而且通过在不同尺度上进行池化操作，还有助于提高模型的运行效率。

多尺度的特征表示增强了 YOLOv4 对不同尺寸和形状对象的泛化能力，从而提高了检测的准确性。此外，SSP 池化通过减少计算量，进一步优化了模型的性能。我们将在该系列综述的 Part5 详细分析这个模块。

3.4.2. FPN+PAN

Yolov4 中 Neck 这部分除了使用 FPN 外，还在此基础上使用了 PAN 结构。和 Yolov3 的 FPN 层不同，Yolov4 在 FPN 层的后面还添加了一个自底向上的特征金字塔，其中包含两个 PAN 结构。

FPN+PAN 借鉴的是 18 年 CVPR 的 PANet，PAN 其具体结构由反复提升特征的意思。当时主要应用于图像分割领域，但 Alexey 将其拆分应用到 Yolov4 中，进一步提高特征提取的能力。



上图为原始的 PANet 的结构，可以看出来其具有一个非常重要的特点就是特征的反复提取。在（a）里面是传统的特征金字塔结构，在完成特征金字塔从下到上的特征提取后，还需要实现（b）中从上到下的特征提取。通过增加横向连接，实现了更为有效的特征融合。

- 通过结合 FPN 和 PAN，YOLOv4 能够同时捕获图像的高分辨率和丰富语义信息，提高了对不同尺寸目标的检测能力。有助于保留图像的空间细节，对于小目标的检测尤为重要。
- 该方法通过缩短特征传播路径，减少了计算量，提高了模型的运行效率。
- 多尺度和多路径的特征融合增强了模型对不同场景的适应性，提升了检测的准确性。

3.5. 头部网络与模型训练

在前面的前面介绍介绍中，我们已经学习过 DropBlock 正则化、Mosaic 数据增强、SAT 自对抗训练、CmBN 等方法了，这几种种方法在模型训练的过程中也有用到，这里不在赘述。现在我们一起看下训练过程中采用的其他策略。

3.5.1. CIoU Loss

损失函数给出了如何调整权重以降低 loss。所以在我们做出错误预测的情况下，我们期望它能给我们指明前进的方向。但如果使用 IoU，考虑两个预测都不与 ground truth 重叠，那么 IoU 损失函数不能告诉哪一个是更好的，或者哪个更接近 ground truth。这里我们先分析常用的几种 loss 的形式，如下：

- 经典 IoU loss：

IoU Loss 是一种在目标检测任务中广泛使用的损失函数，用于衡量预测的边界框（与真实边界框之间的相似度。其公式为：

$$\text{IoU} = \text{Area of Overlap} / \text{Area of Union}$$

IoU Loss 通常以 1 减去 IoU 的形式作为损失函数。在训练过程中，目标是最小化 IoU Loss，从而最大化 IoU 值，使预测的边界框尽可能接近真实边界框。

- GIoU：Generalized IoU

GIoU 考虑到，当检测框和真实框没有出现重叠的时候 IoU 的 loss 都是一样的，因此 GIoU 引入了一个新的概念，即联合区域（Union）的尺度。它通过考虑预测框和真实框的最小外接矩形（smallest enclosing box）来计算它们之间的相似度。这样就可以解决检测框和真实框没有重叠的问题。但是当检测框和真实框之间出现包含的现象的时候 GIoU 就和 IoU loss 是同样的效果了。其公式为：

$$\text{GIoU} = \text{IoU} - \frac{\text{Area of Enclosing Box} - \text{Area of Union}}{\text{Area of Enclosing Box}}$$

其中 Area of Enclosing Box 是最小外接矩形的面积；Area of Union 是两个边界框并集的面积。

GIoU Loss 可以直接通过 1 减去 GIoU 值得到。

GIoU Loss 在边界框完全不重叠或完全重叠时仍然可以提供有效的梯度，这有助于改善模型的训练动态。但是他的计算比 IoU Loss 更复杂。

- DIoU：Distance IoU

DIoU Loss 是 GIoU Loss 的一个扩展，它进一步考虑了预测边界框和真实边界框之间的中心点距离，从而提高了对边界框定位精度的敏感性。DIoU Loss 的设计旨在解决 GIoU Loss 在某些情况下仍然无

法完全解决的问题，尤其是在边界框重叠度很高时，梯度可能会消失，导致模型难以进一步优化。DIoU Loss 公式为：

$$1 - \mathrm{IoU} + \frac{d^2}{c^2}$$

其中：d 是预测框中心点到真实框中心点的欧氏距离；c 是最小外接矩形的对角线长度。

- CIoU：Complete IoU

CIoU 就是在 DIoU 的基础上增加了长宽比（aspect ratio），其定义为定义为其宽度（width）除以高度（height），例如，一个很宽而不高的框可能有一个较大的长宽比，而一个高而不宽的框则有一个较小的长宽比。考虑长宽比可以显著提升边界框预测的准确性，特别是在目标检测任务中，不同对象（如行人与车辆）可能具有非常不同的形状。

这样预测框就会更加的符合真实框。CIoU Loss 公式为：

$$1 - \mathrm{IoU} + \frac{\rho_u}{c^2} + \alpha v$$

ρu 是预测框中心点到真实框中心点的欧氏距离的平方。

c 是最小外接矩形的对角线长度。

v 是一个基于宽高比的动态权重因子。

α 是一个根据 IoU 动态调整的权重因子，以平衡 v 的影响。

CIoU Loss 通过其全面性的特点，在提高目标检测模型性能方面显示出了潜力，尤其是在需要精确预测边界框的定位、大小和形状的任务中。通过确保预测框与真实框在长宽比上的一致性，CIoU 帮助模型学习到更加准确的边界框形状，从而改善整体的检测性能。尽管计算上更为复杂，但它提供了一种有效的边界框回归方法。

3.5.2. DIoU-NMS

DIoU-NMS（Distance Intersection over Union Non-Maximum Suppression）是一种在目标检测中用于后处理的改进型非极大值抑制（NMS）技术。这种方法主要用于消除冗余的边界框，确保每个目标只有一个最优的边界框被保留。在 YOLOv4 中，DIoU-NMS 被用来提高检测的准确性和鲁棒性。

在传统的 NMS 中，我们首先要计算所有预测框之间的 IoU 值。然后从所有候选框中选择一个得分最高（即置信度最高）的框作为基准框。

在 DIoU-NMS 中，除了计算 IoU 值外，还考虑了框的中心点距离。DIoU 值是在 IoU 的基础上加入了预测框和真实框中心点之间的距离因素，用来衡量两个框的相对位置。具体来说，它计算了：

- **中心点距离**：预测框和另一个框中心点之间的欧氏距离。
- **归一化因子**：使用包含两个框的最小闭合矩形的对角线长度来归一化这个距离。

DIoU-NMS 的优势在于：

- **减少位置偏移**：通过考虑中心点距离，DIoU-NMS 能够更有效地抑制位置偏移较大的重叠框，从而减少错误检测。

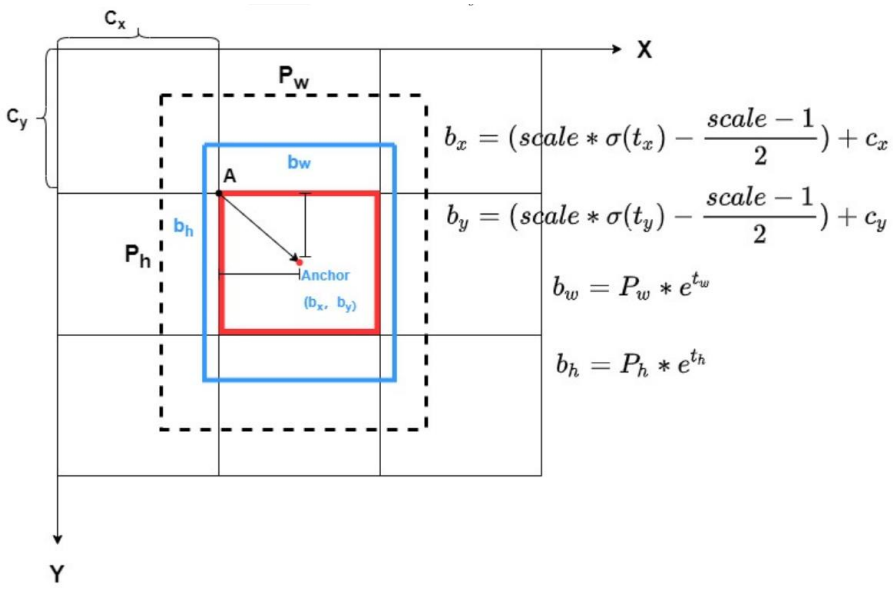
- **形状感知**：由于考虑了框的外形和位置，DIoU-NMS 在处理不同形状和尺寸的物体时，能提供更好的性能。

DIoU-NMS 提供了一种更为精确和鲁棒的后处理方法，显著提升了目标检测的整体性能。

3.5.3. 锚框偏移机制

yolov3 中对锚框中心点偏移量(t_x, t_y)和先验框缩放量(t_w, t_h)进行了限制，使得锚框中心点的偏移量在[0,1]之间，这样大大减少了锚框中心点偏移过远导致召回率和精度过低.

yolov4 中考虑了 sigmoid 函数的局限性，并对锚框中心点偏移量(t_x, t_y)的进一步限制，如下图所示



其中：(c_x, c_y)为网格中心点坐标；(P_w, P_h)为先验锚框宽与高；(b_x, b_y)为最终预测框的坐标；(b_w, b_y)为最终预测框的宽与高。

由上图可知，yolov4 在处理锚框的偏移量(t_x, t_y)的公式于 yolov3 不一样，由于 Sigmoid 函数将输入映射到 0 到 1 之间的输出，它在输入值趋向正负无穷大时输出结果才无限接近于 1 和 0。这种特性可能导致预测的边界框中心点（尤其是位于网格单元边缘的中心点）难以精确预测。为了改善这一点，YOLOv4 引入了一个缩放参数 scale，用于调整通过 Sigmoid 函数后的输出。这个改进意味着实际的t_x, t_y输出会被缩放，从而扩大它们的有效输出范围。这样做的目的是为了增加模型在预测靠近或位于网格单元边界的目标时的准确性和灵活性。

实际应用中，scale 参数的选择很关键，经验表明，当 scale 设置为 2 时，通常能够取得较好的效果。这样的设置帮助模型在整个网格单元内更平滑、更准确地调整预测框的位置，尤其是在预测边界接近网格边缘的目标时。所以公式可以简化为：

$$b_x = (2 * \sigma(t_x) - 0.5) + c_x \qquad b_w = P_w * e^{t_w}$$

$$b_y = (2 * \sigma(t_y) - 0.5) + c_y \qquad b_h = P_h * e^{t_h}$$

3.5.4. 锚框选择与调整

在 YOLOv3 和 YOLOv4 中，精确地选择和调整锚框对于提高检测准确性至关重要。这个过程主要涉及两个关键步骤：选择最合适的锚框模板和调整锚框的位置。

1- 锚框模板选择：

- 在每个网格单元中，模型需要预测目标的存在与否以及目标的位置。这一预测过程开始于选择一个与真实目标框（Ground Truth, GT）交并比（IoU）最高的锚框模板。
- 我们首先计算 GT 与一系列预定义的锚框模板的 IoU。这些锚框模板是根据训练数据集中常见的目标尺寸和形状预先定义的。选择 IoU 最高的模板，因为这意味着该模板在形状和大小上与 GT 最为接近。

2- 锚框位置调整

- 选定最佳锚框模板后，接下来的任务是调整该锚框，确保其中心与 GT 的中心对齐。这一步骤对于提高模型的定位精确度尤为重要。
- GT 的中心点可能并不完全位于其所在网格单元的几何中心。在这种情况下，模型会计算必要的偏移量t_x, t_y，这两个参数表示锚框中心点需要从当前位置移动到 GT 中心的相对距离。这些偏移量通过训练自动学习得到，并应用于预测过程中，以确保锚框尽可能精确地覆盖目标。

在 YOLOv3 中，锚框的尺寸是基于聚类分析预先定义的，并固定不变。这些尺寸通常基于训练数据集来优化，但在模型部署后不会进行调整。

YOLOv4 引入了自适应锚框尺寸的概念，允许模型在训练过程中自动调整锚框尺寸。这意味着模型可以更好地适应其训练数据的特定特征和变化，从而提高检测的准确性和泛化能力。YOLOv4 不仅继续使用 IoU 来优化锚框选择，还通过增加锚框调整的动态性，例如通过引入锚框调整的新机制和更复杂的损失函数，来改进锚框的选择和调整过程，这一点已经在先前提及。

3.5.5. Spatial Attention Module（SAM 模块）

SAM 是一种用于增强特征表达能力的注意力机制，通过专注于图像的空间维度，来改善目标检测的性能。SAM 模块的核心思想是突出那些重要的空间区域，以便模型更加关注于图像中目标的关键部分。这通过计算每个位置的重要性来实现，具体步骤如下：

1. **特征聚合**：首先，将输入的特征图通过全局平均池化和全局最大池化进行处理。这两种池化策略分别提取全局的平均信息和最极端的特征信息。
2. **融合与激活**：接着，这两个池化结果被融合（例如通过相加或串联），然后通过一个卷积层进一步处理，最终通过 Sigmoid 函数生成一个空间注意力图。这个注意力图的维度与输入特征图相同，但每个空间位置的值介于 0 到 1 之间，表示该位置的重要性。
3. **特征重加权**：最后，原始的输入特征图与空间注意力图逐元素相乘（element-wise multiplication），从而增强（或抑制）原始特征图中的某些区域。这样，模型的注意力就被引导至那些更为关键的区域。

在 YOLOv4 中，FPN 概念逐渐被实现/替换为经过修改的 SPP、PAN 和 PAN。SAM 通过这种方式显著增强了模型对于空间信息的处理能力。在目标检测任务中，这意味着模型能更好地定位和识别图像中的目

标，尤其是在背景复杂或目标尺寸、形状多样时。通过强化关键区域的特征表示，SAM有助于提高检测的精确度和减少误检。

3.5.6. 遗传算法优化超参

在YOLOv4中，**遗传算法（Genetic Algorithm, GA）**被用于优化模型的超参数，这是一种模拟自然选择过程的优化技术。遗传算法通常用于在复杂的、多维度的参数空间中找到最优解，尤其是当参数之间存在非线性关系且难以手动调整至最佳状态时。在YOLOv4的训练过程中，使用遗传算法来优化超参数可以帮助提升模型的性能，特别是在目标检测的精确度和速度方面。

3.6. 模型架构总结

所以，对于YOLOv4模型，我们可以总结为：

YOLOv4=(CSPDarkNet53+SPP+PANet+YOLOv3-head)⊕训练和推理策略

其中 ⊕ 表示“结合”，而“训练和推理策略”可以包括但不限于：

- 数据增强技术，如Mosaic和CutMix。
- 自对抗训练（SAT）。
- DropBlock正则化。
- 损失函数的选择，如CIoU Loss。
- 改进的注意力模块，如SAM。
- 改进的路径聚合，如PAN的修改。
- 跨小批量标准化（CmBN）。
- 超参数优化，遗传算法的使用。

4. 笔记中涉及相关知识点

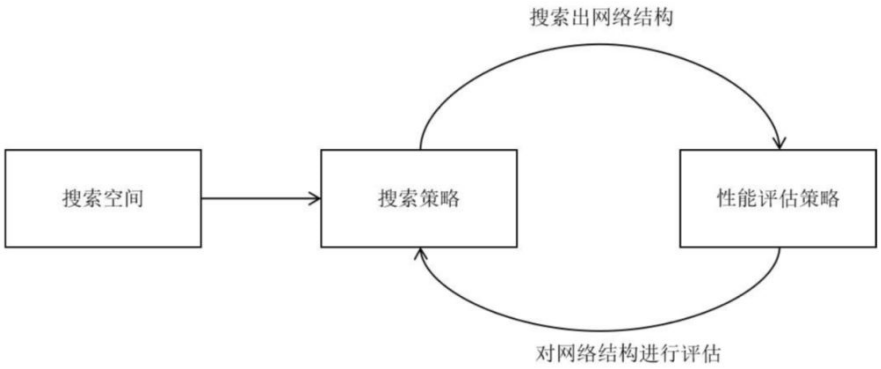
4.1. 人工神经架构搜索（NAS）

在深度学习领域，设计高性能的神经网络通常需要大量专业知识和反复试验，这不仅成本高昂，也限制了神经网络在许多应用场景中的使用。尽管各种神经网络模型层出不穷，但模型性能的提升往往伴随着对超参数更为严格的要求，一旦超参数设置稍有不同，就可能无法复现论文中的结果。其中，网络结构作为一种特殊的超参数，在深度学习的各个环节中扮演着至关重要的角色。

为了应对这些挑战，人工神经架构搜索（NAS，Neural Architecture Search）技术应运而生。NAS通过自动化的搜索过程，挑选出最佳的网络结构和超参数，优化特定任务如图像识别和语言处理的性能。此技术的一大优势是其能发现一些创新的网络结构，这些结构可能未被人类设计过，从而大幅降低了

神经网络的设计和实现成本。这一领域当前还处于高速发展阶段，各种新的方法不断出现。目前已有商业化的 NAS 系统，如 Google 公司的 Cloud AutoML 服务，百度公司的 AutoDL。

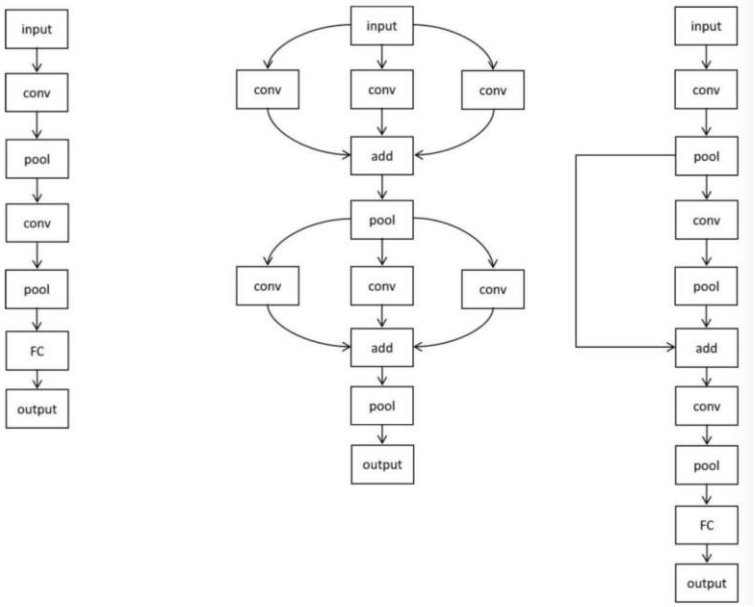
NAS 的核心包括定义一个广泛的候选网络结构集合，即“**搜索空间**”，然后通过各种**搜索策略**在这个空间中寻找最优解。每个候选解，或称为“子网络”，都在训练集上进行训练，并在验证集上进行**性能评估**。这个过程中，逐步优化子网络的架构，直到找到表现最佳的配置。搜索空间的定义、搜索策略的选择以及性能评估方法构成了 NAS 算法的基石，不同的实现方法导致了各种不同的 NAS 变体的产生。这一过程如下图所示：



4.1.1. 搜索空间

在人工神经架构搜索（NAS）中，搜索空间是至关重要的组成部分，因为它定义了算法可以探索的所有可能的神经网络架构。神经网络的计算过程可以被抽象为一个无孤立节点的有向无环图（DAG），其中图的节点代表神经网络的层，边则代表数据流的方向。每个节点从其前驱节点接收数据，处理后再将结果传递给后继节点。神经网络的结构定义包含如下层次的信息：

- **网络的拓扑结构：**网络有多少个层，这些层的连接关系。从简单的图结构到任意的 DAG 也反映了整个神经网络结构的发展历程。最简单的神经网络是线性链式结构，其对应的图的每个节点最多只有一个前驱，一个后续，类似于数据结构中的链表。早期的全连接神经网络，卷积神经网络都是这种拓扑结构。Inception、ResNet、DenseNet 中的节点允许有多个前驱，多个后续，从而形成了多分支、跨层连接结构，它们是更复杂的图。这些拓扑结构如下图所示：



- **每个层的类型**：除了第一个层必须为输入层，最后一个层必须为输出之外，中间的层的类型是可选的，它们代表了各种不同的运算即层的类型。典型有全连接，卷积，反卷积，空洞卷积，池化，激活函数等。但这些层的组合使用一般要符合某些规则。
- **每个层内部的超参数**：卷积层的超参数有卷积核的数量，卷积核的通道数，高度，宽度，水平方向的步长，垂直方向的步长等。全连接层的超参数有神经元的数量。激活函数层的超参数有激活函数的类型，函数的参数（如果有）等。各种典型层的超参数如下表所示：

层/运算	超参数
卷积	卷积核数量 卷积核通道数 卷积核宽度 卷积核高度 水平方向步长 垂直方向步长
池化	池化核高度 池化核宽度 水平方向步长 垂直方向步长
全连接	神经元数量
激活	激活函数类型 各种激活函数的参数
相加（add）	无
拼接（contact）	无

如果一个节点的前驱节点只有一个，则直接以前驱节点的输出值作为本节点的输入。如果前驱节点有多个，需要将前驱节点的值汇总后输入本节点，这里有两种策略：相加和拼接，前者的典型代表是 ResNet，后者的典型代表是 DenseNet。由于神经网络的层数不固定，每层的超参数数量也不固定，因此描述网络结构的参数是变长的。

为了提高搜索效率，有时候会搜索空间进行限定或简化。在某些 NAS 实现中会把网络切分成基本单元（cell，或 block），通过这些单元的堆叠形成更复杂的网络。基本单元由多个节点（神经网络的层）组成，它们在整个网络中重复出现多次，但具有不同的权重参数。另外一种做法是限定神经网络的整体拓扑结构，借鉴于人类设计神经网络的经验。这些做法虽然减少了 NAS 算法的计算量，但也限定了算法能够寻找的神经网络的类型。

由于描述神经网络结构的参数含有离散数据（如拓扑结构的定义，层的类型，层内的离散型超参数），因此网络结构搜索是一个离散优化问题。定义结构的参数数量一般比较大，因此属于高维优化问题。另外，对于该问题，算法不知道优化目标函数的具体形式（每种网络结构与该网络的性能的函数关系），因此属于黑盒优化问题。这些特点为 NAS 带来了巨大的挑战。

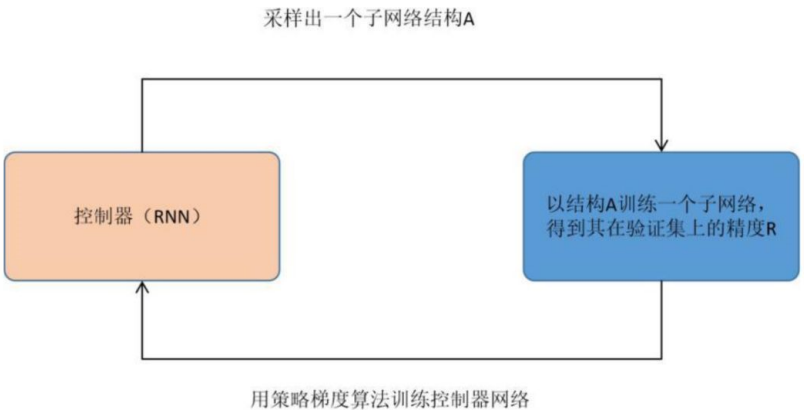
4.1.2. 搜索策略 - 基于强化学习的 NAS 算法

搜索策略定义了如何找到最优的网络结构，通常是一个迭代优化过程，本质上是超参数优化问题。其中强化学习，遗传学习，基于梯度的优化是目前的主流算法。我们将从强化学习开始介绍。

强化学习在连续决策建模中的应用广泛，特别是在神经网络架构搜索（NAS）领域。在这里，强化学习算法扮演智能体（agent）的角色，与环境交互来输出最优的神经网络架构。通过这种方法，智能体的动作决定了网络中各层的具体结构，而网络在验证集上的性能表现则作为回馈（奖励值）。

NAS 中的强化学习算法采用**循环神经网络（RNN）**作为控制器，负责生成网络结构的描述。这一过程可以被视为一个序列生成问题，其中每一步的输出定义了网络层的特定参数，例如卷积核数量、尺寸和步长。例如，卷积核的尺寸可能被限定为 [1, 3, 5, 7]，而 RNN 的 softmax 层输出这些尺寸的相对概率。

通过策略梯度方法，RNN 的参数根据子网络的验证集精度进行优化。在迭代过程中，高精度的网络结构被赋予更高的选择概率，从而推动控制器连续优化并输出最优结构。这一优化过程考虑了网络结构的层数，并在达到预设的层数后停止输出，确保模型的可管理性和计算效率。如下图所示：



面对 NAS 的计算量大和搜索空间广的挑战，一种常见的解决策略是简化搜索空间，例如通过预设的基本块（如 ResNet 的跨层连接块或 GoogLeNet 的 Inception 块）来构建网络。这不仅减少了计算资源的消耗，还允许网络结构根据输入数据的尺寸动态调整，增加了模型的灵活性和适用范围。

基于策略梯度的优化效率可能较低，并且由于子网络采样的随机性，可能产生较大的方差。尽管 RNN 能够有效生成网络描述，但无法直接通过模型精度进行优化。这要求算法在追求精度的同时，也需要考虑网络的其他性能指标，如延迟和能效。

4.1.3. 搜索策略 - 基于遗传算法的 NAS 算法

遗传算法是一种灵活的优化方法，可用于求解 NAS。这一方法依赖于自然选择的原理，通过迭代改进找到性能最优的神经网络结构。其算法实施步骤为：

- **子网络的初始化**：首先随机初始化一组子网络作为初始种群。每个子网络的结构被编码成一个固定长度的二进制串，这一编码代表了网络的各个级（stage）和节点（node）。
- **训练与评估**：在每次迭代中，所有子网络都会被训练，并在验证集上评估其性能，性能表现（通常是精度）用作适应度函数值。

- **交叉与变异**：根据适应度函数值选择若干优秀的子网络进行遗传操作。通过交叉和变异生成新一代的子网络结构，这些操作引入新的遗传多样性，有助于探索解空间。
- **迭代优化**：重复训练、评估和遗传操作的过程，直至达到预设的迭代次数或其他停止条件，最终确定表现最优的网络结构。

神经网络的每个级别都以池化层为界限，划分为包含多个卷积层的单元。每个节点代表一组卷积核，所有节点在一个级别内具有相同的通道数。网络中的每次卷积操作后均应用批量归一化和 ReLU 激活函数。全连接层不参与编码过程。对于每一组卷积核，数据处理包括逐元素相加的操作，随后执行卷积。

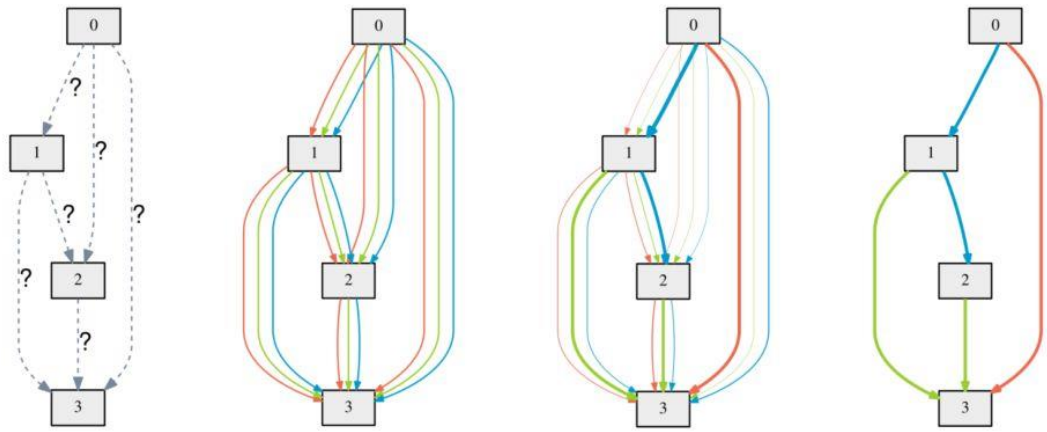
通过这种方式，遗传算法能够有效地搜索并优化复杂的神经网络架构，尽管计算量大，但可以通过适当简化搜索空间或引入先验知识来提高效率

4.1.4. 搜索策略 - 基于梯度下降的 NAS 算法

传统的神经网络架构搜索（NAS）方法常将问题处理为黑盒优化问题，操作在离散的网络空间中，这导致效率往往不尽人意。为了提高搜索效率，一种新的方法——可微分结构搜索（DARTS），被提出，DARTS（Differentiable Architecture Search）是一种基于梯度下降的 NAS 算法。这种方法的核心创新在于将网络架构搜索问题表述为一个可微分的问题，将网络空间表示为连续分布，使得可以直接应用梯度下降方法来有效地搜索最优的网络架构。

DARTS 方法与早期的 ENAS（Efficient Neural Architecture Search）类似，都将网络空间表示为一个有向无环图（DAG）。该方法的创新之处在于将节点间的连接和激活函数巧妙地组合到一个矩阵中，每个矩阵元素代表了连接和激活函数的权重。在架构搜索过程中，这些权重通过 Softmax 函数处理，从而将搜索空间从离散转换为连续，使得目标函数成为可微函数。

在搜索过程中，DARTS 遍历全部节点，并使用节点上所有可能连接的加权和来进行计算。这不仅优化了结构权重，也同时优化了网络权重。通过这种方法，搜索过程中的每一步都是基于梯度的，确保了搜索的高效性和精确性。如下图所示：



- **第一幅图**：展示了一个有向无环图（DAG），图中的节点（0, 1, 2, 3）表示网络层或计算单元，节点间的边（用问号表示）代表不同的网络操作，如卷积、池化等。这表示在 DARTS 的初始化阶段，所有可能的操作都是候选项，需要通过训练来确定最优操作。

- **第二三幅图**：展示了在 DARTS 优化过程中，每条边代表的操作都有一个权重，这些权重通过 softmax 函数参数化，以反映不同操作的选择概率。在进一步的优化之后，一些连接的权重变得更强，表示这些操作更有可能被选择为最终架构的一部分。
- **第四幅图**：表示最终选择的架构，只有权重最大的连接被保留，其他的都被移除，这表明了从连续搜索空间到离散网络架构的转换。确保了所选结构在训练过程中表现的优越性和效率。

4.1.5. 性能评估策略

在神经网络架构搜索（NAS）中，搜索策略的核心目标是发现一个最优化特定性能度量指标的网络架构，通常是未见数据集上的精度。为了指导这一搜索过程，NAS 算法需要有效估计给定网络架构的性能，这个过程被称为性能评估策略。

准确评估网络架构的性能需要在完整的训练集上进行训练和在验证集上测试，这是资源密集和时间消耗的。为了降低这些成本，开发了多种策略：

- **训练时间简化**：减少训练迭代次数或在训练样本的子集上进行训练，可以显著减少资源消耗，但可能会影响性能评估的准确性。
- **使用低分辨率图像**：在降低分辨率的图像上训练网络可以减少计算需求，同样也可能导致评估结果的偏差。
- **减少卷积核数量**：简化模型的复杂度，通过在训练时减少某些层的卷积核数量来降低计算负担。

为了进一步降低成本，一些高级方法被提出：

- **早期性能外推**：通过分析网络在训练早期阶段的性能来预测其最终性能。
- **权重共享**：通过共享先前训练过的网络权重来初始化新网络，从而加快训练速度和收敛。尽管这种方法可以显著提高训练速度，但也可能带来结果的偏差。

基于 One-Shot 的结构搜索是目前的主流方法，One-Shot NAS 方法通过定义一个超级网络（supernet），其中包含所有可能的网络结构，来实现搜索。在这种方法中，整个网络结构通过交替训练网络权重和模型权重来进行搜索，最终保留其中的最优子结构。

- **共享权重的偏差**：尽管权重共享提高了搜索效率，但不同的网络架构共享同一套权重可能会引入显著的偏差，因为不同的网络结构可能需要不同的参数配置以达到最佳性能。
- **内存和规模限制**：共享权重虽然能够降低资源需求，但在处理大规模网络时可能导致显存占用过高，限制了搜索的规模和深度。

4.2. 对抗性训练（Adversarial Training）

对抗性训练（Adversarial Training）是一种增强机器学习模型，尤其是深度神经网络，在面对对抗性攻击时的鲁棒性的训练方法。在对抗性攻击中，**输入数据被故意微小地修改**，以使模型做出错误的预测或分类。这种训练策略旨在通过明确训练网络识别并抵抗这类故意的干扰，从而提高模型的安全性和可靠性。对抗性训练涉及以下几个关键步骤：

- **对抗样本生成**：首先，基于当前模型生成对抗样本。这通常通过对输入数据应用小的、有目的的扰动来实现，这些扰动足以欺骗模型但又不至于让人类观察者察觉。常用的方法包括**快速梯度符号法（FGSM）**和**投影梯度下降（PGD）**。
- **模型训练**：然后，这些对抗样本被用作训练数据，训练模型正确地识别和处理这些扰动的输入。这种方法强迫模型学习和概括那些可能导致错误预测的特征扰动。
- **迭代优化**：通过迭代地生成新的对抗样本和重新训练模型，不断地强化模型对未来潜在扰动的抵抗能力。

对抗性训练是一种有效的策略，用于提升模型在面对敌意环境下的表现，特别是在安全关键的应用中非常重要。然而，应用对抗性训练需要考虑其对资源的需求和对模型性能的潜在影响，合理地设计训练过程和对抗样本生成策略。

4.3. PANet

PANet（Path Aggregation Network）是一种用于提升目标检测网络中信息流的神经网络架构。它最初由 Shu Liu 等人在 2018 年的论文中提出，主要目的是改进特征金字塔网络（Feature Pyramid Networks，FPN）在多尺度目标检测中的表现。PANet 通过增强特征层之间的连接，提高了小尺寸对象的检测准确性，并增强了各尺度特征的语义强度。

PANet 的核心在于增强 FPN 中不同层级间的信息流动，它通过以下几个关键技术实现这一目标：

- **自底向上的路径增强**：除了 FPN 中常见的自顶向下的路径，PANet 引入了从底层到高层的信息路径。这种结构帮助底层的高分辨率特征直接影响顶层的语义强的特征，增强了特征图的语义信息，尤其是在较小对象的检测上提供了帮助。
- **适配性特征池化**：PANet 采用适配性特征池化层（Adaptive Feature Pooling），使得网络能够在不同层级上整合特征，无论它们来自于哪个尺度。这样的设计使得每个检测框都能从所有尺度的特征中获取信息，提高了检测的准确性和鲁棒性。
- **全连接融合层**：在检测头部，PANet 利用全连接层来融合从不同层级抽取的特征，进一步强化了模型对各种尺寸对象的识别能力。

PANet 通过其创新的路径聚合结构，有效地解决了传统特征金字塔网络在处理多尺度对象时的局限性。通过增强低层与高层之间的信息流动，PANet 不仅提升了小尺寸对象的检测性能，还增强了整体网络的语义表示能力。此外，PANet 的这些特点使其在复杂环境下的目标检测任务中表现出色，成为了目标检测领域中的一项重要技术。

4.4. 遗传算法（Genetic Algorithm, GA）

遗传算法是一种综合了随机自适应全局搜索算法的技术，其理论基础来源于达尔文的进化理论（自然选择，优胜劣汰）和孟德尔的遗传学说（基因）。这类算法是对生物进化过程的一种数学仿真，为难以通过传统数学模型解决的问题提供了一种有效的解决方法。以下是遗传算法的核心原理概述：

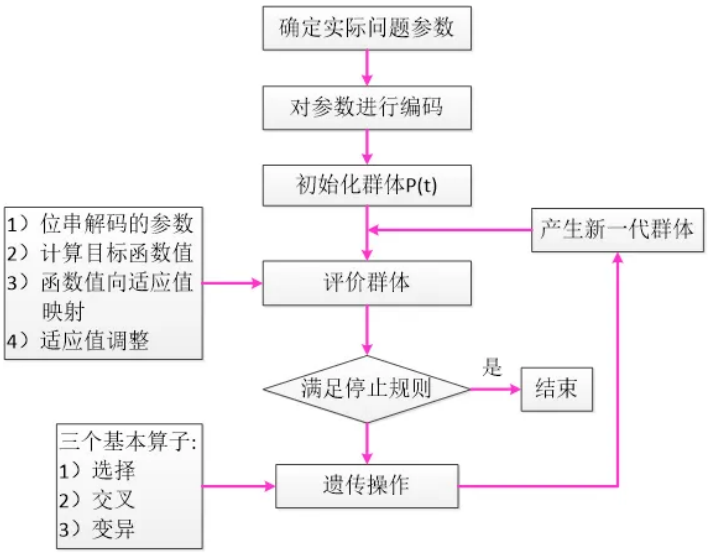
- **变异**：种群中单个样本的特征（性状、属性）可能存在差异，这导致样本之间具有一定程度的不同。变异确保了种群的多样性，为自然选择提供了必要的基础。

- **遗传**：某些特征可以从父母遗传给后代，这保证了后代与亲代具有一定程度的相似性。遗传是种群特征传递的基本方式，它保持了种群的连续性和稳定性。
- **选择**：种群通常在有限的环境中争夺资源。更适应环境的个体在生存和繁殖方面具有优势，因此能产生更多后代。自然选择是进化的驱动力，促进了适应性强的性状在种群中的保留和积累。

进化的重要推动因素是交叉（crossover）或重组（recombination）或杂交——通过结合双亲的特征产生后代。这一过程有助于维持种群的多样性，并随时间推移将更优秀的特征结合在一起。此外，变异（mutations）或突变（特征的随机变异）可以引入非预期的变化，为进化提供新的遗传变异。尽管大多数突变可能无益或有害，但偶尔也能产生对环境适应性更强的新特征，从而在自然选择中被保留下来。这些机制共同作用，使得种群随着世代的更迭逐渐适应其生存环境。遗传算法通过这些生物学原理的数学模拟，有效地解决了各种复杂的优化问题。

遗传学术语	生物学解释	遗传算法中的应用
染色体(Chromosome)	DNA 的结构形式，含有多个基因	解决方案的完整数据结构，如字符串或数字数组
基因(Gene)	染色体上的一段 DNA，决定个体的特征	解决方案中的一个单元，如参数或决策变量
基因型(Genotype)	个体的遗传信息，即其基因的组成	问题解决方案的编码形式
表现型(Phenotype)	基因型的物理表达，如体貌特征	基因型解码后的具体表现，即实际的解决方案
个体(Individual)	生物学中的一个生命实体	遗传算法中的一个潜在解决方案
适应性(Fitness)	一个生物体适应其环境的能力	解决方案的优良程度，用于评估和选择过程
种群(Population)	一组生物个体	所有当前潜在解的集合
交叉(Crossover)	染色体的交换材料过程，性繁殖的一部分	通过结合两个解的部分生成新解的方法
突变(Mutation)	随机 DNA 变化，可能导致新的性状	随机改变解决方案的某部分以引入多样性

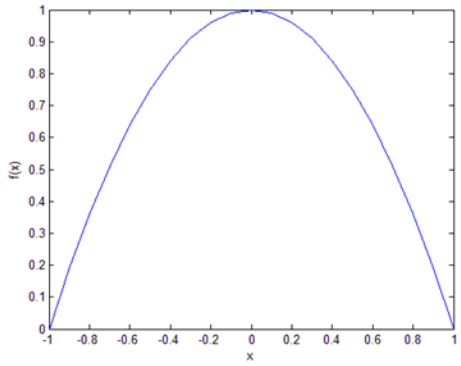
上表展示了自然界的遗传概念及其在遗传算法中的应用。以下为遗传算法的流程图：



我们将通过一个例子在介绍遗传算法的具体实现，我们假设问题是找出以下式子的最大值：

$$\max_{-1\geq x\leq 1} f(x), \quad f(x) = 1 - x^2$$

很显然，在该范围内存在 x=0 时，f(x)=1 最大。如下图:



至此，我们已经确定了问题的参数，即在[-1,1]上找出 f(x)的最大值。现在我们开始算法：

1- 对参数进行编码。

采用二进制编码，将某个变量值代表的个体表示为一个{0,1}二进制串。串长取决于求解的精度。如果确定求解精度到 3 位小数，由于区间长度为 1-(-1)=2，必须将区间[-1，1]分为 2×10^3 等份。所以二进制串长至少需要 11 位。<00000000000>与<11111111111>表示区间的两个端点-1 和 1。至此，我们对该问题进行了二进制串编码。

2- 初始化种群

一个二进制串叫做一个个体(individual)。有若干个个体组成个体的集合，称为种群(population)，种群中含有的个体的数量叫做种群的规模(population size)。随机生成初始种群：

$$
\begin{aligned}
S1 &=< 11100011101 > \\
S2 &=< 01010001111 > \\
S3 &=< 00110011011 > \\
S4 &=< 10111100110 > \\
S5 &=< 10011011011 > \\
S6 &=< 01110110000 > \\
S7 &=< 11110011001 > \\
S8 &=< 00001001011 >
\end{aligned}
$$

3- 评价群体

要评价群体，就得必须有一个评价标准，遗传算法是根据个体的适应值进行评价个体是否进入下一代的。很显然这里的适应值函数就是我们的目标函数 f(x)，直接将目标函数作为适应值函数。有了适应值函数，可以对初始种群进行评估。

$$
\begin{aligned}
S1(0) &=< 11100011101 > f(s1) = f(x1) = 0.393 \\
S2(0) &=< 01010001111 > f(s2) = f(x2) = 0.870 \\
S3(0) &=< 00110011011 > f(s3) = f(x3) = 0.642 \\
S4(0) &=< 10111100110 > f(s4) = f(x4) = 0.774 \\
S5(0) &=< 10011011011 > f(s5) = f(x5) = 0.954 \\
S6(0) &=< 01110110000 > f(s6) = f(x6) = 0.994 \\
S7(0) &=< 11110011001 > f(s7) = f(x7) = 0.189 \\
S8(0) &=< 00001001011 > f(s8) = f(x8) = 0.141
\end{aligned}
$$

4- 停止条件

在遗传算法中，常见的进化停止算法有三种：

- 设置进化代数，当种群进化 N 代之后，进化停止，选出适应值最高的个体，该个体即是最优解。
- 设置评价次数，当种群进化过程中的评价次数达到 M 后，进化结束，输出适应值最高个体。
- 种群收敛，如果种群收敛，则输出最优个体，进化结束。当然，如果没达到进化停止的条件，则对种群进行遗传操作，来产生新个体。

5- 遗传操作

遗传操作一般分为三种：交叉，变异，复制

- 交叉：两个个体随机以某个点为交叉点进行交叉点后的基因互换。如下图，两条染色体将第四个基因后面的基因互换：

$$
\begin{aligned}
s_2^{(0)} &=<0101|0001111> \qquad s'_2=<0101|0110000>, \quad f(s'_2)=0.893 \\
s_6^{(0)} &=<0111|0110000> \qquad s'_6=<0111|0001111>, \quad f(s'_6)=0.988
\end{aligned}
$$

- 变异：在某个基因上随机选出一个变异位置，将该位置上的基因进行随机互换。如下图选择第一个基因将 0 变异成为 1：

$$
s_3^{(0)} =<0110011011> \qquad s'_3=<10110011011>, \quad f(s'_2)=0.838
$$

- 复制：复制就是将优秀的个体，原封不动的复制到下一代种群中，以保存优秀基因。这里出现了一个问题：选择哪些基因进行遗传操作呢？

6- 适者生存

和自然进化一样在选择的时候一般按照一个原则：适应值高的存活的概率大，即选中进行遗传操作的概率大。一般有以下几个方法进行选择：

- 轮盘赌选择法 (Roulette Wheel Selection)：利用各个个体适应度所占比例的大小决定其子孙保留的可能性。
- 锦标赛选择法（tournament selection）：每次随机选取几个个体之中适应度最高的一个个体遗传到下一代群体中，重复 M 次.图片
- 随机遍历选择法：像轮盘赌一样计算选择概率，然后根据指针等距离地选择个体。图片这样，适应值高的个体存活的概率大，进行遗传操作的概率高，产生后代的机会就大，符合自然进化的选择方法。

在进行遗传操作后，在保证种群大小不变的情况下进行淘汰适应值低的个体。然后进行下一代进化。直至进化结束，产生出最优个体为止。

4.5. 循环神经网络（RNN）

循环神经网络（Recurrent Neural Network, RNN）是一类用于处理序列数据的神经网络。与传统的前馈神经网络（比如 CNN）不同，RNN 拥有一定的“记忆能力”，能够处理序列长度可变的输入数据，这使得它们非常适合于语言处理、时间序列分析等任务。RNN 的核心特点是它们在模型内部具有循环结构，这意味着网络能够将信息从一个时间步传递到下一个时间步。因此，RNN 能够保留过去的信息，并利用这些信息影响当前和未来的输出，这种特性使得 RNN 特别适合处理需要考虑时间依赖性的问题。

RNN 的基本构架中，每一个单元接收来自前一个时间步的输入以及当前时间步的新输入。这些输入通过网络中的权重处理，合并成一个状态，通常称为隐藏状态。隐藏状态随后被传递到下一个时间步，同时也用来计算当前时间步的输出。因此，在 RNN 中，同一组权重被用于每一个时间步的数据，这种权重共享极大地减少了模型的复杂性和所需的训练参数量。

然而，传统的 RNN 在处理长序列时面临着“梯度消失”或“梯度爆炸”的问题，这使得网络难以学习到依赖于长距离时间步的特征。为了解决这一问题，研究者提出了几种改进型的 RNN，如长短时记忆网络（Long Short-Term Memory, LSTM）和门控循环单元（Gated Recurrent Unit, GRU）。这些变体通过引入门控机制来调节信息的流动，有效地解决了长序列训练中的梯度问题，从而使得网络能够学习到更长距离的依赖关系。

总的来说，循环神经网络是一种强大的工具，适用于各种需要处理顺序数据的应用。无论是在自然语言处理、语音识别还是时间序列预测等领域，RNN 都展示了其独特的优势。随着研究的深入和技术的发展，RNN 及其变体将继续在复杂序列建模任务中扮演重要角色。

4.6. 快速梯度符号法（FGSM）

快速梯度符号法（Fast Gradient Sign Method, FGSM）是一种用于生成对抗性样本的简单而有效的方法，由 Ian Goodfellow 等人在 2014 年提出。该技术的核心思想是利用输入数据的梯度信息来创

建可以欺骗神经网络的新输入。FGSM 通过对输入数据进行一次梯度更新，以最大化损失函数，从而快速有效地生成对抗性扰动。这种方法的主要优势在于其计算效率高，仅需要一步梯度计算便可产生对抗样本，使得它在实际应用中非常受欢迎。

FGSM 的操作简单直观：首先计算输入数据相对于模型损失的梯度，然后使用这个梯度的符号乘以一个小的扰动因子（通常称为 ϵ 或 epsilon），添加到原始输入上，从而产生对抗性样本。这个过程简化了对抗样本的生成，但也有效地揭示了神经网络在面对微小扰动时的脆弱性。FGSM 广泛用于评估模型对对抗攻击的抵抗力，以及在对抗性训练过程中增强模型的鲁棒性。这种方法虽然有助于理解和改进神经网络的安全性，但同时也暴露了当前模型在安全关键应用中可能面临的风险。

4.7. 投影梯度下降（PGD）

投影梯度下降（Projected Gradient Descent, PGD）是一种在机器学习中常用于生成对抗性样本的方法，尤其在测试模型对抗攻击鲁棒性的场景中得到广泛应用。PGD 可以视为快速梯度符号法（FGSM）的一种迭代版本，它通过多步迭代优化过程来精细调整扰动，生成更具攻击性的对抗样本。与 FGSM 相比，PGD 在每次迭代中对输入数据进行微小的扰动，并将结果投影回合法的扰动空间（通常是原始数据周围的一个 ϵ-ball），确保生成的样本不会偏离原始样本太远。

PGD 的核心步骤包括计算模型输出相对于输入数据的梯度，使用这些梯度信息来制造出可以最大化预定损失函数的对抗性扰动。通过连续多次迭代这一过程，并在每一步中适当限制扰动的幅度（即投影步骤），PGD 能够逐渐找到导致模型错误分类的最优或近似最优扰动。这种方法被认为是生成对抗性样本中较为严格和有效的方式，常用于评估和提升深度学习模型在实际应用中的安全性和鲁棒性。