杭州电子科技大学

硕 士 学 位 论 文

题　　目：基于强化学习的目标检测算法研究

研 究 生　　舒 朗

专　　业　　电子与通信工程

指导教师　　郭春生　副教授

完成日期　　2018 年 4 月

杭州电子科技大学硕士学位论文

基于强化学习的目标检测算法研究

研 究 生： 舒朗

指导教师： 郭春生 副教授

2018 年 4 月

Dissertation Submitted to Hangzhou Dianzi University

for the Degree of Master

Research on Object Detection Based on Reinforcement Learning

Candidate: **Shu Lang**

Supervisor: **Associate Professor Guo Chunsheng**

April，2018

杭州电子科技大学
学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本章的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。申请学位论文与资料若有不实之处，本人承担一切相关责任。

论文作者签名：舒朗 日期：2018 年 6 月 11 日

学位论文使用授权说明

本人完全了解杭州电子科技大学关于保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属杭州电子科技大学。本人保证毕业离校后，发表论文或使用论文工作成果时署名单位仍然为杭州电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。（保密论文在解密后遵守此规定）

论文作者签名：舒朗 日期：2018年6月11日

指导教师签名：郭春生 日期：2018 年 6 月 11 日

摘 要

目标检测是计算机视觉领域最热门的研究方向之一，其目的是在输入的图片或者视频中定位所有出现的目标，并确定其每个目标所属的类别，典型的目标检测过程是使用一个边界框将目标紧紧包围起来。得益于卷积神经网络强大的特征提取能力，基于深度学习的目标检测算法在准确度与速度上相较于传统检测算法均取得了巨大的提升，但即便如此，大量冗余候选区域的处理过程已经成为限制算法速度的瓶颈。针对上述问题，本文从减少所需处理的候选区域数量出发，提出了两种基于深度强化学习的目标检测算法：

1. 联合边框回归的深度强化学习目标检测算法。基于强化学习的目标检测算法在检测过程中通常采用预定义的搜索行动，其产生的候选区域的形状和尺寸变化单一，导致目标检测的精确度较低。为此，在基于 DQN 的目标检测算法基础上，提出了联合边框回归与深度强化学习的目标检测算法。算法首先由 DQN 根据初始候选区域所提取的信息决定相应的搜索行动，根据行动选择下一个逼近真实目标的候选区域；然后重复上述过程，直至 DQN 有足够的信心确定当前区域为目标区域时，终止搜索过程；最后由回归网络对当前区域坐标进行边框回归，达到精确定位的目的。在 Pascal VOC 单类别数据集上的实验结果表明，通过引入边框回归有效地提高了视觉目标检测的精确度。

2. 基于多层特征与深度强化学习的视觉目标检测算法。由于不同尺寸大小的目标在不同深度的特征网络上的表达能力不同，仅使用单层特征图的目标检测算法，很难保证所有尺寸大小的目标信息都能得到充分表达，导致此类算法对尺寸变化较大的目标的检测效果较差。因此，为了使不同尺寸大小的目标的特征都能够得到充分表达，本文在基于深度强化学习的目标检测算法基础上，引入多层特征，智能体能够在进行区域搜索同时，按照候选区域-特征映射关系，提取相应的特征层上的特征，实现多层特征与强化学习相结合的目标检测。在 Pascal VOC 数据集单类别目标检测中的实验结果显示，相较于未使用多层信息的基于深度强化学习的目标检测算法，该算法能够有效提高检测的准确率，验证了本文算法的有效性。

关键词：目标检测，强化学习，深度学习，多层特征

ABSTRACT

Object detection is one of the hottest research directions in the field of computer vision. The goal of object detection is to locate all the presented objects in the input image or video and determine the category to which each object belongs. The typical object detection process is to cover the object with a bounding box tightly. Since the powerful feature extraction capabilities of convolutional neural networks, object detection algorithms based on deep learning have achieved tremendous improvements in accuracy and speed compared to traditional detection algorithms, however, there is still a large number of redundant regions need to be processed. The process of the regions has become a bottleneck that limits the speed of the algorithm. In order to reducing the number of regions that need to be processed, this paper proposes two object detection algorithms based on deep reinforcement learning.

1. Deep reinforcement learning for visual object detection with bounding box regression. The object detection algorithm based on reinforcement learning usually adopts predefined search actions in the detection process, the shape and size of the proposal regions generated by them are not changed much, resulting in a low accuracy of object detection. For this reason, based on the deep Q-Network (DQN) object detection algorithm, we proposed an object detection algorithm by combining bounding box regression with deep reinforcement learning. Firstly, the DQN determines the search action according to the information extracted from the initial proposal regions, and then selects the next proposal region approaching the ground truth according to the action. Then repeat the above process until DQN has enough confidence to determine the current region as the ground truth, and then the search process is terminated. Finally, the current region coordinates are regressed by the regression network to achieve a better localization. The experimental results on the Pascal VOC single-category dataset show that the accuracy of visual object detection is effectively improved by the introduction of bounding box regression.

2. Object detection algorithm based on multi-layer features and deep reinforcement learning . The objects of various sizes have different expression capabilities on the same depths of feature maps, using only single-layer feature maps for all object may result in poor performance in detection. Therefore, in order to fully express the features of the object of different sizes, this paper introduces multi-layer features on the basis of the object detection algorithm based on deep reinforcement learning. The reinforcement learning agent can extract the corresponding feature

layer according to the region-feature mapping and combination of multi-layer features and reinforcement learning for object detection. The experimental results in the single-category object detection of Pascal VOC dataset show that the proposed algorithm can effectively improve the accuracy of detection compared with the algorithm based on deep reinforcement learning without using multi-layer information, which proved the advantages of the proposed algorithm.

Key Words: object detection, reinforcement learning, deep learning, multi-layer features

目 录

第 1 章　绪论

1.1 论文研究的背景及意义

人们在观察一幅图片时能够立刻知道图片中目标的位置以及目标的类别，使我们不需要太多的思考过程就能够完成很多复杂的任务，这对人类来说是非常轻松的事情，而对于计算机来说，其实现起来却相当困难。对于计算机而言，这些图片只不过是由二进制组成的数据，而数据背后所代表的事物，计算机却很难像人类那样理解。人们每天接触到的大量的信息，而其中绝大部分信息属于视觉信息，而在这些视觉图像中通常仅有某些特定的目标才是我们所关心的。通过对这些我们感兴趣的目标进行准确地定位与识别能够大幅度减少计算机所需要处理的数据量，提高信息处理的效率。如今，目标定位已经成为各种应用的关键技术，比如机器人系统、自动驾驶技术、智能交通监管系统等。

在现实世界中针对图像的目标定位效果主要受到两方面的影响，一是图像方面，一是算法方面。图像方面主要有两点，一是拍照时的光照、天气以及相机内部产生的噪声对图像质量的影响，二是图像中的内容，如目标所处背景过于复杂或与目标过于相似、目标不完整以及不同目标之间的相互遮挡等。算法方面主要集中在如何提取高质量特征，以及如何使算法具有良好的准确度以及实时性。因此，如何设计能够满足准确定位且在速度上达到工业级的要求的目标定位算法是研究的关键。

传统的方法主要通过手工设计的特征来进行目标检测，此类方法的性能很大程度上取决于这些手工特征的质量，通常这些特征是由在该领域具有深厚经验的专家针对某一特定类别目标设计的，很难泛化到其他的类别。得益于 CNN 强大的特征提取能力，基于深度学习的目标检测算法极大地提高了目标检测的精确度以及检测速度。这类目标检测方法大多基于 R CNN，主要改进为减少候选区域的数量或者优化候选区域生成算法，但这些方法仍然需要对大量具有冗余性的候选区域进行处理。

作为人工智能领域重要的研究内容之一，强化学习已经广泛应用于工业制造、优化与调度、机器人控制等领域，其通过智能体(agent)不断与环境进行交互，agent 从不断试错的过程中学习到一种最优策略，通过该策略能够最大化其从环境中获得的累积奖赏值。强化学习更加侧重于决策的优化，适用于复杂决策系统的构建。为了提高目标检测的速度，本文从减少评估的候选区域出发，结合深度学习技术与强化学习技术，并将其应用于目标检测场景中。该目标检测算法并不需要对每个候选区域进行评估，通过引入强化学习智能体，使其能够根据当前收集到的信息，有选择地对特定的候选区域进行评估，将候选区域的搜索过程序列化，在一定步骤内检测到目标，从而减少所需评估的候选区域数量，进而提高目标检测的速度，

对视觉目标检测技术领域具有重要的研究意义。

1.2 目标检测技术的国内外研究现状

根据目标检测技术的发展历程，可将目标检测分为基于传统方法的目标检测算法、基于深度学习的目标检测算法以及基于强化学习的目标检测算法，下面将围绕着该三个方面，对国内外的目标检测研究进展进行阐述。

1.2.1 基于传统方法的目标检测算法

传统的视觉目标检测算法通常基于滑动窗口的框架，如图 1.1 所示，该框架将目标检测过程分为三个步骤，首先利用不同尺寸的窗口，在图像中以一定步长进行滑动，选择窗口中的区域作为候选区域；然后对该候选区域使用特征提取器提取相关的视觉特征，该特征提取器通常是根据目标类别而精心设计的，比如用于人脸检测的 Haar 特征[1]、用于行人检测的 HOG[2]特征以及一般目标检测中常用到的 SIFT[3]特征等；最后利用一些使用大量训练数据预训练过的分类器对这些候选区域进行分类识别[4]，常用的分类器有支持向量机 SVM[5]，Adaboost[6]等。典型的基于传统方法的目标检测算法为可变部件模型（Deformable Parts Model，DPM）[7]，该算法将目标看作由多个组件构成，使用组件之间的关系来描述目标。在 HOG 的基础上，将目标的模板划分为类似传统 HOG 的根模型与代表组件的部分模型，在检测的时候，根模型用来对目标可能存在的位置进行定位，而使用部分模型来进行进一步的确认。其检测效果要明显优于基于传统 HOG 的算法，在 2007 年至 2009 年的 Pascal VOC[8]视觉目标检测任务中取得了第一名的好成绩。但由于 DPM 过于复杂[9]，导致其检测过程需要大量的计算，在检测速度方面仍存在较大的提升空间。

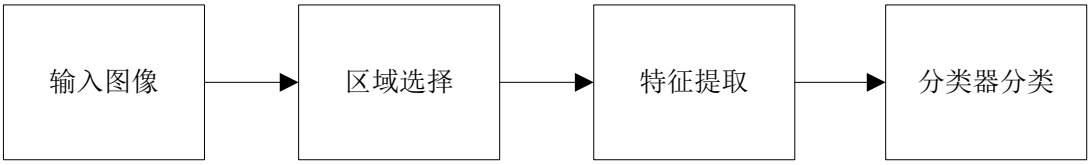


图 1.1 传统目标检测框架

基于传统方法的目标检测算法主要存在以下两方面的缺陷：一是由于在图像中，目标可能出现在任何位置，且目标的尺寸大小、宽高比各不相同，若采用滑动窗口的方式对图像区域进行遍历，为了覆盖到形状、大小浮动较大的各个目标，需要设置不同的尺度以及不同的宽高比的窗口。因此基于滑动窗口的区域选择策略缺乏针对性，时间复杂度高且容易产生大量冗余窗口，对接下来对这些区域进行的特征提取和分类的速度和性能影响较大；二是受目标的形态多样性、光照变化以及复杂背景等条件影响，提取的特征质量需要领域专家的精心设计来保证，且这些特征非常依赖特定目标与场景，对现实场景中的目标缺乏鲁棒性，而特征提取过程的速度直接影响到算法的应用范围。

1.2.2 基于深度学习的目标检测算法

在 1990 年代，CNN 大量应用于计算机视觉任务中，但是当支持向量机 SVM 出现之后，CNN 的发展开始了一段低迷期。直到近些年来随着科技的发展，计算能力得到大幅度地提高，手机、数码相机等设备的普及极大地降低了数据获取的难度，使得使用 CNN 从大量原始数据中提取高层特征成为可能,在计算机视觉[10][11]，语音识别[12][13]等方面取得了方面取得了巨大的进步。在 2012 年的 ILSVRC[14]比赛中，Krizhevsky 等人[15]通过构建层数更深的卷积神经网络 CNN，在图像分类任务上取得了显著的成绩，引起了国内外研究者对深度学习技术的广泛关注。针对传统目标检测中区域选择策略缺乏针对性，产生大量冗余窗口，导致算法时间复杂度高的问题，出现了以候选区域(region proposal)取代滑动窗口来进行区域选择的方案。Region proposal 利用图像的纹理、边缘、颜色等信息来产生候选区域，能够在减少生成的候选区域数量的同时，获得质量较高的候选区域，极大地降低了后续处理的复杂度,典型的候选区域生成方法如 CPMC[16]，Selectvie Search[17],MCG[18]等。

2013 年 Erhan 等人[19]提出了 Multibox，其将目标检测建模为多个边框坐标回归问题，网络在输出多个边框坐标的同时，给出每个边框包含目标的置信度分数，将边框检测器的训练作为网络训练的一部分，为了提高检测器的对多个目标类别的泛化性能，以无关类别的方式对网络进行训练，在 ILSVRC2012 上取得了领先的成绩。2014 年，Sermanet 等人[20]提出一种集成了识别、定位和检测任务的目标检测算法 OverFeat，该算法使用一个卷积网络实现了多尺度与滑动窗口，并提出了一种基于深度学习通过预测目标边界来进行定位的方法，Sermanet 的工作表明了可以通过使用单一的共享网络来同时学习不同的任务，该算法获得了 2013 年 ILSVRC 目标定位任务的冠军。2014 年 Ross Girshick 等人[21]将 Region Proposal 与 CNN 特征结合起来，提出了 R-CNN 检测算法。该算法在 PASCAL VOC 以及 ImageNet 数据库上取得了领先的成绩，但 R-CNN 仍存在训练步骤繁琐、占用空间大以及速度慢等缺点。为了提高目标检测的速度，He 等人[22]提出了 SPP-Net，SPP-Net 通过在候选区域与卷积特征图之间建立映射关系，实现对输入图片进行一次特征提取，就能得到所有候选区域的卷积特征。为了保证提取的特征维度相同，SPP-Net 使用了空间金字塔采样的方式对特征区域进行采样。由于不需要对每个候选区域重新计算卷积特征，因此其相比 R-CNN 极大地提高了目标检测的速度。2015 年 Yoo 等人[23]提出了 AttentionNet,其将目标检测问题转换为迭代分类问题,AttentionNet 能够通过预测一些指向目标的弱方向,最终获得精确的边界盒。Ross Girshick 等人[24]在 SPP-Net 的基础上提出了 Fast R-CNN，将特征映射从空间金字塔采样映射改为使用 ROI pooling 层进行映射，同时引入多任务损失函数，将边框回归任务加入到网络中进行联合训练。由于 Fast R-CNN 使用 Selective Search 的方法来产生候选区域，该候选区域生成算法占据了目标检测过程大量时间，因此，该算法在检测速度上仍存在着较大的提升空间。为了提高检测速度，Ren 等人[25]在 Fast R-CNN 的基础上提出了 Faster R-CNN，其主要改进在于利用 RPN 网络取代 Selective Search 来生成高质量的候选区域。RPN 网络是一种小型网络，使用该小型

网络在最后的卷积层上进行滑动，由于使用了 anchor 机制以及边框回归，因此通过这种方式可以得到多种尺度及宽高比的候选区域，将 region proposal 与 CNN 分类进行融合，能够实现端到端的目标检测，在检测速度和精度上均有较大的提升。2016 年 Dai 等人[26]提出了基于区域的全卷积网络 R-FCN，用于精确而有效的目标检测。基于候选区域的检测器如 Faster R-CNN，在特征图上数百次地应用含有全连接层的子网络来产生候选区域，而 R-FCN 所有的计算都是基于卷积操作，因此其在整个图像上的是可以共享计算的。R-FCN 提出位置敏感分数映射，以解决图像分类中平移不变性与目标检测中平移变化性之间的难题。因此，R-FCN 可以采用完全卷积的图像分类网络架构，如最新的残差网络 ResNet[27]用于目标检测，在 VOC2007 上取得了 83.6% mAP 的成绩。现有的基于区域的目标检测器局限于使用固定几何形状的区域来表示目标，而这些区域往往并不是矩形的，为了应对该问题，Mordan 等人[28]于 2017 年提出了一种面向目标检测的 DP-FCN 模型，该模型适用于具有可形变部分的目标。在没有附加标注信息的情况下，模型能够学会了专注于那些具有区分度力的元素并将它们对齐，同时为分类和几何信息带来更多的不变性，以细化定位。DP-FCN 由三个主要模块组成：一个全卷积网络，以有效地保持空间分辨率；一种基于可变形部件的 ROI 池层，用于优化部件的位置和建立不变性；以及一个能够利用部件位移的形变感知定位模块，以提高边界盒回归的精度，在 VOC2007 与 VOC2012 上取得了不错的成绩。

基于候选区域的目标检测算法是目标检测领域的主流，虽然其检测精准度较高，但由于其需要处理大量的候选区域，在检测速度上仍很难满足实时性要求，而基于回归的深度学习检测算法弥补了这一缺陷。Joseph Redmon 等人[29]提出 YOLO 检测框架，该框架通过将输入图像划分为 7×7 的网格，然后对每个网格进行预测两个边框，每个边框包含目标的置信度以及所属目标类别的概率，最后去除可能性较低的目标窗口以及冗余的窗口，即可得到最终检测的结果。YOLO 应用回归的思想，将目标检测问题转化为回归问题，能够大幅度地提高检测的速度，但由于其仅使用 7×7 的网格进行回归，不能对目标进行精准的定位，因此其检测精度有所下降。为了提高 YOLO 的精准度，Liu 等人[30]将 YOLO 中的回归思想与 Faster R-CNN 中的 anchor 机制结合起来，提出了 SSD 检测框架。该检测框架利用多个不同分辨率的特征图来处理不同尺寸大小的目标，在预测时网络对每个默认框中的目标预测其属于各类别的分数，并对使用每个网格位置的局部特征进行回归，以更好地匹配目标的形状，既能保证 YOLO 的检测速度，也能达到近似 Faster R-CNN 的准确度。

1.2.3 基于深度强化学习的目标检测算法

2013 年 DeepMind 在 Atari 上的成功开启了深度强化学习的研究热潮，2016 年的 AlphaGo[33]的成功更是将深度强化学习的研究推向了新的高度[31]。研究人员纷纷投入对深度强化学习的研究中，并积极将深度强化学习向各个领域拓展，其中就有不少学者为将深度强化学习引入目标检测任务做了非常多的尝试。Mathe 等人[34]提出了一种序列模型，能够通过累积从一小部分图像位置上收集的证据来有效地检测视觉目标。将序列化搜索过程转化为强

化学习中的策略搜索过程，可以有效地平衡每个类别，其检测速度能够比滑动窗口速度提高两个数量级。Caicedo 等人[35]提出了一种基于深度强化学习目标定位算法，该算法将整幅图片看作一个环境，通过引入一个 agent 来学习对边界框进行自顶向下的搜索策略，该 agent 能够根据学习到的策略对边界框执行一系列简单的变形行动，最终将目标准确定位出来。Bueno 等人[36]提出了一种基于分层的深度强化学习目标检测框架，该框架主要思想是根据收集的线索，不断将注意力聚集在包含更多信息的区域，通过训练一个 agent 根据当前收集的信息，从预定义的五个候选子区域中选择最有可能包含目标的区域，极大的提高检测的速度，但在精确度方面仍存在不足。不同类别目标之间存在着能够某种相互关联，利用这种关联能够促进更有效的搜索，基于该思想，Kong 等人[37]提出了一种基于协作的深度强化学习来进行不同目标的联合搜索算法，该算法将每个检测器看作一个 agent，使用基于多 agent 协作的深度强化学习算法来学习协同目标定位的最优策略，通过利用这些上下文信息，该算法能够有效提高目标定位的准确度。当对一个场景进行感知时，人类能够有能力对场景中的多个不同的位置和尺度的感知点进行感知，而这些感知点通常具有不同的场景内容。当前的视觉目标检测方法中缺少这种机制，因此在 2017 年 Hara 等人[38]提出了一种用于视觉目标检测任务的增强深度神经网络，该网络引入了一个注意机制能够自适应地在图像中的不同位置与形状的区域进行序列化的探索，通过对这些区域探索，提取目标存在以及其位置的证据，并将这些信息融合用于估计目标的类别以及其边界盒坐标，针对这些不同位置的探索，Hara 等人使用了基于策略梯度的强化学习算法来实现。实验表明，该算法性能优于不使用注意力机制的目标检测网络。

由于目前主流的视觉目标检测为了达到较高的检测效果，其多依赖于大量的候选区域，因此如何在减少候选区域数量的同时，保留高质量的候选区域，依旧是目标检测技术研究的重点。

1.3 本文的主要内容和结构

本文主要对基于深度强化学习的目标检测进行研究，提出联合边框回归的深度强化学习视觉目标检测算法和提出基于多层特征与深度强化学习目标检测算法，本文的各章节结构安排如下：

第一章，介绍了本文的研究背景和意义，并从基于传统传统方法的目标检测、基于深度学习的目标检测以及基于深度强化学习的目标检测三个方面对目标检测技术的国内外研究现状作简要的介绍。

第二章，首先介绍强化学习的相关理论基础，着重介绍马尔可夫决策过程 MDP，以及构成强化系统的组成部分；其次介绍深度学习相关理论，对神经元模型的构成以及训练深度网络的反向传播算法原理进行描述；最后对深度学习与强化学习的结合进行分析，并重点介绍本文所使用到的 DQN 的相关原理。

第三章，提出联合边框回归的深度强化学习视觉目标检测算法。基于强化学习的目标检

测算法在检测过程中通常采用预定义的搜索行动，其产生的候选区域的形状和尺寸变化单一，导致目标检测的精确度较低。为此，在基于 DQN 的目标检测算法基础上，提出了联合边框回归与深度强化学习的目标检测算法。算法首先由 DQN 根据初始候选区域所提取的信息决定相应的搜索行动，根据行动选择下一个逼近真实目标的候选区域；然后重复上述过程，直至 DQN 有足够的信心确定当前区域为目标区域时，终止搜索过程；最后由回归网络对当前区域坐标进行边框回归，达到精确定位的目的。在 Pascal VOC 数据集上的实验结果显示，在 dog 与 areoplane 类别目标检测中，与 Caicedo 算法相比，其准确率分别提高了 12.86%与 10.99%，表明通过引入边框回归有效地提高了视觉目标检测的精确度。

第四章，提出基于多层特征与深度强化学习的视觉目标检测算法。由于不同尺寸大小的目标在不同深度的特征网络上的表达能力不同，仅使用单层特征图的目标检测算法，很难保证所有尺寸大小的目标信息都能得到充分表达，导致此类算法对尺寸变化较大的目标的检测效果较差。为了充分利用多层网络特征，本文在基于深度强化学习的目标检测算法基础上，引入多层特征，智能体能够根据候选区域的尺寸大小，按照候选区域-特征映射关系，提取相应的特征层上的特征，实现多层特征与强化学习相结合的目标检测。在 Pascal VOC 数据集中 areoplane 与 areoplane 类别目标检测中的实验结果显示，该算法相较于 Bueno 算法在准确率上分别提高了 12.01%与 16.15%，验证了本文算法的有效性

第五章，对本文在基于强化学习目标检测算法领域中所做的研究进行总结，分析研究中仍存在的问题，以进一步确认此后需要进行研究的方向。

第 2 章　深度强化学习基础

深度学习[39]是机器学习的一个重要分支，近些年来，深度学习的研究取得了巨大的进展[40]，在图像分析、自然语言处理以及视频分析等领域取得了显著的成果[40]。典型的深度学习模型是层数很深的神经网络，其能够根据输入的原始数据逐层提取更为抽象的特征信息，以此来学习数据集的高层信息表征。强化学习是机器学习领域的一个重要研究方向，是一种通过和环境不断地交互进而从试错中学习最优策略的方法。强化学习通过定义一个智能体来与环境进行交互，智能体输入是对其所在环境的观察信息，其根据这些信息来决定接下来的行动，通过行动来与环境进行交互，智能体根据环境对该行动的反馈来调整自身的参数，学习状态到动作的映射策略，以达到累积期望奖赏最大化的目的。基于强化学习的目标检测算法主要应用了深度学习理论与强化学习理论，因此，本章首先介绍强化学习的相关理论基础，着重介绍马尔可夫决策过程，以及构成强化学习系统的组成部分；其次介绍深度学习，对神经元模型的构成以及训练深度网络的反向传播算法进行描述；最后对如何将深度学习与强化学习进行结合，进行相关分析，并对 DQN 的相关原理进行阐述。

2.1 强化学习

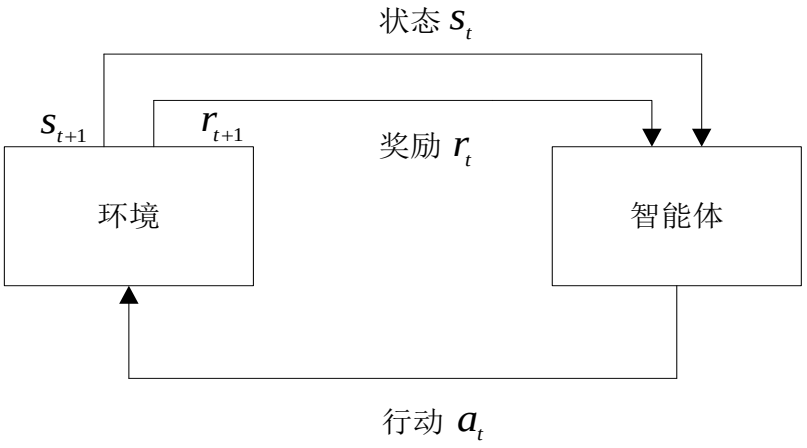


图 2.1 强化学习模型

强化学习的本质是从试错中学习[46]，其模型如图 2.1 所示，一个智能体(agent)通过执行相应的行动与它所处的环境进行互动，观察环境对自己行动的反馈，进而改变自己的行为来响应自己收到的奖赏。这种试错学习的模式符合行为心理学的认知，同时也是强化学习的基础。在强化学习架构中，拥有行动自主权的 agent，在 t 时刻能够通过从环境中观察到状态 s_t，在状态 s_t 下，agent 通过采用行动 a_t 来与环境交互。当 agent 执行行动 a_t 之后，环境基于当前

选择的行动 a_t，将状态从 s_t 转移到一个新的状态 s_{t+1}。这里假设的前提条件是，状态 s 是对环境的充分统计，因此其包含 agent 采取行动所需的所有信息。

最优的行动序列由环境提供的奖赏决定，每次环境从一个状态转移到另一个状态，作为反馈，环境都会提供一个标量化的奖赏给 agent。而 agent 的目标就是学习一个策略来最大化从环境中期望得到的回报，包括累积奖赏和折扣奖赏等。强化学习与最优控制要解决的问题在这方面表现相同，然而与最优控制不同的是，强化学习 agent 必须在环境中通过试错来学习执行行动带来的后果，而且对于 agent 来说，状态转移的模型是不可知的，agent 通过与环境的每一次交互所得到的信息来更新自己的知识。

强化学习算法与一般的机器学习算法不同，其区别主要体现在如下几点：

1) 没有监督器，环境只能够给予一个相应的奖赏信号。
2) 奖赏具有滞后性，执行当前行动的影响，可能需要经过较长的时间后才能够得到体现。
3) 数据之间具有高度相关性，非独立同分布。
4) agent 当前所执行的行动会影响到后面所有的子过程。

强化学习是一个序列决策最优化的过程，其目标是根据当前对环境的观察，选择相应的动作，能够最大化将来得到的回报。动作的选取不仅仅取决于当前奖赏的多少，更多的是依据该动作在长期能够带来多少的回报。因此，强化学习 agent 在很多情况下需要牺牲短期的奖赏来获取最大化将来的回报。

一个完整的强化学习系统主要包括四个组成部分：策略、奖赏函数、价值函数以及相应的环境模型。下面将对强化学习的各个部分分别进行阐述。

2.1.1 强化学习组成部分

2.1.1.1 策略

策略(policy)定义了 agent 在给定时间内的行为方式。也就是说，策略是 agent 对环境的感知状态到所要采取的动作之间的映射。它对应于心理学中的一组应激或联想反应。在某些情况下，策略可能是一个简单的函数或查找表，而在其他情况下如搜索过程，则可能涉及额外的计算。策略是增强学习 agent 能够充分确定自身行为的核心，一般来说，策略是确定的，但也可能是随机的。

2.1.1.2 奖赏

奖赏信号定义了强化学习问题的目标，在每个时间步骤，环境向强化学习 agent 发送一个标量数字，也就是奖赏 reward，agent 唯一的目标就是最大化其在长时间段内收到的总的奖赏。该奖赏信号因此定义了在当前情境下对 agent 来说什么是好的行动，什么是坏的行动。可以将奖赏看作是生物系统中快乐或者痛苦的经历，奖赏是即时的，同时奖赏也定义了 agent 所面对的问题特征，是改变策略的主要依据。当策略选择了一个带来低奖赏的行动，那么在下次遇到相同情境下时，策略会选择一些其他的行动，而不是该带来低奖赏的行动。一般来

说，奖赏信号是以环境状态 s 与采取行动 a 为参数的随机函数。

2.1.1.3 价值函数

奖赏信号表明当前场景下什么是好的，而价值函数则具体指明了从长远来看什么是好的。粗略地说，一个状态的价值是 agent 从该状态开始可以期望在将来积累奖励的总额，而奖赏决定当前环境状态的即时的愿望。价值表明了在考虑到接下来可能出现的状态，以及这些状态可获得的奖赏之后，环境状态长远的愿望。例如，一个状态可能总是生成一个低的即时奖赏，但仍然有很高的价值，这是因为该状态之后经常伴随着其他能生成高额奖赏的状态，反之亦然。奖赏就像人类的快乐和痛苦，然而价值是对我们处于某种特定环境状态下，有多么高兴或不高兴的一种更为精确和有远见的判断。

对于 agent 来说，奖赏作为对事件的评价，是主要的；而价值一般可看做是对奖赏的预测，是次要的。没有奖赏就没有价值，而估计价值的唯一目的是获得更多的回报。然而，在制定和评估决策时，我们最关心的是价值。行动选择是基于价值判断的，我们寻求能带来最高价值而不是最高奖赏的行动，因为这些行动使我们在长期获得最大的回报。一般来说，确定价值要比确定奖赏困难，这主要是因为奖赏基本上是由环境直接给出的，而价值必须从 agent 整个生命周期所作的观察序列中不断地估计、再估计。事实上，几乎所有的强化学习算法都是一种能够有效估计价值的方法。

2.1.1.4 环境模型

构成强化学习系统的最后一个部分是环境模型，但该部分并不是系统所必需的。环境模型能够模仿环境的行为，或者更一般地说，它允许对环境将要执行的反馈作出推断。例如给定一个当前具体的状态以及执行的动作，环境模型可以预测下一个状态以及相应的奖赏。模型可以被用于规划，通过在实际经历之前考虑所有可能的未来情况来决定应该执行的行动方案。使用模型和规划解决强化学习问题的方法被称为基于模型的方法，而基于无模型的算法，通常是明确通过不断地试错进行学习。

2.1.2 马尔可夫决策过程

马尔可夫决策过程(Markov Decision Process, MDP)起源于随机优化控制，是强化学习领域研究的关键问题之一，在决策与规划领域中，MDP 提供了一种简单、通用的表达方法。强化学习通常建立在马尔可夫决策过程的思想之上，马尔可夫决策过程完整地表征了强化学习的环境模型，绝大部分的强化学习问题都能够表示为一个马尔可夫决策过程，其基本思想为根据当前的状态信息即可给出一个合理的行动，该行动的给出与之前的状态信息无关，即认为未来的发展仅与当前时刻有关。状态 S_t 被认为是马尔可夫的，当且仅当：

$$P\left[S_{t+1}|S_t\right]=P\left[S_{t+1}|S_1,...,S_t\right] \tag{2.1}$$

其中函数 $P(.)$表示状态转移概率函数，状态 S_t 中包含了所有的所需要的历史相关信息，一旦状态已知，那么之前历史中的所有信息都可以丢弃，当前的状态能够充分反映未来所有的可

能。形式上，MDP 由以下几个部分组成：

1) 状态集合 S，以及初始化状态分布 $p(s_0)$

2) 行动集合 A

3) 状态转移函数

$$T_{ss'}^{a}=T\left[S_{t+1}=s'|S_t=s,A_t=a\right] \tag{2.2}$$

在 t 时刻从一个状态 s_t 与行动 a_t 映射到下一个状态 s_{t+1} 的分布。

4) 一个即时奖赏函数

$$R_s^a=E\left[R_{t+1}|S_t=s,A_t=a\right] \tag{2.3}$$

5) 一个折扣系数 $\gamma\in[0,1]$，较低的 γ 表明模型更加看重即时奖赏。

一般来说，策略 π 是一种从状态到行动映射的概率分布：

$$\pi:S\to p\left(A=a\,|\,S\right) \tag{2.4}$$

在模型已知的情况下，对任意策略 π 能估计出该策略带来的期望累积奖赏；函数$V^{\pi}(s)$是在给定策略条件下，状态 s 在该策略下的期望回报，定义$Q^{\pi}(s,a)$指的是从状态 s 出发，执行动作 a 后再使用策略 π 所期望带来的累积奖赏。这里，$V(\cdot)$称为“状态价值函数”，而$Q(\cdot)$称为“状态-动作值函数”，分别表示指定“状态”上以及指定“状态-动作”上的期望回报。当 MDP 过程是情节式的，例如每当一个情节达到一定的长度 T，状态就会被重置，那么在一个情节中，状态、行动以及奖赏构成了一个策略的轨迹。Agent 根据策略 π 执行相应的动作会从环境中累积奖赏，得到回报值：

$$G_t=r_1+\gamma r_2+...+\gamma^{T-1}r_T=\sum_{t=0}^{T-1}\gamma^t r_{t+1} \tag{2.5}$$

强化学习的最终目标是找到一个最优策略π^*，该策略对于所有状态都能达到最大的期望回报：

$$\pi^*=\arg\max_{\pi}\mathrm{E}\left[G\,|\,\pi\right] \tag{2.6}$$

由累积奖赏定义，有定义状态值函数如下：

$$\begin{cases} V_T^{\pi}(s)=E_{\pi}\left[\dfrac{1}{T}\sum_{t=1}^{T}r_t|s_0=s\right] & T\text{ 步累积奖赏；}\\[2ex] V_{\gamma}^{\pi}(s)=E_{\pi}\left[\sum_{t=0}^{+\infty}\gamma^t r_{t+1}|s_0=s\right] & \gamma\text{ 折扣累积奖赏.} \end{cases} \tag{2.7}$$

其中 s_0 表示起始状态，a_0 表示起始状态上采取的第一个动作；对于 T 步累积奖赏，用下标 t 表示后续执行的步数。可定义状态-动作值函数：

$$\begin{cases} Q_T^{\pi}(s,a) = E_{\pi}\left[\frac{1}{T}\sum_{t=1}^{T} r_t \middle| s_0 = s, a_0 = a\right] \\ Q_{\gamma}^{\pi}(s,a) = E_{\pi}\left[\sum_{t=0}^{+\infty} \gamma^t r_{t+1} \middle| s_0 = s, a_0 = a\right] \end{cases} \tag{2.8}$$

由于 MDP 具有马尔可夫性质，即系统下一时刻的状态仅由当前时刻的状态决定，不依赖于以往的你和状态，因此值函数可以写成以下的简单递归形式，对于 T 步累积奖赏，对其进行全全概率展开：

$$\begin{aligned} V_T^{\pi}(s) &= \mathrm{E}_{\pi}\left[\frac{1}{T}\sum_{t=1}^{T} r_t \middle| x_0 = x\right] \\ &= \mathrm{E}_{\pi}\left[\frac{1}{T} r_1 + \frac{T-1}{T}\frac{1}{T-1}\sum_{t=2}^{T} r_t \middle| s_0 = s\right] \\ &= \sum_{a\in A}\pi(s,a)\sum_{s'\in S} P_{s\to s'}^{a}\left(\frac{1}{T} R_{s\to s'}^{a} + \frac{T-1}{T}\mathrm{E}_{\pi}\left[\frac{1}{T-1}\sum_{t=1}^{T-1} r_t \middle| s_0 = s'\right]\right) \\ &= \sum_{a\in A}\pi(s,a)\sum_{s'\in S} P_{s\to s'}^{a}\left(\frac{1}{T} R_{s\to s'}^{a} + \frac{T-1}{T} V_{T-1}^{\pi}(s')\right) \end{aligned} \tag{2.9}$$

其中，$R_{s\to s'}^{a}$表示在状态 s 下，执行动作 a，转换到状态 s' 时，环境所给予 agent 的奖赏，$P_{s\to s'}^{a}$表示执行动作 a 后，由状态 s 转移到 s' 的概率。相应地，对于折扣累积奖赏的状态值函数有：

$$V_T^{\pi}(s) = \sum_{a\in A}\pi(s,a)\sum_{s'\in S} P_{s\to s'}^{a}\left(R_{s\to s'}^{a} + \gamma V_{\gamma}^{\pi}(s')\right) \tag{2.10}$$

由以上状态值函数 V 可直接计算出相应的状态-动作价值函数：

$$Q_T^{\pi}(s,a) = \sum_{a\in A} P_{s-s'}^{a}\left(\frac{1}{T} R_{s\to s'}^{a} + \frac{T-1}{T} V_{T-1}^{\pi}(s')\right) \tag{2.11}$$

$$Q_{\gamma}^{\pi}(s,a) = \sum_{s'\in S} P_{s-s'}^{a}\left(R_{s\to s'}^{a} + \gamma V_{\gamma}^{\pi}(s')\right) \tag{2.12}$$

强化学习的任务的目的是最大化累积奖赏，其理想策略如下所示：

$$\begin{aligned} \pi^* &= \arg\max_{\pi}\sum_{s\in S} V^{\pi}(s) \\ &= \arg\max_{a\in A} Q^{\pi}(s,a) \end{aligned} \tag{2.13}$$

通过将 Bellman 等式(2.5)与(2.6)中对动作的求和改为求最优，可得到最优的状态值函数表达式：

$$V_T^*(s)=\max_{a\in A}\sum_{s'\in S}P_{s\to s'}^a\left(\frac{1}{T}R_{s\to s'}^a+\frac{T-1}{T}V_{T-1}^*(s')\right) \tag{2.14}$$

$$V_T^*(s)=\max_{a\in A}\sum_{s'\in S}P_{s\to s'}^a\left(R_{s\to s'}^a+\gamma V_\gamma^*(s')\right) \tag{2.15}$$

相应的最优状态-动作值函数可表达为以下形式：

$$Q_T^*(s,a)=\sum_{a\in A}P_{s-s'}^a\left(\frac{1}{T}R_{s\to s'}^a+\frac{T-1}{T}\max_{a'\in A}Q(s',a')\right) \tag{2.16}$$

$$Q_\gamma^*(s,a)=\sum_{s\in S}P_{s-s'}^a\left(R_{s\to s'}^a+\gamma\max_{a'\in A}Q_\gamma^*(s',a')\right) \tag{2.17}$$

从一个策略出发，通过对策略或值函数进行评估与改进，不断迭代直到策略收敛、不再改变为止，即可得到最优的策略。

2.2 深度学习

2.2.1 神经元

神经网络通常是由大量简单的非线性单元通过并行互连构成的网络，其能够模拟生物的神经网络对外界刺激所做出的交互反应。构成神经网络基本单元是神经元模型，在生物学中，神经元与神经元相连，当一个神经元受到外界刺激而被激活时，就会向与其相连的神经元传递相应的信号，从而改变该神经元内的电位；如果该神经元的电位超过一个阈值，该神经元将会被激活，从而向其他神经元传递信号。

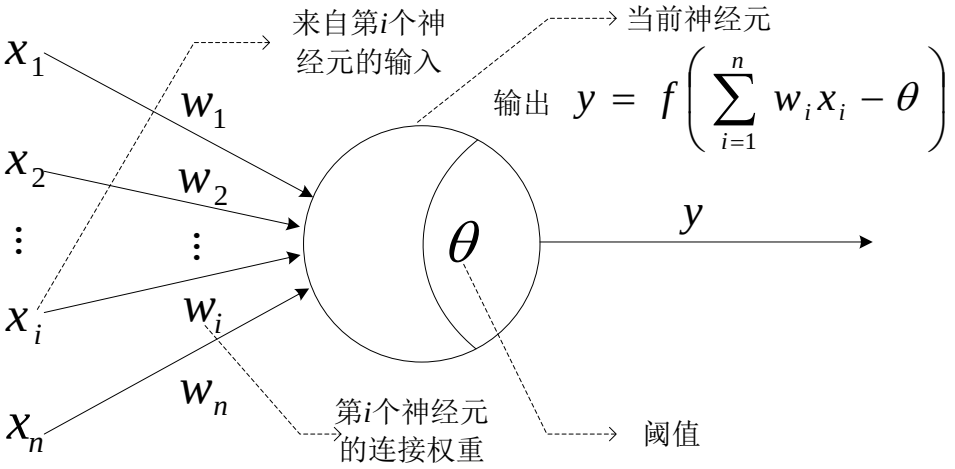


图 2.2 神经元模型

上述过程可抽象为如图 2.2 的简单模型，该模型称为“M-P 神经元模型”。多个神经元向该神经元传递信号，这些输入经过不同的权重加权后被神经元所接收，神经元将加权后的输入值总值与内部的阈值进行比较，最后由激活函数对结果进行处理，即产生神经元相应的输出：

$$y = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right) \tag{2.18}$$

其中 y 代表神经元的输出，函数 f 代表激活函数，w_i 与 x_i 代表第 i 个输入参数与相应的连接权重。理想的激活函数是阶跃函数，其将输入值映射为输出值“1”或“0”，其中“1”代表高电平，即神经元处于兴奋状态，而“0”代表低电平，即神经元处于抑制状态。

$$\mathrm{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2.19}$$

由于阶跃函数具有很多缺陷，如不连续，不光滑等，因此实际应用中的激活函数通常使用 Sigmoid 函数，其能够将较大范围内的输入值挤压到(0,1)输出范围内。

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{2.20}$$

将多个神经元按照一定的层次结构连接起来，即可组成一个相应的神经网络。

2.2.2 误差反向传播算法

多层神经网络具有强大的学习能力，理论上，只需要一个包含足够多神经元的隐层，该网络就能够拟合任意复杂度的连续函数。但此时常见的学习规则并不能够完成网络的训练[47]，现实任务中多使用误差反向传播算法(error BackPropagation)，简称 BP 算法进行神经网络的训练。

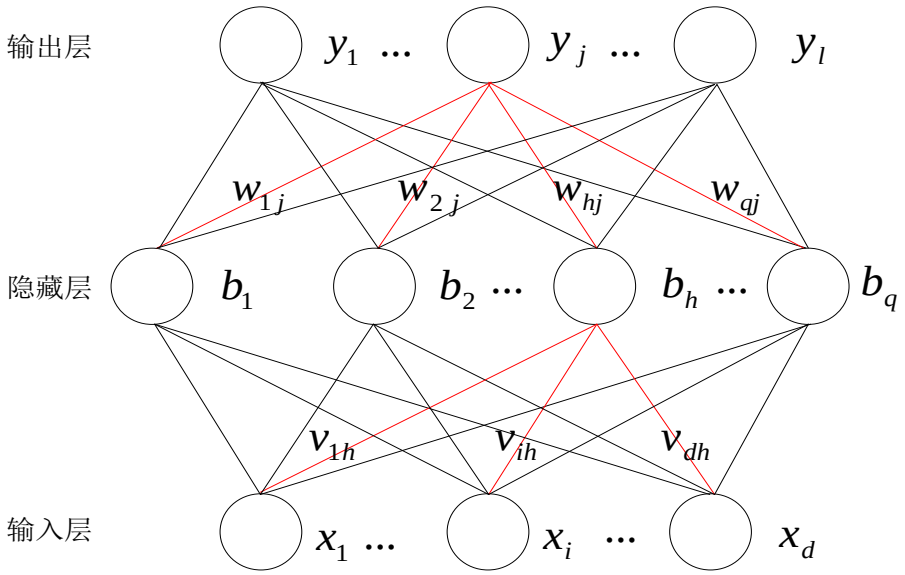


图 2.3 BP 网络以及算法符号

图 2.3 为一个简单的多层前馈网络模型，其包含输入层、隐藏层以及相应的输出层。d 表示输入神经元数量、l 表示输出神经元数量、q 表示隐层神经元的数量，使用 θ_j 和 γ_h 分别表示输出层第 j 个神经元以及隐层第 h 个神经元的阈值，这里 v_{ih} 表示连接输入层第 i 个与隐层第

h 个神经元的连接权重，w_{hj} 表示隐层第 h 个神经元与输出层第 j 个神经元之间连接权重。因此隐层第 h 个神经元所接收到的总的输入可表达为以下加权和形式：

$$\alpha_h = \sum_{i=1}^{d} v_{ih} x_i \tag{2.21}$$

相应地，输出层第 j 个神经接收到的输入为

$$\beta_j = \sum_{h=1}^{q} w_{hj} b_h \tag{2.22}$$

其中 b_h 表示隐层第 h 个神经元的输出，这里选用 sigmoid 函数作为神经元的激活函数。给定训练集 $D=\{(x_1,y_1),(x_2,y_2),...,(x_m,y_m)\}$，$x_i \in R^d$，$y_i \in R^l$，输入为 d 维，输出为 l 维。对于训练样本(x_k,y_k)，其对应的神经网络的输出为：

$$\hat{y}_j^k = f\left(\beta_j - \theta_j\right) \tag{2.23}$$

则网络在(x_k,y_k)上的均方误差为

$$\mathrm{E}_k = \frac{1}{2}\sum_{j=1}^{1}\left(\hat{y}_j^k - y_j^k\right)^2 \tag{2.24}$$

BP 算法根据梯度下降策略，以目标函数的负梯度方向对参数进行更新。对参数 w_{hj} 的更新如下：

$$\Delta w_{hj} = -\eta \frac{\partial \mathrm{E}_k}{\partial w_{hj}} \tag{2.25}$$

其中 η 为学习率，E_k 表示第 k 个样本的误差，根据链式法则：

$$\frac{\partial \mathrm{E}_k}{\partial w_{hj}} = \frac{\partial \mathrm{E}_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}} \tag{2.26}$$

令 $g_j = -\frac{\partial \mathrm{E}_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j}$，即：

$$\begin{aligned} g_j &= -\frac{\partial \mathrm{E}_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \\ &= -\left(\hat{y}_j^k - y_j^k\right) f'\left(\beta_j - \theta_j\right) \\ &= \hat{y}_j^k\left(1-\hat{y}_j^k\right)\left(y_j^k - \hat{y}_j^k\right) \end{aligned} \tag{2.27}$$

其中 $\frac{\partial \beta_j}{\partial w_{hj}} = b_h$，这样即可得到 BP 算法中的关于 w_{hj} 的更新公式

$$\Delta w_{hj} = -\eta g_i b_h \tag{2.28}$$

类似可得到：

$$\Delta\theta_j = -\eta g_i ,\quad \Delta v_{ih} = -\eta e_h x_i ,\quad \Delta\gamma_h = -\eta e_h \tag{2.29}$$

其中

$$\begin{aligned} e_h &= -\frac{\partial \mathrm{E}_k}{\partial b_h}\cdot\frac{\partial b_h}{\partial \alpha_h} \\ &= -\sum_{j=1}^{l}\frac{\partial \mathrm{E}_k}{\partial \beta_j}\cdot\frac{\partial \beta_j}{\partial b_h} f'(\partial_h - \gamma_h) \\ &= \sum_{j=1}^{l} w_{hj} g_i f'(\partial_h - \gamma_h) \\ &= b_h(1-b_h)\sum_{j=1}^{l} w_{hj} g_i \end{aligned} \tag{2.30}$$

BP 算法流程如算法 2.1 所示，其对网络参数的更新过程如下：首先将样本数据 x_k 输入神经网络，通过将数据层层向前传播，得出网络的输出结果 $\hat{y}_k$；然后计算样本真实标签 y_k 与网络输出值的误差 $\hat{y}_k$，将误差层层反向传播至隐藏层神经元，计算隐层神经元的误差，最后计算误差对参数的梯度，利用梯度下降算法来对神经元之间的连接权重和内部阈值进行更新。循环迭代以上过程，直到网络误差达到一定阈值或者训练达到一定次数为止。

算法 2.1　BP 算法

输入：训练集 $D=\{(x_k,y_k)\}$ $k\in[1,m]$；学习率 η.

过程：

1. 随机初始化网络中所有的权重和阈值
2. *do*
3. 　*for all*　$(x_k,y_k)\in D$　*do*
4. 　　根据当前参数和公式(2.23)计算当前样本的输出 $\hat{y}_k$;
5. 　　根据公式(2.27)计算输出层神经元的梯度 g_i;
6. 　　根据公式(2.30)计算隐层神经元的梯度 e_h;
7. 　　根据公式(2.28)与(2.29)跟新权重 w_{hj},v_{ih} 与阈值 θ_j 和 γ_h
8. 　*end for*
9. *until* 达到停止条件

输出：参数确定的神经网络

2.2.3 深度学习

理论上，模型参数的数量与模型的学习能力成正比，但复杂模型的训练效率通常较低，容易导致过拟合，有时候效果反而没有简单模型好。但计算机技术的发展使网络训练效率得到提高，同时，大量训练数据的轻易获取，使得过拟合的风险大大降低。以卷积神经网络为代表的深度学习技术在各个领域取得了显著的成就，典型的深度学习模型就是层数非常深的神经网络。通过模型的网络层数，能够增加模型的复杂度，提高模型的学习能力。然而多隐

层的神经网络在网络训练方面比较困难，因为误差在隐层内进行传播时，往往会导致损失值难以收敛，因此难以直接使用 BP 算法来训练。在实际中，通常采用 BP 算法与其他方法相结合的方式对多隐层网络进行训练。

在典型的神经网络-卷积神经网络（CNN）中，为了节省训练开销，采用了“权重共享”的策略，即让一组神经元使用相同的连接权重。图 2.4 为使用 LeNet5[48]进行手写数字识别任务，将尺寸大小为32×32的手写数字图像输入网络，网络输出对该图像的识别结果。该网络通过使用多个“采样层”以及“卷积层”对输入图像提取特征，然后在全连接层输出识别的结果。卷积层通过卷积滤波器来提取输入数据的特征，每个卷积层包含了多个由神经元构成的特征平面。该模型中的第一个卷积操作由5×5大小的卷积滤波器构成，生成 6 张大小为28×28的特征图。采样层也称为“汇合”层，其作用是基于局部相关性原理对输入特征进行下采样，从而在减少数据量的同时保留有用的信息。第一个采样层由大小为2×2的滤波器构成，生成 6 个大小为14×14的特征图。重复上述过程，LeNet5 将原始图像映射为 120 维的特征向量，最后通过一个由 84 个神经元构成的全连接层，完成网络的识别过程。LeNet5 使用 BP 算法进行训练，但由于其采用了权重共享思想，因此其可以大幅减少学习的参数。

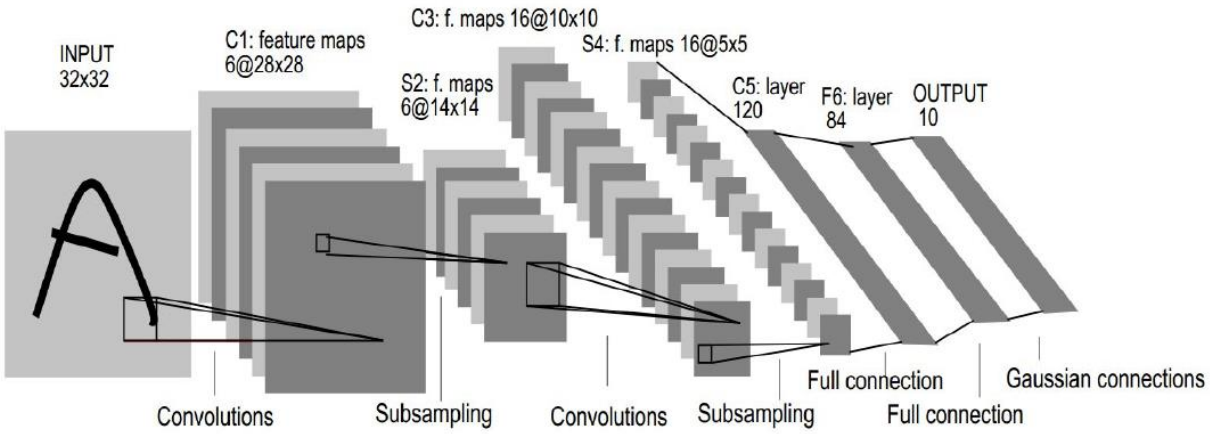


图 2.4 CNN 网络结构图

深度网络模型通常还会加入一些 ReLU[49]、Dropout[50]以及 Batch Normalization[51]等隐藏层，通过对多个隐层进行堆叠，实现了对输入信息的逐层加工，从而将低层特征表示转化为高层的特征表示，通过使用一些结构简单的模型，即可完成一些复杂的任务。这种强大的对事物的感知与表达能力使深度学习在图像分类、目标检测、语音识别等领域发挥着越来越重要的作用。

2.3 深度强化学习

2.3.1 深度学习与强化学习的结合

传统的强化学习，具有完善的理论模型，算法具有通用性，但其存在训练效率低，难以处理高维数据等缺陷。而深度学习通过堆叠多层的网络结构和一系列非线性变换，能够从原

始高维输入数据中逐层抽象，提取数据的高层表征，对数据具有强大的抽象能力，具有完整的训练机制，提供了一种对最优化问题的近似求解方法，在很多应用中能够达到端到端的效果。

深度学习侧重于对事物的感知与表达，而强化学习更加侧重于学习解决问题的策略。深度强化学习有机整合深度学习的感知能力和强化学习的决策能力，既能够利用深度学习自动学习大规模输入数据的信息，又能够利用强化学习根据这些信息为基础进行决策优化，是一种端到端的感知与控制系统，具有很强的通用性。谷歌的 DeepMind 团队将具有感知能力的 DL 和具有决策能力的 RL 相结合，创新性地提出了深度强化学习模型，即 DQN 模型，开创了深度强化学习领域研究[52]。下面将对 DQN 的相关原理进行介绍。

2.3.2 DQN 模型

DQN 模型架构如图 2.5 所示，DQN 仅使用原始视频图像信息作为输入，在 Atari[31]上取得了超越人类水平的成绩，遥遥领先于传统强化学习算法，引起了研究人员对深度强化学习极大的兴趣。随后基于 DQN 网络的改进版本纷纷被提出，在各个领域均取得了不错的效果。

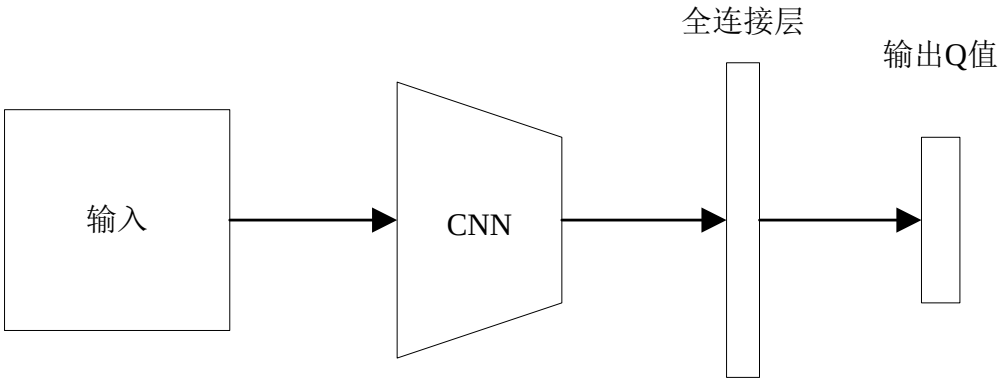


图 2.5 DQN 网络模型图

本文在目标检测任务中引入了 DQN 网络，来学习对候选区域搜索策略，下面将对 DQN 基本相关理论进行阐述。

在 agent 与环境 E 也就是 Atari 模拟器交互的任务中，会进行一系列的行动、观察和奖励。在每个时间步骤，agent 从行动集合 $A=\{1,...,K\}$ 中选择一个行动。该动作被传递给模拟器并改变其内部状态和游戏得分。通常情况下环境是随机的，模拟器的内部真实状态不被 agent 观察到;agent 从模拟器观察到的只是一个代表当前屏幕的原始像素矩阵。在交互过程中，agent 会从模拟器中接收到代表游戏得分变化的奖励 r_t。一般来说，游戏得分可能取决于之前的整个行动和观察序列，关于某个行动的反馈可能经过数千个步骤才能体现出来。

由于 agent 只观察当前屏幕上的图像，也就是说其观察到的信息是对模拟器内部状态的部分描述，即仅从当前屏幕 x_t 中不可能完全理解当前状态。因此，DQN 依赖由行动和观察构成的序列 $x_1,a_1,x_2,...,a_{t-1},x_t$ 来学习整个游戏的策略。DQN 假定所有序列在仿真器中经过有限

数量的时间步骤后都会被终止，这种定义可将问题看作一个有限长的马尔可夫决策过程(Finite Markov Decision Process, FMDP)，其中每个序列是一个不同的状态。因此，我们只需使用完整序列 s_t 作为时间 t 的状态表征，就可以使用标准强化学习方法来解决该 MDP 问题。

Agent 的目标是通过选择相应的行动与环境进行交互来最大化将来接收到的回报。DQN 做了一个标准的假设，未来每一个时间不说获得的即时奖赏都要乘以一个折扣系数 γ，则从 t 时刻开始到情节结束时所获得的奖赏之和定义为：

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \tag{2.31}$$

其中 T 表示游戏在达到时间步 T 时终止。定义最优的行动-价值函数 $Q^*(s,a)$，即任何可实现的最大预期回报的策略：

$$\begin{aligned} Q^*(s,a) &= \max_\pi \mathrm{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots \mid s_t = s, a_t = a, \pi\right] \\ &= \max_\pi \mathrm{E}\left[R_t \mid s_t = s, a_t = a, \pi\right] \end{aligned} \tag{2.32}$$

其中 π 是将序列映射到行动分布的一种策略。最优的行动-价值函数服从贝尔曼方程（Bellman Equation），如果序列 s' 下一个时间步的最优价值对所有行动 a' 来说是已知的，那么最优策略是选择动作 a' 最大化期望值 $r + gQ^*(s',a')$：

$$Q^*(s,a) = \mathrm{E}_{s'\sim\varepsilon}\left[r + \gamma \max_{a'} Q^*(s',a') \mid s,a\right] \tag{2.33}$$

强化学习算法主要依据 Bellman 方程来对算法进行迭代更新：

$$Q_{i+1}(s,a) = \mathrm{E}\left[r + \gamma \max_{a'} Q_i(s',a') \mid s,a\right] \tag{2.34}$$

当 $i \to \infty$ 时，上式迭代收敛到最佳的行动价值函数，即 $Q_i \to Q*$。但在实践中，研究人员通常使用一些函数近似器对动作-价值函数进行近似估计，即 $Q(s,a;\theta) \approx Q^*(s,a)$，该近似器通常是一个线性函数近似器，但有时使用非线性函数作为近似器，例如神经网络。DQN 使用权重为 θ 的神经网络函数作为逼近器，称为 Q 网络。一个 Q 网络可以通过最小化决策序列的损失函数 $L_i(\theta_i)$ 来迭代训练：

$$L_i(\theta_i) = \mathrm{E}_{s,a\sim\rho(\cdot)}\left[\left(y_i - Q(s,a;\theta_i)\right)^2\right] \tag{2.35}$$

其中 y_i 是迭代的目标，定义如下：

$$y_i = \mathrm{E}_{s'\sim s}\left[r + \gamma \max_{a'} Q(s',a';\theta_{i-1}) \mid s,a\right] \tag{2.36}$$

其中 $\rho(s,a)$ 是序列状态 s 到行动 a 的一种概率分布，即行动分布。在优化损失函数 $L_i(\theta_i)$ 时，需要将上一次迭代的参数 θ_{i-1} 保持固定。与监督学习不同的是，优化过程中的目标值取决于网络权重;而监督学习在学习开始之前必须将学习的目标固定。对损失函数关于参数求导可以得到参数的梯度如下：

$$\nabla_{\theta_i} L_i(\theta_i) = \mathrm{E}_{s,a\sim p(\cdot);s'\sim\varepsilon}\left[\left(r+\gamma \max_{a'} Q(s',a';\theta_{i-1}) - Q(s,a;\theta_i)\right)\nabla_{\theta_i} Q(s,a;\theta_i)\right] \quad (2.37)$$

通常使用随机梯度下降算法来优化损失函数，如果在每个时间步骤都对权重进行更新，那么求期望操作就被行动分布中的单个样本取代，此时算法退化到常见的 Q 学习算法。需要注意的是，该算法是无模型的：它直接使用仿真器 E 中的样本解决了强化学习任务，而没有直接构建对 E 的估计。该模型也是无策略的：它学习贪婪策略 $a=\max_a Q(s,a;\theta)$，同时遵循一个行动分布，能够保证对状态空间足够的探索，平衡强化学习中探索与利用的难题。在实践中，行为分布通常使用 ε 贪心策略来进行状态分布的选择，即以概率 1-ε 来根据贪心策略选择行动，以概率 ε 随机选择一个行动。

计算机视觉和语音识别方面的最新突破主要建立在使用非常大的训练集来有效地训练深度神经网络。最成功的方法是直接使用原始输入数据，基于随机梯度下降对网络进行微量更新。通过向深度神经网络输入足够的数据，通常可以学习比手工设计还要好的特征。DQN 的目标就是将强化学习算法与深度神经网络结合起来，以原始 RGB 图像作为输入数据，并通过随机梯度下降算法来进行有效地训练。其流程图如图 2.6 所示。

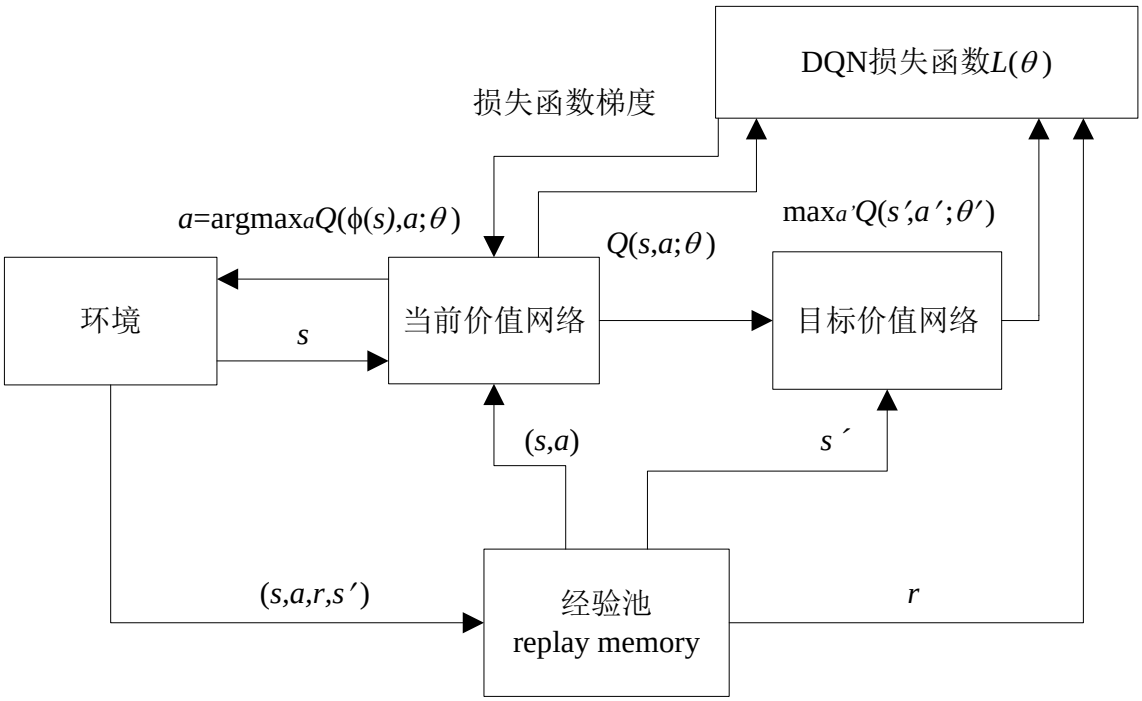


图 2.6 DQN 网络流程图

DQN 利用了一种称为经验回放(experience replay)的技术，在数据集 $D=e_1,...,e_N$ 每个时间步存储 agent 的经验，即: $e_t=(s_t,a_t,r_t,s_{t+1})$ 存储到经验池 replay memory 中。算法内部的循环中，从经验池存储的样本中随机采样一些经验，根据这些经验，应用 Q-learning 对模型进行更新。更新之后，agent 根据 ε 贪心算法选择和执行一项行动。DQN 算法流程如算法 2.2 所示：

算法 2.2 DQN 算法流程

初始化经验池 D,容量为 N

随机初始化行动-价值函数 Q

for episode=1, M *do*

初始化序列 $s_1=\{x_1\}$并进行预处理 $\phi_1=\phi(s_1)$

for t=1, T do

以概率 ε 选择随机行动 a_t

以 1-ε 的概率选择 $a_t=\max_a Q^*(\phi(s_t),a;\theta)$

在模拟器中执行行动 a_t,并观察奖赏 r_t 以及图片 x_{t+1}

这是 $s_{t+1}=s_t,a_t,x_{t+1}$ 并进行预处理 $\phi_{t+1}=\phi(s_{t+1})$

将这些过渡样本 $(\phi_t,a_t,r_t,\phi_{t+1})$ 存储到 D 中

从 D 中随机采样一批样本:

设置目标 $y_j=\begin{cases} r_j & \text{当}\phi_{j+1}\text{为终止序列} \\ r_j+\gamma\max_{a'}Q(\varphi_{j+1},a';\theta) & \text{当}\phi_{j+1}\text{ 非终止序列} \end{cases}$

根据公式(2.37)对 $(y_i-Q(\phi_{j+1},a';\theta))^2$ 使用梯度下降算法进行更新.

end for

end for

DQN 相较于标准在线 Q 学习有以下优点。首先，每一步经验可能用于许多权重的更新，对数据的重复使用可以提高数据的利用效率；其次由于连续样本之间的强相关性，直接使用连续样本进行学习效率低下，而通过对样本随机采样能够打破了这些相关性，因此减少了更新过程中参数的方差；最后，当学习策略时，当前模型的参数决定下一次更新参数时所使用的数据样本。例如，如果当前参数选择的动作是向左移动，那么训练样本将由左侧的样本支配;如果当前参数选择的动作切换到右侧，然后训练样本分布也将随之切换。这样很容易会产生不必要的反馈环路，参数可能会陷入局部极小值，或者甚至发生灾难性的发散。通过使用经验重放，将行动的分布在过去的许多经验数据基础上均匀化，能够使得训练过程更加平稳，避免了训练过程中参数产生严重波动或偏离。按照经验重放学习时，由于目前的参数与用于生成样本的参数并不相同，因此选择 Q 学习来学习离策略算法。在实践中，仅存储经验重放中的最后 N 个经验元组，并从 D 中随机采样有限数量的样本进行更新训练。这种方法是有一定的局限性的，这是因为存储器大小是有限的，且内存缓冲区没有区分重要的经验，因此旧的经验总是被最新的经验说覆盖。均匀采样赋予经验重放中所有经验相等的重要性，采用复杂的抽样策略对参数更新贡献最多的经验赋予更大的重要性，将会大大提高算法的性能。

在 DQN 网络之后，很多基于 DQN 扩展的深度增强学习算法被提出来，如深度双 Q 网络

DDQN[53]，Dueling DQN[54]以及深度循环 Q 网络 DRQN[55]等，目前基于策略的深度强化学习有 A3C 算法[56]、UNREAL 算法[57]等。由于 DQN 是深度强化学习的经典算法，具有良好的性能与通用性，因此，本文主要基于 DQN 进行目标检测的研究。

2.4 本章小结

本章从视觉目标检测算法所使用的深度强化学习技术出发，首先介绍了强化学习的相关理论基础，着重介绍了马尔可夫决策过程，以及构成强化系统的组成部分；其次介绍了深度学习，对神经元模型的构成以及训练深度网络的反向传播算法进行了描述；最后对如何将深度学习与强化学习进行结合，即深度强化学习做了相关分析，并介绍了本文所使用到的 DQN 网络的相关原理，为接下来的相关实验打好理论基础。

第3章 联合边框回归的深度强化学习目标检测算法

3.1 引言

近些年来，随着计算机性能的大幅提高以及对大量数据的轻易获取，卷积神经网络 CNN 在各领域中逐渐得到广泛应用。基于神经网络的目标检测算法在检测准确度上取得了显著的提高，基于候选区域的算法如 Faster R-CNN 等，通过使用候选区域生成算法得到高质量的候选区域，然后对这些候选区域进行一系列后续处理，最终完成对目标的检测。由于基于候选区域的目标检测算法通常需要处理大量冗余的候选区域，因此在速度上仍然存在着提升的空间。

2013 年 DeepMind 在 Atari 上的成功展示了深度强化学习在决策控制领域的巨大优势，研究人员在如何将深度强化学习引入目标检测任务，以减少所需处理的候选区域数量方面做了大量的研究。Mathe 等人[34]提出了一种序列模型，能够通过累积从一小部分图像位置上收集的信息来有效地检测视觉目标。将序列化搜索过程转化为强化学习中的策略搜索过程，可以有效平衡强化学习中的探索与利用难题，其检测速度能够比滑动窗口速度提高两个数量级。Caicedo 等人[35]提出了一种基于深度强化学习目标定位算法，该算法将整幅图片看作一个环境，通过引入一个智能体（agent）来学习对边界框进行自顶向下的搜索策略，该 agent 能够根据学习到的策略对边界框执行一系列简单的变形行动，最终将目标准确定位出来。Bueno 等人[36]提出了一种基于分层的深度强化学习目标检测框架，该框架主要思想是根据收集的线索，不断将注意力聚集在包含更多信息的区域，通过训练一个 agent 根据当前收集的信息，从预定义的五个候选子区域中选择最有可能包含目标的区域，极大的提高检测的速度，但在精确度方面仍存在不足。不同目标之间存在着某种相互关联，利用这种关联能够促进更有效的搜索，基于该思想，Kong 等人[37]提出了一种基于协同深度强化学习来进行不同目标的联合搜索算法，该算法将每个检测器看作一个 agent，使用基于多 agent 的深度强化学习算法来学习协同目标定位的最优策略，通过利用这些关联性信息，能够有效提高目标定位的准确度。

基于强化学习的目标检测算法根据收集到的信息执行相应地区域探索策略，能够显著减少所需处理的候选区域数量，但存在精确度较低的缺陷。为了提高强化学习目标检测的精确度，本文引入 ROI 回归，研究了 ROI 回归网络和 DQN 网络的联合优化问题，并利用经验池优选训练数据，改善网络训练效率。通过对 DQN 搜索到的候选区域作进一步的微调，以达到提高目标检测精确度的目的。

3.2 联合回归网络的深度强化学习目标检测

基于深度学习的目标检测算法通常依赖大量的候选区域，对这些候选区域的处理过程成

为提高检测速度的瓶颈，而基于强化学习的目标检测算法通过对候选区域进行选择性地搜索，能够显著减少所需处理的候选区域数量，但由于其主要按当前候选区域的尺寸按照一定比例进行区域搜索，存在精确度较低的缺陷。ROI 回归网络能够根据从当前区域所提取的信息，推测目标整体所在图像中的位置分布，通过对当前区域进行边框回归，达到精确定位的目的。因此本文在基于深度强化学习的目标检测框架上，引入 ROI 回归，通过 DQN 网络与 ROI 网络相融合，以提高目标检测的准确度。

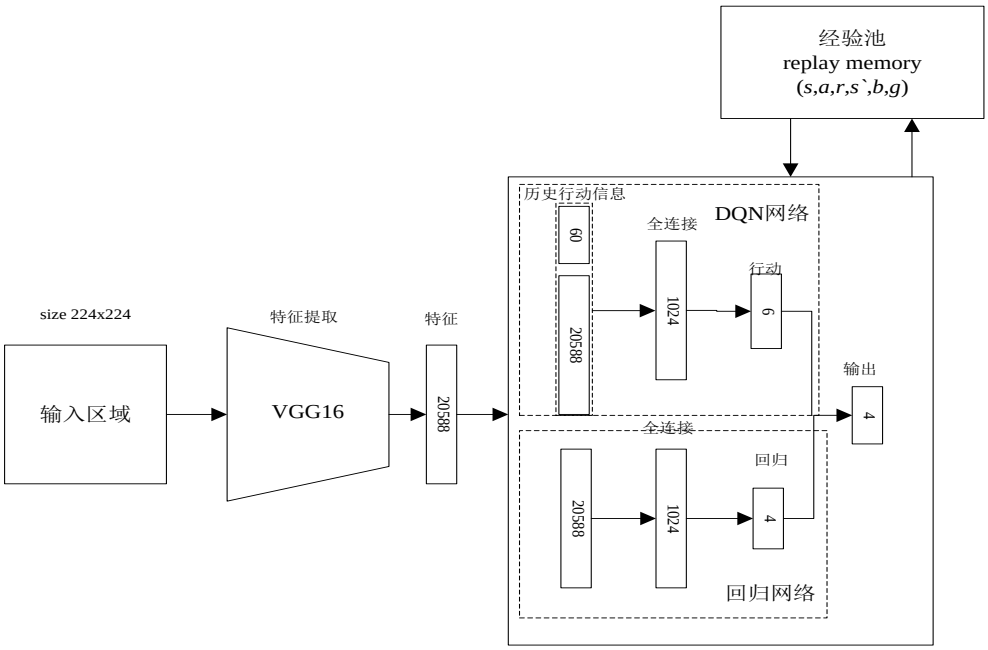


图 3.1 算法流程图

如图 3.1 所示，算法模型的框架主要由三部分组成，特征提取网络，DQN 网络，以及 ROI 回归网络，其中特征提取网络为预先训练好的 VGG[58]网络。首先由预训练的 VGG 网络对输入图像区域进行特征提取；其次将提取的特征向量送入 DQN 网络，DQN 网络负责确定搜索的路径;最后当 DQN 网络终止搜索时,ROI 回归网络根据特征向量对候选区域进行边框回归，输出最终检测的结果。DQN 网络的训练需要经验池存储大量的经验样本，而 ROI 回归网络的训练则需要大量的 IoU 大于一定阈值的经验样本；DQN 网络着重解决区域探索策略问题，而 ROI 回归网络则主要提高候选区域的准确度。两网络的训练数据与优化目标均不相同，为了对 ROI 回归网络和 DQN 网络进行联合优化，本文做了以下的工作。

3.2.1 MDP 建模

强化学习提供了一种正式的框架来解决智能体（agent）在环境中采取何种策略来最大化累积奖赏的问题。将整幅图像看做是一个环境，agent 通过行动集合中的行动来对边界盒进行变形，其目标是使边界盒将目标紧紧包围起来，算法模型如图 3.1 所示。为了构建一个完整的强化学习系统，针对目标定位任务，定义了具体的行动，状态，以及奖赏函数等系统组成部分。

行动集合 *A*: agent 为了达到其目标所能采取的行动集合。本文定义了九个对候选区域进行搜索的行动，其中包含八个对候选区域进行变形的行动，以及一个终止搜索的行动。行动集合的具体定义如下：*A*:{向右，向左，向上，向下，变大，变小，变宽，变高，终止}，每个行动根据当前边界盒的尺寸大小，按照一定的比例 α 对其尺寸进行一个离散的变化，终止的行动更代表 agent 认为已经找到目标。

状态集合 *S*：代表了 agent 对当前环境的信息的理解，状态定义为一个元组：*s*:=(*o*,*h*)。*o* 是一个当前观察区域的一个特征向量，该特征向量由一个预训练的 CNN 网络进行提取，而 *h* 是一个固定大小的向量，代表 agent 曾采取的行动历史。将向量 *o* 与 *h* 连接起来送入一个 DQN 网络，输出为一个维度为 9 的向量，代表九个行动。

以及一个奖赏函数 *R*：代表着环境对选择的行动该状态下好坏的评判，用来指导 agent 根据当前状态学习最优策略。当 agent 采取行动 *a*，由状态 *s* 进入下一个状态 *s'* 时，环境给予 agent 的奖赏 *R*(*a*,*s*-> *s'*)，奖赏信号定义了在当前状态下对 agent 来说什么是逼近目标的行动，什么是远离目标的行动，也即当前状态下所采取的行动是否有助于目标的定位。奖赏函数的定义如下式所示。

$$R_a(a, s \to s') = sign(IoU(b', g) - IoU(b, g)) \tag{3.1}$$

其中 IoU 是目标 *g* 与边界盒 *b* 之间的交并比：

$$IoU(b, g) = area(b \cap g) / area(b \cup g) \tag{3.2}$$

其中 area()为求面积函数。对于终止行动，其奖赏函数如下。

$$R_t(s \to s') = \begin{cases} +\eta \ \ if \ \ IoU(b,g) \geq \tau \\ \\ -\eta \ \ otherwise \end{cases} \tag{3.3}$$

根据行动集合，状态集合以及奖赏函数，能够应用 Q-learning 来学习最优策略，为了处理高维数据，引入 DQN 网络来近似 Q(*s*,*a*)，智能体 agent 根据 *Q* 函数选择具有最高 *Q* 值的行动，使用贝尔曼方程来对 *Q* 函数进行迭代更新:

$$Q(s, a) = r + \gamma max_a Q(s', a') \tag{3.4}$$

其中 *s* 为当前状态，*a* 为当前选择的行动，*r* 为即时奖赏，γ 代表折扣系数，*s'*代表下一状态，*a'*代表接下来采取的行动。通过建立经验池(replay memory)存储其更新过程需要使用经验数据(*s*,*a*,*r*,*s'*)。采取终止行动后，为了对多个目标进行检测，算法中还应用了返回抑制机制(Inhibition-of-Return)。最后，应用一个预训练的 SVM 分类器对 agent 所探索的区域进行分类。

根据当前区域特定的比例来对边界框进行变形，是一种比较粗糙的目标定位方式，对于一些尺寸大小的目标很难完全覆盖到，因此在精确度方面仍存在着提升的空间。

3.2.2 损失函数

为了对 DQN 网络与 ROI 回归网络进行联合优化，本文将损失函数设定为多任务损失函数，即 DQN 损失函数与回归网络损失的加权和。其中对于 DQN 网络本文采用均方误差损失函数，而回归网络的损失函数采用鲁棒性较强的 *smoothL*$_1$ 损失函数。整体损失函数的定义如下所示。

$$L(s,a,t)=\frac{1}{N_{dqn}}\sum_i\left(y_i-Q(s_i,a)\right)^2+\lambda\frac{1}{N_{reg}}\sum_i R\left(t_i-t_i^*\right) \tag{3.5}$$

其中 i 是该样本在最小批数据样本集中的索引值，参数 y_i 代表第 i 个样本下 DQN 网络的输出，$Q(s_i,a)$代表目标输出，N_{dqn} 代表输入 DQN 网络的样本数，而 N_{reg} 代表送入回归网络的样本数，λ 是一种加权系数，用来平衡 DQN 网络损失与回归网络损失，函数 $R(t_i\text{-}t_i*)$代表回归损失，其中 R 函数即为平滑的 L_1 损失函数:

$$smooth_{L1}(x)=\begin{cases}0.5x^2 & if\ |x|<1\\ |x|-0.5 & otherwise\end{cases} \tag{3.6}$$

t 是维度大小为 4 的向量，代表参数化的坐标，即 $t=(t_x,t_y,t_w,t_h)$，具体的，本文将一般的坐标 $b=(x,y,w,h)$进行如下的参数化：

$$t_x=(x-x_a)/w_a\quad t_y=(y-y_a)/h_a\quad t_w=\log(w/w_a)\quad t_h=\log(h/h_a) \tag{3.7}$$

$$t_x^*=(x^*-x_a)/w_a\quad t_y^*=(y^*-y_a)/h_a\quad t_w^*=\log(w^*/w_a)\quad t_h^*=\log(h^*/h_a) \tag{3.8}$$

参数 x、y 分别代表回归网络输出的候选区域的中心点，w 和 h 为其宽和高；x_a、y_a *分别代表* DQN 网络得到的候选区域的中心点，w_a、h_a 为其宽和高，x^*,y^*分别代表真是目标区域的中心点，w^*,h^*为其宽和高。

3.2.3 模型训练

本文 DQN 网络与 ROI 回归网络的架构相同，下面将针对该回归网络与 DQN 网络之间的联合训练进行具体的阐述。

在构建经验池并定义好损失函数之后，针对 DQN 网络与回归网络的联合训练，本文采取以下策略进行：

1）为了平衡 DQN 网络的探索与利用难题，本文使用了 ε 贪心算法(ε-greedy policy)，每次训练以概率 ε 来进行行动的探索，以 1-ε 的概率利用已学习到的策略进行决策，其中 ε 的初始值为 1，然后随着 epoch 的增加，不断降低至 0.1。对于 agent 来说，终止行动的学习是比较困难的，因此为了帮助 agent 学习该行动，本文在当前区域与真实区域之间的 IoU 大于 0.6 时，强制使得其选择终止行动。

2）经验池里存放的经验为(s,a,r,s',b,g)，其中 s 为当前状态，a 为采取的行动，r 在状态 s

下执行行动 a 后所得到的立即奖赏，s'为转换到的下一个状态，b 为当前区域的坐标，而 g 代表着目标真实区域的坐标。DQN 网络与回归网络共用一个经验池，其中 DQN 网络训练时使用的部分数据是(s,a,r,s')，而 ROI 回归网络训练使用的数据是(s,b,g)。两网络的输入数据均为状态 s。

3）在对回归网络进行训练的时候，本文为了回归的准确性，仅使用目标区域与真实区域之间 IoU 大于一定阈值的经验样本送入回归网络进行训练。

对于一幅图片，初始候选区域是整个图像区域，将该候选区域的尺寸归一化为 224×224，然后送入一预训练好的 VGG 来进行特征提取，然后以概率 ε 来从合理行动集合中随机选取一个行动进行搜索，以 1-ε 的概率利用已学习到的策略进行决策。执行行动 a 后，得到的新候选区域 b'，根据公式(3.1)来赋予 agent 相应的奖赏 r，并将新的候选区域尺寸归一化为 224×224，送入特征提取网络提取特征，与历史行动向量结合，得到下一个状态 s'。重复上述过程，直至行动 a 为终止行动或者搜索步骤达到最大步骤数。若行动 a 为终止行动，则根据公式(3.3)得到终止奖赏 r_t，执行终止行动后，将将该特征送入回归网络，对候选区域进行精细定位，得到最终的定位结果。将每一步行动后后所得到的将经验信息元组(s,a,r,s',b,g)存入经验池，利用该经验池数据对整个网络进行联合训练，其中对于候选区域与真实区域之间 IoU 小于 0.4 的样本数据不参加回归网络的训练。按照公式(3.5)计算网络的损失函数，并使用随机梯度下降算法(Stochastic Gradient Descent, SGD)对两网络进行参数更新。

3.3 实验及结果分析

3.3.1 实验平台及参数设定

本文使用 Torch7 深度学习平台，在数据库 VOC2007 与 VOC2012 上进行仿真实验，其中采用 VOC2007 与 VOC2012 的训练集数据进行训练，采用 VOC2007 中的测试集来进行测试。在此本文仅考虑单一具体类别的目标进行检测。选择较大的比例值 α 时生成的候选区域很难覆盖到目标，而较小的比例会经过更长的步骤才能定位到目标，因此本文权衡之后将 α 设置为 0.2。DQN 网络使用两个全连接层，输出维度为行动数量，同时本文在网络中还加入了 Dropout 层以及 ReLU。在使用贝尔曼方程更新 Q 函数时，选用的折扣系数 γ 值为 0.9。本文经验池的大小设定为 1000，每次随机采样的最小批大小为 128，训练次数为 20 个 epoch。

3.3.2 实验结果与分析

模型训练过程中，损失函数的变化如图 3.2 所示，从图中可以看出，随着迭代次数的增加，模型的损失值在急速下降，当训练次数达到 20000 次时，网络逐渐收敛，损失值变化趋于平稳。由此可见在训练过程中模型的参数得到了更新，网络学习到了相关的定位知识。

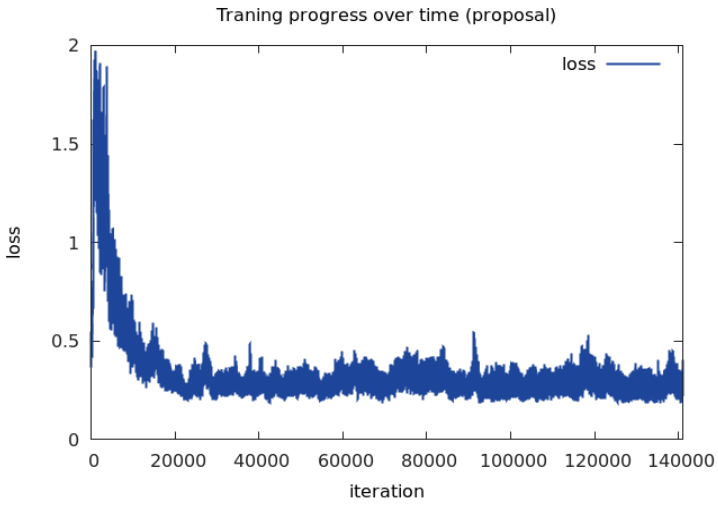
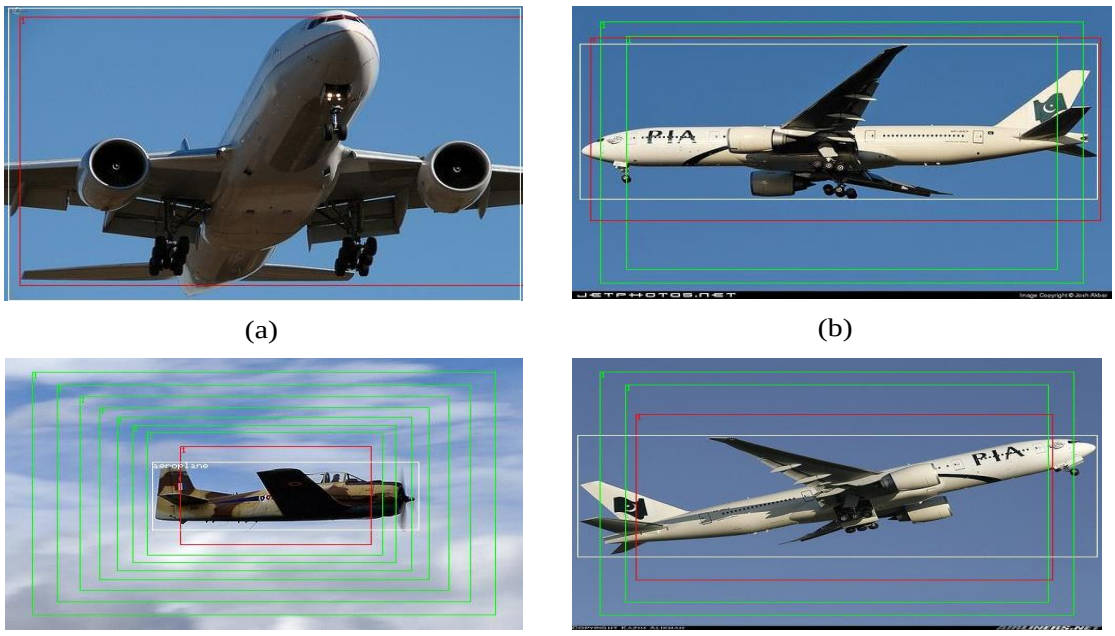


图 3.2 模型训练损失值变化

图 3.3 是在简单背景条件下对飞机类别的目标检测效果图，其中绿色框代表 DQN 网络每次产生的候选区域，红色框代表结合回归网络所得到的最终定位结果，白色框代表真实目标区域。对于正常尺寸大小的目标，如图 3.3(b)和图 3.3(d)所示，模型仅需很少的搜索步骤即可定位到飞机目标所在的位置。对于尺寸较大的目标，如图 3.3(a)所示，DQN 网络根据当前区域特征，仅需执行一次搜索行动，便能准确定位目标位置，随后通过回归网络再对目标区域进行精确定位。对于尺寸较小的目标，如图 3.3(c)所示，由于目标较小，DQN 网络便会朝着目标区域的方向进行不断的搜索，直到收集到足够的信息，才会终止搜索行动，并确定当前区域即为目标区域（如图中尺寸最小的绿色框），并由回归网络对目标位置进行更加准确地定位。

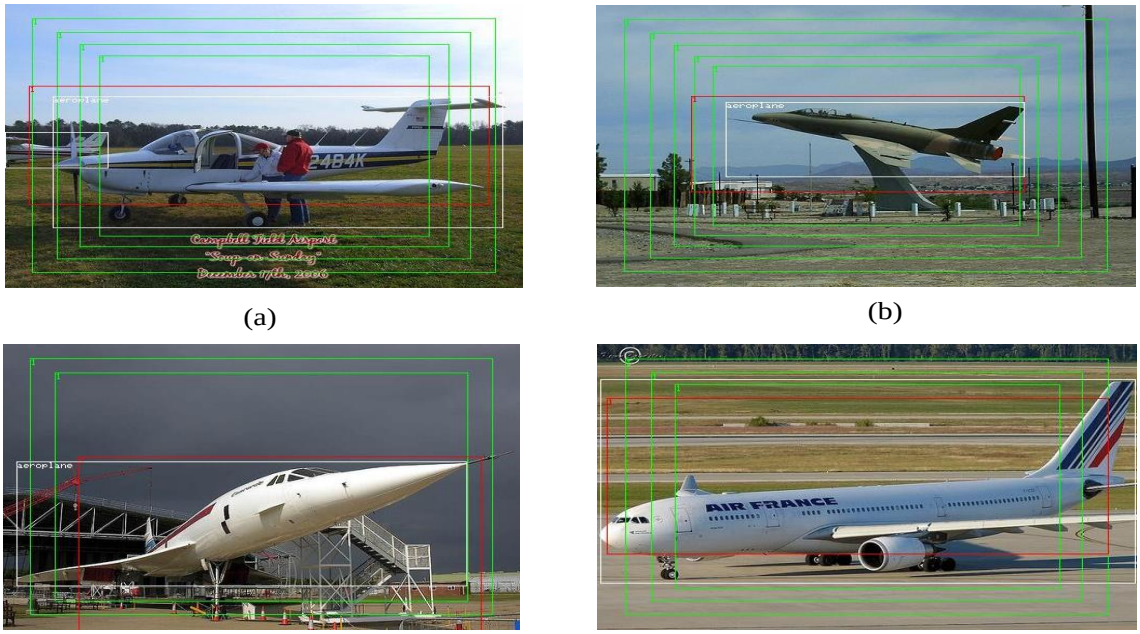


(a) (b)

(c) (d)

图 3.3 简单背景下的目标检测

图 3.4 是在复杂背景条件下对飞机类别的目标检测效果图，从图中可以看出，背景中除了经常出现的蓝天白云外，还存在建筑物、草地以及行人等多种干扰物体。传统的目标检测方法容易受这些干扰物的影响，难以精确地对目标进行定位；而本文算法通过 DQN 网络可以确定目标所在的大体位置，然后利用边框回归网络进一步对候选区域坐标进行精确定位，从而实现对复杂背景条件下的目标定位。



(a) (b)

(c) (d)

图 3.4 复杂背景条件下的目标检测

更加量化的指标如表 3.1 所示，由于 Caicedo 算法并没有公开代码，本文按照作者文中的算法流程仿真实现，但由于一些算法中的经验参数未公开等原因，导致本文实现的结果低于作者论文中的数据结果。本文的算法对比实验均在相同参数、相同数据集以及相同训练次数的条件下进行，确保对比试验的公正性。实验结果表明，本文算法在单类别的目标检测中，相较于其他基于强化学习的目标检测算法具有明显的优势，其中在 areoplane 类别中，相较于 Bueno 等人的算法在精确度方面提高了 18.03%，相较于 Caicedo 等人的算法精确度提高了 12.86%；在 dog 类别中，相较于 Bueno 等人的算法在精确度方面提高了 13.38%，相较于 Caicedo 等人的算法精确度提高了 10.99%，表明了本文算法能有效提升目标定位的精确度。

表 3.1 各算法在 Pascal voc2007 数据库上的检测准确率

算法/类别	areoplane	dog
Bueno 算法	32.23%	32.19%
Caicedo 算法	37.40%	34.58%
本文算法	50.26%	45.57%

3.4 本章小结

为了克服基于强化学习的目标检测算法中精确度较低的缺点，本文提出了将边框回归网络与 DQN 网络相融合的定位方式，首先由 DQN 网络对目标进行粗定位，然后利用边框回归网络对 DQN 网络产生的候选区域坐标进行矫正，以得到更准确的定位。在模型训练阶段，本文通过共享经验池的方式对 DQN 网络和边框回归网络进行联合优化，在简化训练过程的同时，提高数据的利用效率。实验结果表明，相比于原算法，本文算法在对单一类别目标检测中，能够有效提高其精确度。

第 4 章　基于多层特征与深度强化学习目标检测算法

4.1 引言

人们在观察一幅图片时能够立刻知道图片中目标的位置以及目标所属的类别，人类的视觉系统快速而准确，使我们不需要太多的思考就能够完成很多复杂的任务。作为计算机视觉领域最重要的一部分，快速而精准的目标检测是实现很多如机器人系统、自动驾驶以及行人检测等应用的前提。很多目标检测系统主要是基于分类的思想，为了定位目标，通常会使用一个分类器在图片中的各个位置对不同尺寸的区域进行评估，如使用不同尺寸的滑动窗口以固定步长在整幅图片上滑动，并对每次滑动产生的区域应用分类器进行分类。领先的算法如 Faster R-CNN 等，通过一个候选区域生成网络生成一些的边界盒，然后对这些盒子在特征图上进行自适应采样，生成相同维度大小的特征信息，并将这些信息送入高质量的分类器进行分类。尽管这类算法准确度很高，但是由于通常会生成大量的候选区域，因此其在检测速度上还存在着较大的提升空间。Faster R-CNN 的另一个缺陷在于，其仅使用了最后一层的特征图来进行检测，而不同层的特征对于不同尺寸的目标贡献各不同，一些较小的目标在更深的特征层中几乎无法体现出来。因此如何将多层的特征综合利用起来，也是提升算法的关键。

人类的注意机制[59]使得人们观察一幅图片时，通常会从一个初始区域出发，根据收集到的信息，对视线聚焦变换，缩小搜索的范围，直至系统收集到足够的证据确定所要搜索的目标准确位置。受该注意机制的启发，本文将目标定位建模为自上而下的序列化搜索问题。通过将候选区域划分为五个不同的子区域，从中选择对最可能是目标的区域进行探索，不断缩小注意区域的范围，直至定位该目标。从初始区域开始，可能需要经过一系列的区域选择才能最终定位到目标，考虑到强化学习在决策控制相关领域具有突出表现，因此本文引入了深度强化学习来学习该序列化决策过程。为了充分利用不同特征层的信息，本文建立一个特征-候选区域映射关系，针对从该区域提取的特征信息，模型可以决定所要执行的搜索策略。

本文的主要贡献在于，将强化学习引入到目标检测任务中，将目标检测建模为在候选区域上自顶向下不断缩小，逐渐逼近候选区域的过程。本文为了充分利用多层特征信息，将特征提取过程中的相关特征层的特征进行存储，通过建立特征-候选区域映射关系，创新性地将不同尺度大小区域对应到不同特征信息图上，深度强化学习根据当前特征来判断下一子区域的选择，使得目标检测任务中所需评估的候选区域数量呈指数级减少，极大的提高了检测的速度。为了提高定位的准确度，本文还引入了一个回归网络对定位结果进行精确定位。实验结果表明，对于 n 层特征，使用本文的方法，最多仅需 n 搜索次数即可成功定位目标。

4.2 多层特征提取

由于不同尺寸大小的候选区域在相同特征层上的表现力不同，因此为了应对较小区域在深层特征图上信息缺失的问题，本文引入多层特征，每层相同大小的特征区域对应着不同尺寸的候选区域，这样能够使得各种尺寸的候选区域的特征信息均能保留下来。

本文定义了五种搜索候选区域的行动，对应着五个子区域，以及一种终止搜索的行动，代表当前区域即为目标区域，无需再向下搜索。五个子区域的宽高均为当前区域宽高大小的3/4，对应到上一层特征图上五个尺寸大小相同，但位置不同的特征区域。本文定义的候选区域与特征区域的映射方式如图 4.1 所示。

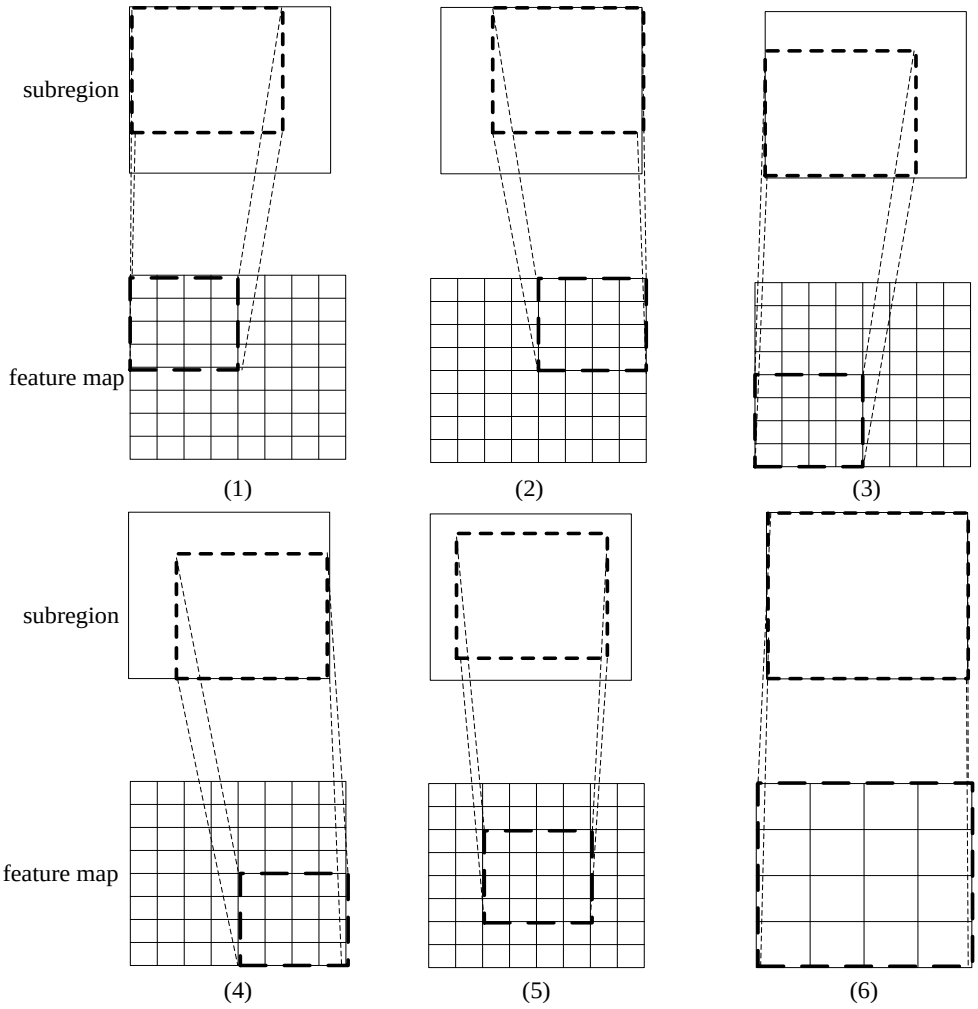


图 4.1 区域-特征映射关系

为了提取多层特征，本文构建了一个特征提取网络，该网络使用 VGG16 网络架构作为基础特征提取器。具体地，本文去掉 VGG16 最后的全连接层，在 pool5 层后再接入一个两层卷积核大小为 3×3 卷积层，并加入一个池化层 pool6，这样最后的 pool6 卷积图大小为 pool5 特征图大小的一半。在参数初始化方面，本文使用基于 ImageNet 2012 数据集进行预训练的 VGG16 模型，进行参数初始化，新加入的层使用 xavier 方式进行初始化。本文共使用三层特

征图来进行实验，包括 pool4、pool5 以及新定义的 pool6。对于每个候选区域，其对应的特征尺寸大小均为512×4×4。注意到，最后一层卷积图 pool6 尺寸大小为512×4×4，对应初始候选区域为原图区域。

4.3 强化学习建模

在决策与规划领域中，马尔可夫过程(Markov Decision Process, MDP)提供了一种简单、通用的表达方法。强化学习通常建立在马尔可夫决策过程的思想之上，马尔可夫决策过程完整地表征了强化学习的环境模型。强化学习提供了一种正式的框架来解决智能体（agent）在环境中采取何种策略来最大化累积奖赏的问题。在目标检测任务中，可将整幅图像看做是一个环境，agent 通过行动集合中的行动来对候选区域进行搜索，最终达到目标检测的目的。为了构建一个完整的 MDP 过程，针对目标定位任务，定义了具体的行动、状态以及奖赏函数等系统组成部分。

4.3.1 行动

本文定义了六种行动来对候选区域搜索过程建模，其中五种行动用来对候选区域进行搜索，一种行动用来指示搜索过程的终止。五种搜索行动对应着五个部分重叠的子区域，分别为选取左上、右上、左下、右下以及中间区域。五个子区域的宽高均为当前区域宽高大小的3/4。子区域的宽高选择为当前区域宽高的 3/4，主要是考虑到若是比例较小，搜索过程很难覆盖到各个区域，而比例过大，则会花费更长的时间才能定位目标，导致算法很难收敛，因此本文通过实验分析之后选择了一个经验值 3/4。为了使算法在检测到目标时能够及时停止搜索，本文还定义了一种终止搜索的行动，代表当前区域即为目标区域。搜索的及时停止对算法非常重要，如果搜索没有及时停止，最终的检测区域将与实际目标区域存在极大的偏差，停止搜索这意味着算法收集到足够的信息能够判别当前区域为目标区域，减少不必要的搜索步骤。行动的定义如图 4.2 所示：

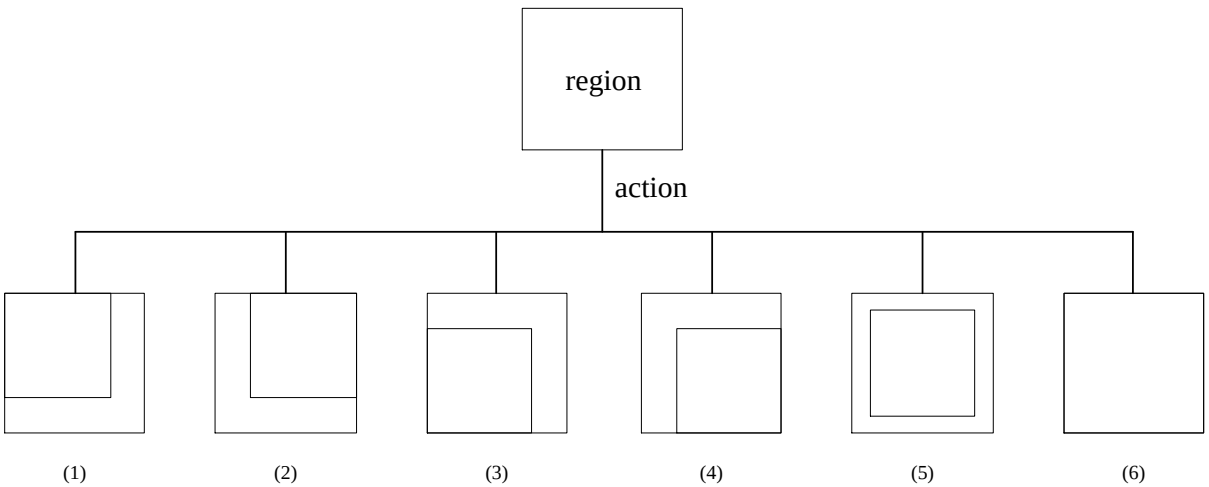


图 4.2 行动的定义

4.3.2 状态

本文的对各个区域所提取的特征大小均为$512\times4\times4$，为了对不同层的特征进行区别开来，本文在提取的特征区域所形成的特征向量上增加了一个 3 维大小的层向量，进行 *one-hot* 编码（层向量仅在当前层数的位置为 1，其他位置为 0），用来指示当前特征所属的特征层，因此本文对区域提取的信息维度大小为 8195。从最后一层特征图开始，每次的搜索，都是一个逐渐紧紧包围真实目标的过程，这种过程反映到特征图上，表现为由深层特征图向浅层特征图跃迁的过程，每个子区域对应着相应位置的特征区域。如当前候选区域的第一个子区域，即左上子区域，对应着特征图上左上4×4大小的子区域，候选区域右下子区域，对应着特征图上与左上特征子区域相同尺寸的右下子区域。本文仅使用三层特征图，因此对于一个目标，该模型最多只需要 3 步即可定位目标所在位置。

4.3.3 奖赏

奖赏代表着环境对选择的行动该状态下好坏的评判，用来指导 agent 根据当前状态学习最优策略。当 agent 采取行动 a，由状态 s 进入下一个状态s'时，环境给予 agent 的奖赏$R(a,s\rightarrow s')$，奖赏信号定义了在当前状态下对 agent 来说什么是逼近目标的行动，什么是远离目标的行动，也即当前状态下所采取的行动是否有助于目标的定位。奖赏函数的定义如下式所示。

$$R_a(a,s\rightarrow s')=sign(IoU(b',g)-IoU(b,g)) \tag{4.1}$$

其中 IoU 是目标 g 与边界盒 b 之间的交并比：

$$IoU(b,g)=area(b\cap g)/area(b\cup g) \tag{4.2}$$

4.4 深度强化学习与多层特征的融合

本文引入了一个 agent 来学习行动选择策略。agent 根据当前区域提取到的信息来决定选择哪种行动来对区域进行搜索，而执行该行动后，会产生新的子区域，不断重复上述过程，直至检测到目标为止。根据行动集合，状态集合以及奖赏函数，能够应用 Q-learning 来学习最优搜索策略，利用贝尔曼方程进行参数更新：

$$Q(s,a)=r+\gamma max_a Q(s',a') \tag{4.3}$$

其中 s 为当前状态，a 为当前选择的行动，r 为即时奖赏，γ 代表折扣系数，s'代表下一状态，a'代表接下来采取的行动。为了处理高维数据，引入 DQN 网络来近似 Q(s,a)。仅由本文所定义的对候选区域的搜索操作所得到的子区域，难以完全覆盖到所有尺寸大小的目标，因此，为了更加准确的定位目标，本文引入了一个回归网络。通过回归网络对候选区域进行精确定位，来弥补行动选择网络在应对目标尺寸变化上的不足。

4.4.1 改进的经验池

DQN 网络的训练需要经验池存储大量的经验样本，而回归网络的训练则需要大量的 IoU 大于一定阈值的经验样本；DQN 网络着重解决区域探索策略问题，而回归网络则主要提高候选区域的准确度。两网络的训练数据与优化目标均不相同，为了对回归网络和 DQN 网络进行联合优化，本文对原 DQN 所定义的经验池(s,a,r,s')进行改进，建立了一个改进的经验池来存储训练需要用到的经验数据，即(s,a,r,s',b,g)，其中 s 为当前状态，a 为采取的行动，r 在状态 s 下执行行动 a 后所得到的立即奖赏，s'为转换到的下一个状态，b 为当前区域的坐标，而 g 代表着目标真实区域的坐标。

在探索阶段，将每次行动产生的经验数据(s,a,r,s',b,g)存入经验池中。由于在探索过程中当前所采取的行动对接下来所探索的区域产生影响，导致经验池中某些行动的经验数量远大于其他行动的经验数量。因此，为了平衡每个行动的样本数据，本文对每个行动建立一个相应具有相同大小的经验池。在网络优化阶段，从每个行动对应的经验池中随机采样相同批量大小的数据，并将这些数据的顺序随机化，然后送入网络训练。

4.4.2 目标函数

为了对 DQN 网络与 ROI 回归网络进行联合优化，本文将损失函数设定为多任务损失函数，即 DQN 损失函数与回归网络损失的加权和。其中对于 DQN 网络本文采用均方误差损失函数，而回归网络的损失函数采用鲁棒性较强的 $smoothL_1$ 损失函数。整体损失函数的定义如下所示。

$$L(s,a,t)=\frac{1}{N_{dqn}}\sum_i\left(y_i-Q(s_i,a)\right)^2+\lambda\frac{1}{N_{reg}}\sum_i R\left(t_i-t_i^*\right) \tag{4.4}$$

其中 i 是该样本在最小批数据样本集中的索引值，参数 y_i 代表第 i 个样本下 DQN 网络的输出，$Q(s_i,a)$代表目标输出，N_{dqn} 代表输入 DQN 网络的样本数，而 N_{reg} 代表送入回归网络的样本数，λ 是一种加权系数，用来平衡 DQN 网络损失与回归网络损失，函数 $R(t_i\text{-}t_i*)$代表回归损失，其中 R 函数即为平滑的 L_1 损失函数:

$$smooth_{L1}(x)=\begin{cases}0.5x^2 & if\ \ |x|<1\\ |x|-0.5 & otherwise\end{cases} \tag{4.5}$$

t 是维度大小为 4 的向量，代表参数化的坐标，即$t=\left(t_x,t_y,t_w,t_h\right)$，具体的，本文将一般的坐标$b=\left(x,y,w,h\right)$进行如下的参数化：

$$t_x=\left(x-x_a\right)/w_a\ \ t_y=\left(y-y_a\right)/h_a\ \ t_w=\log\left(w/w_a\right)\ \ t_h=\log\left(h/h_a\right) \tag{4.6}$$

$$t_x^*=\left(x^*-x_a\right)/w_a\ \ t_y^*=\left(y^*-y_a\right)/h_a\ \ t_w^*=\log\left(w^*/w_a\right)\ \ t_h^*=\log\left(h^*/h_a\right) \tag{4.7}$$

参数 x、y 分别代表回归网络输出的候选区域的中心点，w 和 h 为其宽和高；x_a、y_a 分别代表 DQN 网络得到的候选区域的中心点，w_a、h_a 为其宽和高，x^*,y^*分别代表真是目标区域的

中心点，w^*,h^*为其宽和高。

4.4.3 模型架构

模型架构如图 4.3 所示，首先由经过预训练的 VGG16 网络对输入图像进行多层特征的提取，其次从最后一层特征层开始经过特征-候选区域映射产生相应的候选区域，将候选区域所对应的特征送入 DQN 网络，以得到相应的搜索行动，最后根据搜索行动决定下一个探索的子区域并由当前特征层跃迁到上一层特征层，重复上述过程，直至 DQN 选择终止行动或者没有特征层可供跃迁为止。此时将该区域对应的特征送入回归网络，通过对候选区域进行回归，以得到更为精确的定位效果。在网络训练阶段，将这些候选区域以及其对应的特征作为训练样本送入 DQN 与回归网络进行训练。

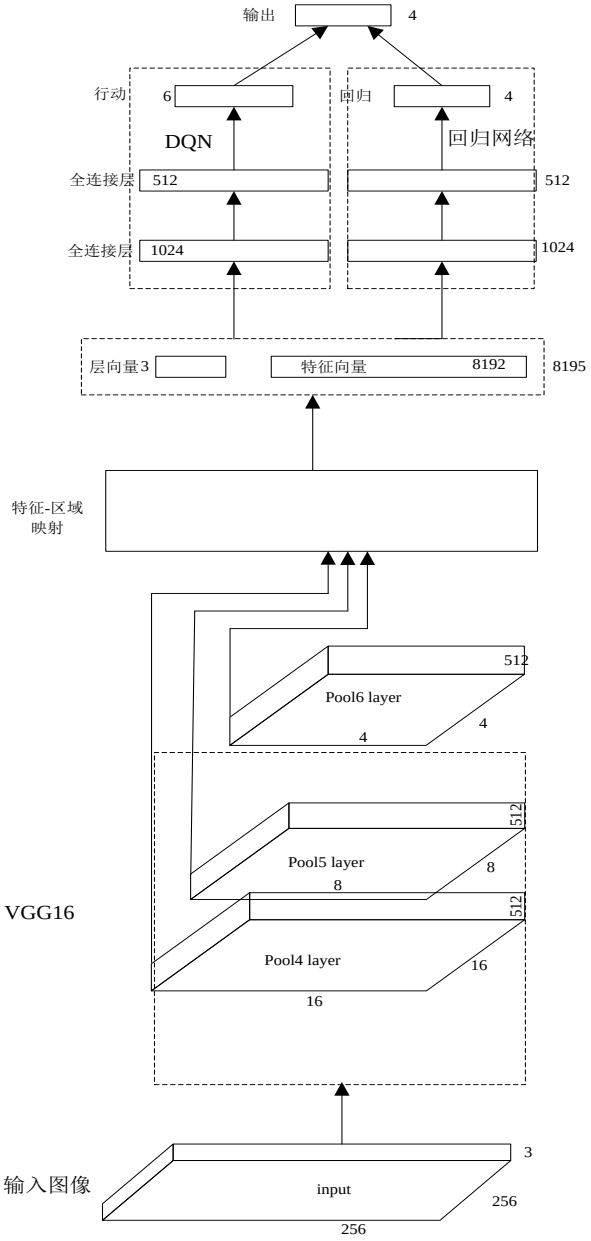


图 4.3 模型框架

为了回归网络训练过程的稳定，本文仅使用候选区域与目标区域之间的 IoU 大于一定阈值的样本对回归网络进行训练。为了确定行动选择网络何时终止搜索，本文通过计算候选区域与目标区域之间的 IoU，以及该候选区域所有子区域与目标区域之间的 IoU′，当所有子区域与真实区域之间的 IoU′均不大于当前区域与目标区域的 IoU 时，则终止搜索，即目标行动值为终止行动。当存在子区域与目标区域的 IoU 大于 IoU 时，随机选择子区域与目标区域之间 IoU′增大的行动。

对于输入的图片，本文一律将其尺寸缩放到 256×256，并将其送入初始化后的 VGG16 网络中，将所得到的 pool4,pool5 以及 pool6 卷积图进行存储下来，进行后续的搜索。初始区域为整幅图片所覆盖的区域，其对应的特征层为 pool6 层，而 pool6 层的尺寸大小为 $512\times4\times4$，将根据该特征图提取的信息向量与层向量进行连接，并送入 DQN 网络。DQN 网络根据该信息决定执行的行动，如继续搜索某个子区域，或停止搜索。但 DQN 选择了一个行动后，候选区域缩小为对应的子区域，而探索的特征层也从 pool6 跃迁到 pool5 层，此时该子区域对应的特征区域也根据对应关系，对应到 pool5 层相应的位置。重复该过程，直至行动网络选择终止搜索行动，或者算法已经跃迁到 pool4 层，无法再向下跃迁为止。此时，将当前的特征送入回归网络，对当前区域的位置进行微调，达到更精准的定位目标的目的。一个典型的搜索路径如图 4.4 所示。

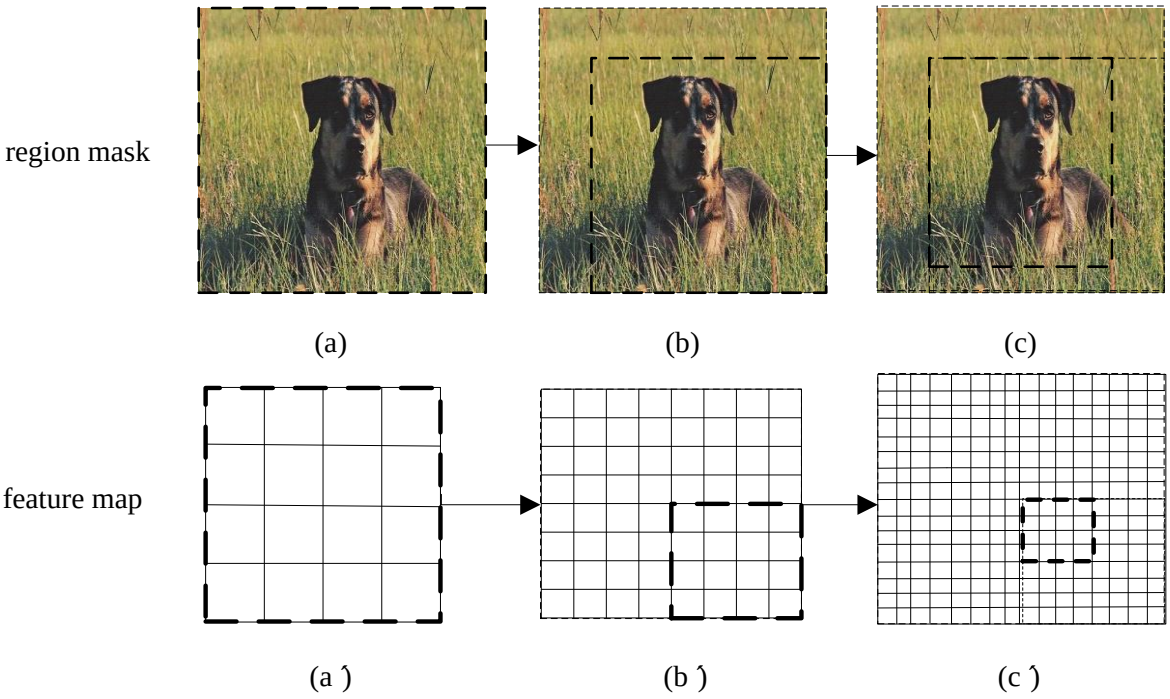


图 4.4 搜索路径示例

从图 4.4 中可以看出，初始候选区域为整幅图像区域，其对应的特征区域为最后一层特征即特征图 4.4(a′)，大小为 (4×4)，agent 根据该特征提取的信息，执行了行动 4，选择对右下角的子区域进行探索，此时候的候选区域更新为右下角的子区域如图 4.4(b)对应的区域，

而其对应的特征为特征图 4.4(b′)的右下角相同大小的特征区域。Agent 根据当前区域特征提取的信息，选择行动 1，进而决定探索的区域为当前区域的左上角子区域如图 4.4(c)，其对应的特征也改变为特征图 4.4(c′)上相应位置相同大小的特征区域。此时终止搜索，所得到的候选区域将目标紧紧包围起来，目标成功被定位。

4.5 实验仿真及结果分析

4.5.1 实验平台及参数设定

为了验证算法效果，本文使用 Torch7 深度学习平台，在数据库 VOC2007 与 VOC2012 上进行仿真实验，其中采用 VOC2007 与 VOC2012 的训练集数据进行训练，采用 VOC2007 中的测试集来进行测试。在使用贝尔曼方程更新 Q 函数时，选用的折扣系数 γ 值为 0.9。本文经验池的大小设定为 1000，每次随机采样的最小批大小为 128，训练次数为 20 个 epoch。在此实验中本文仅考虑单一具体类别的目标进行检测。下面以狗这一类图片来进行实验。

4.5.2 实验结果

模型训练过程中，损失函数的变化如图 4.5 所示，从图中可以看出，在网络训练的过程中，模型的损失值变化幅度比较大，这是由于本文使用了多层特征来进行目标检测，在网络训练过程中，需要使用多个特征层的部分区域的信息对整个网络的参数进行更新，加大了训练过程中的困难性，导致训练过程中损失值的变化比较剧烈。但即便如此，从图中仍然可以看出，其损失值的整体趋势仍然是下降的，表明模型的参数得到正确的更新，由此可见在训练过程中模型学习到了相关的定位知识。

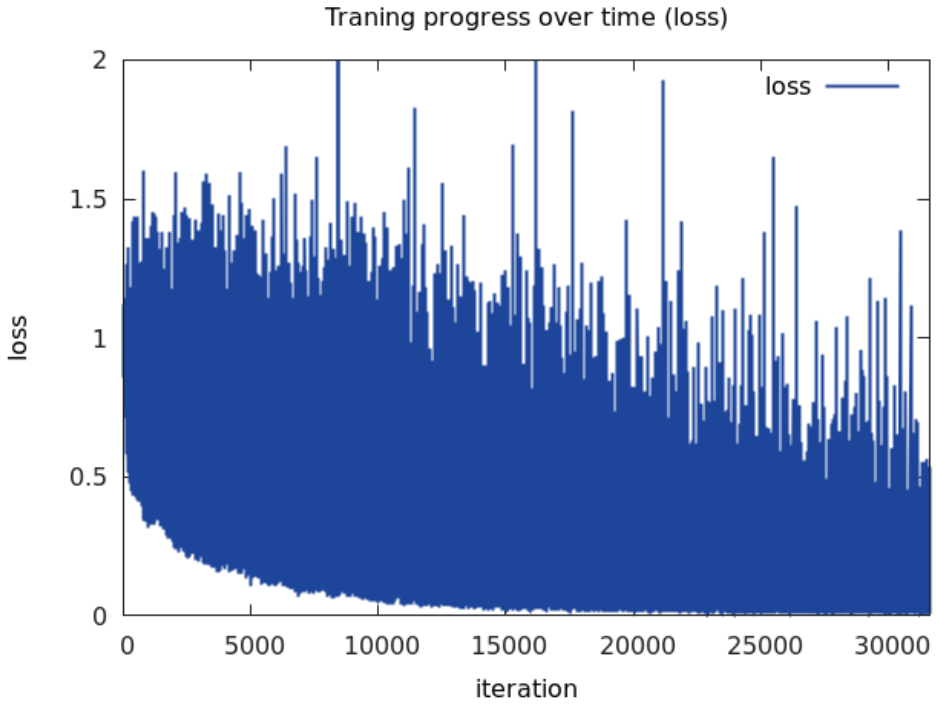
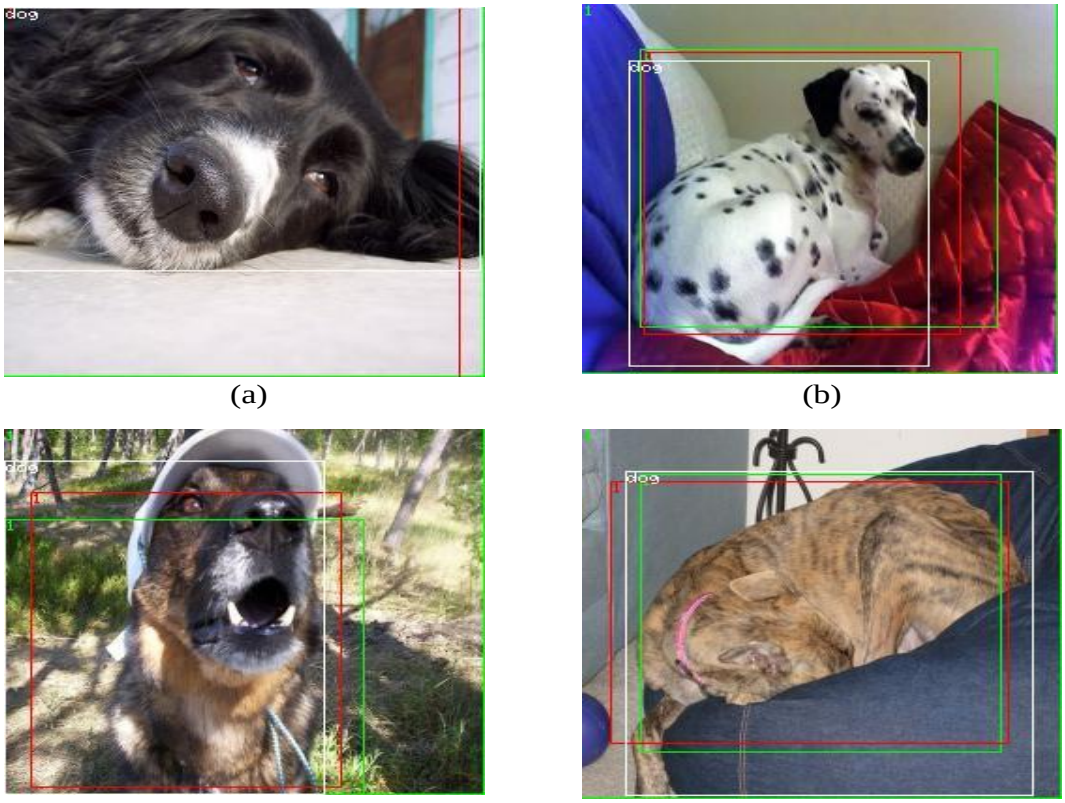


图 4.5 模型损失函数变化图

目标检测效果如图 4.6 所示。其中绿色框代表 DQN 网络每次产生的候选区域，红色框代表结合回归网络所得到的最终定位结果，白色框代表真实目标区域。从图中可以看出，本文所提出的算法能够在评估少量候选区域的情况下，准确定位目标。对于图 4.6(a)，行动选择网络能够在仅评估一个候选区域的情况下就有足够的信心确定当前区域即为目标区域，无需继续搜索。而对于图 4.6(c)，行动选择网络评估了两个候选区域，当评估第一个候选区域时，网络没有足够的信息确定目标，因此只能向着最有可能是目标的区域进行搜索，跃迁到上一层特征，根据得到的第二个候选区域以及其对应的特征，此时能够有足够的证据表明此区域为目标，此时终止搜索，并有回归网络对定位结果进行微调，得到最终的结果。图 4.6(b)与图 4.6(d)表明本文的算法能够根据目标所在的位置，准确的调整搜索的方向，表明算法的鲁棒性较好。



(a) (b)

(c) (d)

图 4.6 目标检测效果图

如表 4.1 所示，本文算法在单类别的目标检测中，相较于其他基于强化学习的目标检测算法具有明显的优势，其中在 aeroplane 类别中，相较于 Bueno 等人的算法在精确度方面提高了 17.19%，相较于 Caicedo 等人的算法精确度提高了 12.01%；其中在 dog 类别中，相较于 Bueno 等人的算法在精确度方面提高了 18.55%，相较于 Caicedo 等人的算法精确度提高了 16.15%，表明了本文算法能有效提升目标定位的精确度。

表 4.1 VOC2007 数据集上的目标定位准确度

算法/类别	areoplane	dog
Bueno 算法	32.23%	32.18%
Caicedo 算法	37.41%	34.58%
本文算法	49.42%	50.73%

从图 4.7 可以看出，对于大部分的目标，使用本文的方法，最多需要进行 3 次搜索即可成功定位目标，相对于等人的算法，能够显著减少定位目标所需处理的候选区域数量，提高目标检测的速度。

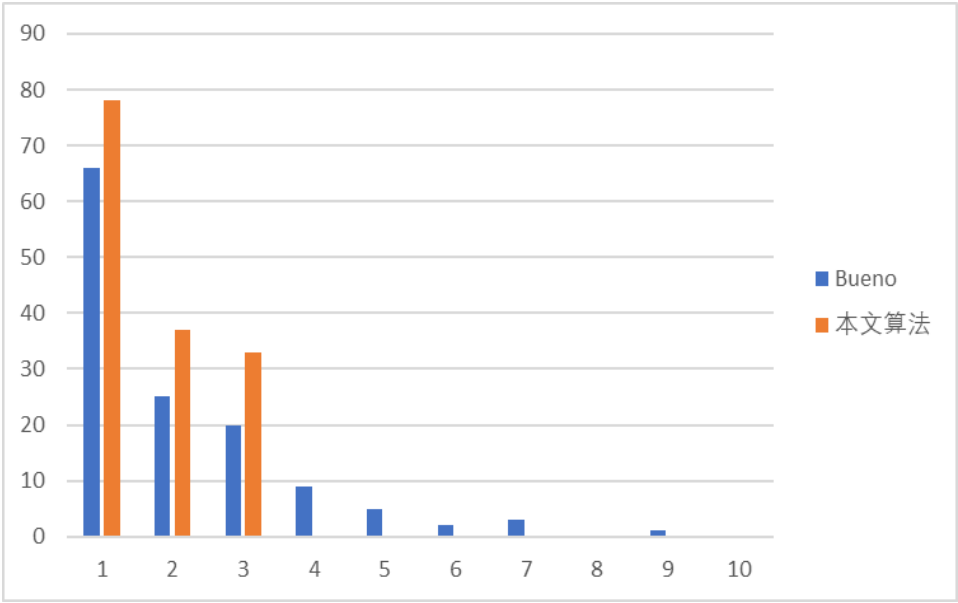
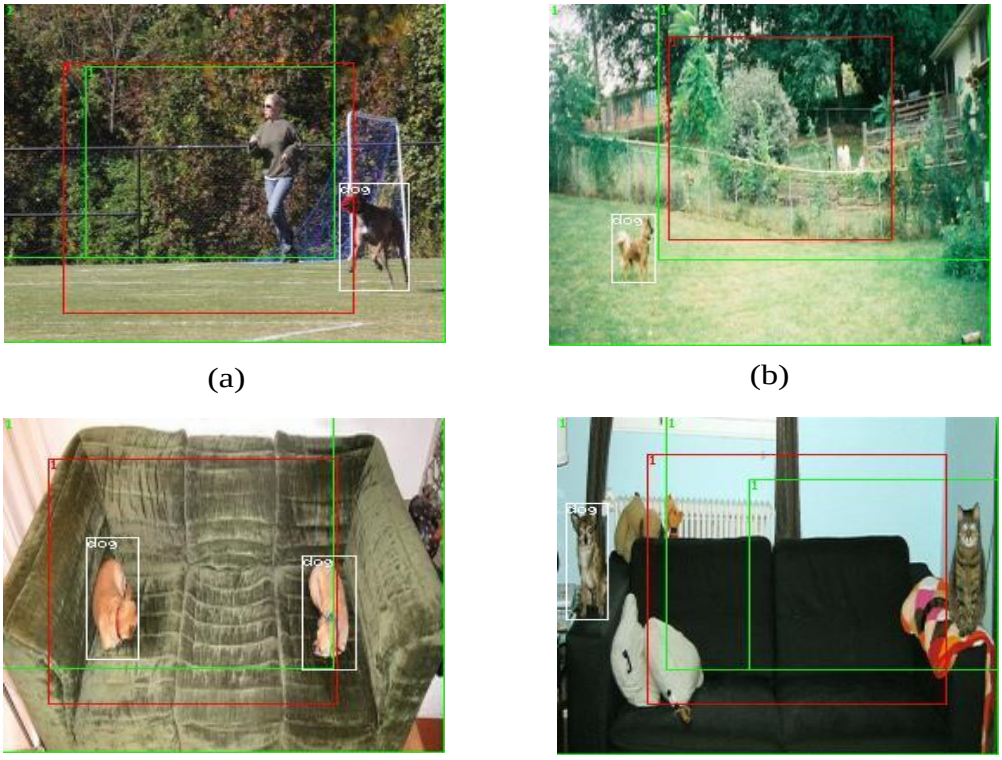


图 4.7 定位目标所需要分析的候选区域数量

4.5.3 误差分析

但本文的算法也存在着一些检测失败的案例，如图 4.8 所示。针对图 4.8(a)与图 4.8(b)，分析原因为：目标尺寸过小，而背景区域比较复杂，且占据位置处于角落中，导致算法很难确定目标位置。对于图 4.8(b)，其存在两个相似的目标，且两个目标的尺寸均很小，背景中的沙发占据了大部分面积，算法难以确定目标所在位置。针对图 4.8(d)，目标所处位置靠近图像的边缘，且图像中存在着其他类别的目标即猫目标，而猫的特征要比狗的类别显著，算法无法判断，导致向着猫目标所在的区域进行搜索，进而导致检测的失败。由于本文仅适用三层特征图，每一次的搜索都会执行一次特征层之间的跃迁，最多仅搜索两次，因此其搜索的区域范围受到限制，对于一些较小的目标，行动选择网络所产生的候选区域无法紧密包围

到这些目标。



(a) (b)

(c) (d)

图 4.8 检测失败案例

通过对失败案例进行分析，从提高定位精度出发，未来研究方向将从以下三个方面进行：

1）在基于多层特征的目标检测算法中，即使加了回归操作，对更小的目标检测的效果也不是很好，这是由于本文仅在三层特征图上进行跃迁，未来将通过引入更多的层来覆盖更多尺寸大小的目标。

2）在本文所提出的目标检测算法中的特征提取网络使用的为 VGG16 架构，而深度学习领域已经出现了很多性能比 VGG16 高很多的网络架构，未来将引入最新的网络架构如 ResNet 和 GoogLeNet[60]等，作为特征提取网络来提高算法性能。

3）采用按面积选择子区域的方法在鲁棒性方面仍存在一些不足。例如开始阶段某一步偏离了目标区域，那么后面的搜索过程就会越来越偏离目标，导致无法检测到目标。未来可尝试的解决方法为：设计更具有鲁棒性的行动，使得行动选择网络在搜索的过程中，可以自动矫正自己的犯下的错误，这样即使初始阶段搜索路线偏离的目标区域，那么在后期也能通过对路线矫正最终回到目标区域。

4.6 本章小结

将强化学习引入到目标检测任务中，将目标检测建模为在候选区域上自顶向下不断缩小，逐渐逼近候选区域的过程。为了充分利用多层特征信息，将特征提取过程中的相关特征层的

特征进行存储，通过建立特征-候选区域映射关系，创新性地将不同尺度大小区域对应到不同特征信息图上，深度强化学习根据当前特征来判断下一子区域的选择，使得目标检测任务中所需评估的候选区域数量呈指数级减少，极大的提高了检测的速度。为了提高定位的准确度，本文还引入了一个回归网络对定位结果进行精确定位。实验结果表明，使用本文的方法，能够有效利用多层特征信息，减少目标检测所需处理候选区域的数量，提高目标检测的准确度与精准度。

第 5 章　总结与展望

5.1 总结

目标检测是计算机视觉领域中最基本的内容之一，其目标是在图片中并定位所有的目标，并确定其所属的类别。典型的目标检测是使用一个边界框将目标紧紧包含起来。主流的目标检测算法通常需要处理大量的候选区域，而这些区域的处理过程已经成为限制算法速度的瓶颈。本文从减少候选区域数量，提高检测速度出发，探讨了基于深度强化学习的目标检测。在本文中，其重点工作和主要研究成果如下：

第一章，系统阐述了目标检测技术的研究背景和意义，并从基于区域的目标检测、基于回归的目标检测以及基于深度强化学习的目标检测算法三个方面对基于深度学习的目标检测算法进行展开阐述，并给出本论文的内容安排。

第二章，首先介绍强化学习的相关理论基础，着重介绍马尔可夫决策过程 MDP，以及构成强化系统的组成部分；其次介绍深度学习，对神经元模型的构成以及训练深度网络的反向传播算法进行描述；最后对如何将深度学习与强化学习进行结合进行分析，并介绍了 DQN 的相关原理。

第三章，提出联合边框回归的深度强化学习视觉目标检测算法。基于强化学习的目标检测算法在检测过程中通常采用预定义的搜索行动，其产生的候选区域的形状和尺寸变化单一，导致目标检测的精确度较低。为此，在基于 DQN 的目标检测算法基础上，提出了联合边框回归与深度强化学习的目标检测算法。算法首先由 DQN 根据初始候选区域所提取的信息决定相应的搜索行动，根据行动选择下一个逼近真实目标的候选区域；然后重复上述过程，直至 DQN 有足够的信心确定当前区域为目标区域时，终止搜索过程；最后由回归网络对当前区域坐标进行边框回归，达到精确定位的目的。在 Pascal VOC 数据集上的实验结果显示，在 dog 与 areoplane 类别目标检测中，与 Caicedo 算法相比，其准确率分别提高了 12.86%与 10.99%，表明通过引入边框回归有效地提高了视觉目标检测的精确度。

第四章，提出基于多层特征的深度强化学习目标检测算法。由于不同尺寸大小的目标在不同深度的特征网络上的表达能力不同，仅使用单层特征图的目标检测算法，很难保证所有尺寸大小的目标信息都能得到充分表达，导致此类算法对尺寸变化较大的目标的检测效果较差。为了充分利用多层网络特征，本文在基于深度强化学习的目标检测算法基础上，引入多层特征，智能体能够根据候选区域的尺寸大小，按照候选区域-特征映射关系，提取相应的特征层上的特征，实现多层特征与强化学习相结合的目标检测。在 Pascal VOC 数据集中 areoplane 与 areoplane 类别目标检测中的实验结果显示，该算法相较于 Bueno 算法在准确率上

分别提高了 12.01%与 16.15%，验证了本文算法的有效性。

第五章，总结与展望。对全文进行总结，分析本文算法中的不足之处以进一步确认此后需要进行研究的方向。

5.2 展望

目标检测是计算机视觉领域研究的重点，近些年来基于深度学习的目标检测算法取得了显著的成功，但这些算法通常依赖特定的候选区域生成算法生成大量的候选区域，然后对这些候选区域进行后续处理，如区域分类、去除冗余窗口等。目标检测的性能依赖于候选区域生成算法，为了保持较高的召回率，需要生成大量的候选区域，候选区域的增多就会影响到检测算法的速度，而为了提高算法的速度，又会难以保持较高的召回率。近些年深度强化学习在交互、探索与控制等领域取得了显著的成功，引起了研究人员对深度强化学习研究的热潮，考虑到强化学习在控制决策方面的巨大优势，本文从较少候选区域的数量出发，将强化学习技术引入目标检测框架中，通过定义的 agent 在与图像环境交互的过程中学习策略，选择相应的行动逐渐逼近真实目标区域，能够显著减少需要评估的候选区域数量，提高了检测的速度。本次论文写作过程中，由于个人经验、技术等方面的不足，还有许多地方需要完善和提高，以下列出后续的参考研究：

(1) 在基于多层特征的目标检测算法中，即使加了回归操作，对更小的目标检测的效果也不是很好，这是由于本文仅在三层特征图上进行跃迁，未来将通过引入更多的层来覆盖更多尺寸大小的目标。

(2) 在本文所提出的两种目标检测算法中的特征提取网络使用的均为 VGG16 架构，而深度学习领域已经出现了很多性能比 VGG16 高很多的网络架构，未来将引入最新的网络架构如 ResNet 和 GoogLeNet[60]等，作为特征提取网络来提高算法性能。

(3) 采用按候选区域长宽比例选择下一探索区域的方法在鲁棒性方面仍存在一些不足。例如开始阶段某一步偏离了目标区域，那么后面的搜索过程就会越来越偏离目标，导致无法检测到目标。未来可尝试的解决方法为：设计更具有鲁棒性的行动，使得 agent 在搜索的过程中，可以自动矫正自己的犯下的错误，这样即使初始阶段搜索路线偏离的目标区域，那么在后期也能通过对路线矫正最终回到目标区域。

(4) 在 DQN 网络之后，很多基于 DQN 扩展的深度强化学习算法被提出来，如深度双 Q 网络 DDQN，以及深度循环 Q 网络 DRQN 等，目前基于策略的深度强化学习有 A3C 算法、UNREAL 算法等。未来将在目标检测中引入这些领先的深度增强学习算法，并对不同的增强学习算法对目标定位算法的性能的提升做一个量化的对比。

(5) 本文中的实验数据均为 VOC2007 与 VOC2012，未来将在更大的数据集中对算法进行验证，如 COCO 以及 ImageNet 数据库等。

致　　谢

时光飞逝，几经彷徨求索，不知不觉，两年半的研究生生活也即将进入尾声。回顾自己将近二十载的漫漫求学路，有汗水、有泪水，但更多的是记忆深处老师的悉心教导和同学们的快乐相伴，在此论文完成之际我要向他们表达最诚挚的感谢。

饮水思源，学成念师。首先，我要向郭春生老师致以诚挚的谢意和崇高的敬意，感谢老师在学习、生活等方面给予的帮助和指导。记得刚研一开学的时候，老师就说过研究生学习不同于本科，而是独立自主、培养兴趣的过程，每周的工作小结让我养成自我归纳、循循渐进的好习惯。老师对我严格要求，并帮助我开展基于强化学习的目标检测算法的研究，从公式推导、程序调试、平台验证一直到论文完成，每一步都离不开老师的悉心指导和亲切关怀。尤其是在临近毕业的半年里，老师多次细心指导小论文修改工作，让我能够按时完成小论文写作并发表。在生活中，老师为鼓励我们多锻炼身体，购置了许多体育器材，并经常带我们一起去打羽毛球。老师渊博的专业知识、求实创新的工作作风、脚踏实地的处事态度使我受益匪浅，在今后的学习、工作中我会谨记老师的每一点教诲，做一个求知若渴、勤勉尽责、诚实守信的人。

其次，我要感谢通信工程学院的唐向宏老师、赵知劲老师、李光球老师等任课老师，感谢你们时时刻刻为我们指引着前进的航程，谆谆教诲，此生不忘。同时还要感谢楼浩英老师、李世尧老师和滕旭超老师，感谢你们在生活方面给予我的指导和帮助。研究生期间，我的进步离不开学院对我的关心和帮助。

然后，我要感谢我的父母，研究生期间我也知道家里的事情比较多，你们不仅在生活上给我提供了良好的物质保障，让我可以全身心地投入到学习和科研中去，当我遇到困难或者压力很大的时候，你们也会时常电话联系，让我放宽心，感谢我这个充满温暖而又坚强的家。

最后，我要感谢和我一起度过研究生生活的朋友们，感谢已毕业的师兄师姐们在学习上和找工作上对我的关心和帮助，感谢实验室的全体同学的陪伴与互助，是你们和我站在了一条战线，是你们给了我安静舒适的学习环境，是你们陪我度过了从无知到有知的学习阶段，是你们给了我继续努力科研的动力，感谢你们对我的包容与关怀。至此，再一次衷心感谢一路上帮助过我的老师、家人、同学和朋友，感谢你们的陪伴、关心和鼓励，我的新征程也即将起航，借用一句诗句激励自己，“风雨不改凌云志，振衣濯足展襟怀。行方智圆煅内蕴，海阔天空铸宏图”。

参考文献

[1] Mita T, Kaneko T, Hori O. Joint haar-like features for face detection[C]//Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. IEEE, 2005, 2: 1619-1626.

[2] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, 1: 886-893.

[3] Lowe D G. Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image: US, US6711293[P]. 2004.

[4] Lim J J, Pirsiavash H, Torralba A. Parsing IKEA objects: fine pose estimation[C]. IEEE International Conference on Computer Vision. IEEE Computer Society, 2013:2992-2999.

[5] Boser B E. A training algorithm for optimal margin classifiers[C]. The Workshop on Computational Learning Theory. ACM, 1992:144-152.

[6] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]//Computer Vision and Pattern Recognition, 2001, 1: 511-518.

[7] Sempau J, Wilderman S J, Bielajew A F. DPM, a fast, accurate Monte Carlo code optimized for photon and electron radiotherapy treatment planning dose calculations.[J]. Physics in Medicine & Biology, 2000, 45(8):2263-2291.

[8] Everingham M, Gool L V, Williams C K I, et al. The pascal visual object classes (VOC) challenge[J]. International Journal of Computer Vision, 2010, 88(2):303-338.

[9] Felzenszwalb P F, Girshick R B, McAllester D, et al. Object detection with discriminatively trained part-based models[J]. IEEE transactions on pattern analysis and machine intelligence, 2010, 32(9): 1627-1645.

[10] Mnih V. Machine learning for aerial image labeling[D]. Toronto：University of Toronto , 2013：1-103.

[11] Sermanet P, Kavukcuoglu K, Chintala S, et al. pedestrian detection with unsupervised multi-stage feature learning[C]. IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2013:3626-3633.

[12] Dahl G E, Acero A. context-dependent pre-trained deep neural networks for large-vocabulary speech recognition[J]. IEEE Transactions on Audio Speech & Language Processing, 2012, 20(1):30-42.

[13] Graves A, Mohamed A R, Hinton G. Speech recognition with deep recurrent neural networks[J]. 2013, 38(2003):6645-6649.

[14] Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge[J]. International Journal of Computer Vision, 2015, 115(3):211-252.

[15] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[16] Carreira J, Sminchisescu C. Cpmc: Automatic object segmentation using constrained parametric min-cuts[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(7): 1312-1328.

[17] Sande K E A V D, Uijlings J R R, Gevers T, et al. Segmentation as selective search for object recognition[C]. IEEE International Conference on Computer Vision. IEEE, 2012:1879-1886.

[18] Pont-Tuset J, Arbelaez P, J T B, et al. Multiscale combinatorial grouping for image segmentation and object proposal generation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, 39(1):128-140.

[19] Erhan D, Szegedy C, Toshev A, et al. Scalable object detection using deep neural networks[C]. Computer Vision and Pattern Recognition. IEEE, 2014:2155-2162.

[20] Sermanet P, Eigen D, Zhang X, et al. Overfeat: integrated recognition, localization and detection using convolutional networks[J]. arXiv preprint arXiv:1312.6229, 2013.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation[J]. CVPR, 2014. 1-8.

[22] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(9):1904-1916.

[23] Yoo D, Park S, Lee J Y, et al. AttentionNet: aggregating weak directions for accurate object detection[C]. IEEE International Conference on Computer Vision. IEEE, 2015:2659-2667.

[24] Girshick R. Fast R-CNN[C]. IEEE International Conference on Computer Vision. IEEE, 2015:1440-1448.

[25] Ren S, Girshick R, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015:91–99.

[26] Dai J, Li Y, He K, et al. R-fcn: Object detection via region-based fully convolutional networks[C]//Advances in neural information processing systems. 2016: 379-387.

[27] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[28] Mordan T, Thome N, Cord M, et al. Deformable part-based fully convolutional network for object detection[J]. arXiv preprint arXiv:1707.06175, 2017.

[29] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[30] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.

[31] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J]. arXiv preprint arXiv:1312.5602, 2013.

[32] Naddaf Y, Naddaf Y, Veness J, et al. The arcade learning environment: an evaluation platform for general agents[J]. Journal of Artificial Intelligence Research, 2013, 47(1):253-279.

[33] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529(7587): 484-489.

[34] Mathe S, Pirinen A, Sminchisescu C. Reinforcement learning for visual object detection[C]. Computer Vision and Pattern Recognition. IEEE, 2016:2894-2902.

[35] Caicedo J C, Lazebnik S. Active object localization with deep reinforcement learning[C]. IEEE International Conference on Computer Vision. IEEE, 2015:2488-2496.

[36] Bellver M, Giró-i-Nieto X, Marqués F, et al. Hierarchical object detection with deep reinforcement learning[J]. arXiv preprint arXiv:1611.03718, 2016.

[37] Kong X, Xin B, Wang Y, et al. Collaborative deep reinforcement learning for joint object search[C]//The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.

[38] Hara K, Liu M Y, Tuzel O, et al. Attentional network for visual object detection[J]. arXiv preprint arXiv:1702.01478, 2017.

[39] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553):436-444.

[40] Yu Kai, Jia Lei, Chen Yu-Qiang, Xu Wei. Deep learning: yesterday,today, and tomorrow[J]. Journal of Computer Research and Development, 2013, 50(9): 1799-1804.

[41] Xu K, Ba J, Kiros R, et al. Show, attend and tell: neural image caption generation with visual attention[J]. Computer Science, 2015:2048-2057.

[42] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[C]//Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing.Vancouver, Canada, 2013:6645-6649

[43] Goldberg Y. Neural network methods for natural language processing[J]. Synthesis Lectures on Human Language Technologies, 2017, 10(1):1-309.

[44] Yeung S, Russakovsky O, Mori G, et al. End-to-end learning of action detection from frame glimpses in videos[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2678-2687.

[45] Karpathy A, Toderici G, Shetty S, et al. Large-scale video classification with convolutional neural networks[C]// Computer Vision and Pattern Recognition. IEEE, 2014:1725-1732.

[46] Mozer S, M C, Hasselmo M. Reinforcement learning: an introduction[J]. Machine Learning, 1992, 8(3-4):225-227.

[47] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets[J]. Neural Computation, 2006, 18(7): 1527-1554

[48] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[49] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]//International Conference on International Conference on Machine Learning. Omnipress, 2010:807-814.

[50] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(1):1929-1958.

[51] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift[C]//International Conference on International Conference on Machine Learning. JMLR.org, 2015:448-456.

[52] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[53] Hasselt H, Guez A, Silver D. Deep reinforcement learning with double Q-Learning[C]//Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI Press, 2016: 2094-2100.

[54] Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning[C]//International Conference on Machine Learning. 2016: 1995-2003.

[55] Hausknecht M, Stone P. Deep recurrent Q-Learning for partially observable MDPs[C]//2015 AAAI Fall Symposium Series. 2015.

[56] Mnih V, Badia A P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]//International Conference on Machine Learning. 2016: 1928-1937.

[57] Jaderberg M, Mnih V, Czarnecki W M, et al. Reinforcement learning with unsupervised auxiliary Tasks[J]. arXiv preprint arXiv:1611.05397, 2016.

[58] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[59] Torralba A, Oliva A, Castelhano M S, et al. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search.[J]. Psychological Review, 2006, 113(4):766-786.

[60] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Computer Vision and Pattern Recognition. IEEE, 2015:1-9.

.

附　录

作者在读期间发表的学术论文及参加的科研项目

发表的学术论文：

[1] 舒朗，郭春生. 基于回归与深度强化学习目标检测算法，软件导刊，已录用.

参加的科研项目：

[1] 视频异常事件检测中的群体特征感知研究.国家自然科学基金资助项目(61372157). 2014-2017.