

分类号	<u>TN929.5</u>	密级	<u>公开</u>
UDC	<u>621.39</u>	学位论文编号	<u>D-10617-308-(2021)-01059</u>

重庆邮电大学硕士学位论文

中文题目	<u>边缘网络 VNF 需求预测与 迁移方法研究</u>
英文题目	<u>Research on VNF Demand Forecast and Migration Method of Edge Network</u>
学 号	<u>S180101060</u>
姓 名	<u>鲍晓萌</u>
学位类别	<u>工学硕士</u>
学科专业	<u>信息与通信工程</u>
指导教师	<u>黄宏程 副教授</u>
完成日期	<u>2021 年 3 月 8 日</u>

摘要

万物互联时代,随着海量智能终端、各种实时网络服务的涌现,目前臃肿的网络不仅难以支撑如虚拟现实(Virtual Reality, VR)、智慧医疗等技术的需求,而且在网络维护升级、扩展方面极为困难。网络功能虚拟化(Network Function Virtualization, NFV)技术应运而生,旨在通过软件的方式实现各种运行在专有硬件之上的网络功能,不仅能够提高网络架构的可扩展性、使得网络能够更加灵活,并降低系统成本,且可以利用边缘计算(Edge Computing, EC)范式,将网络实时业务下沉至网络边缘,从而提高用户服务体验质量,降低用户端的成本。本文围绕边缘网络中虚拟网络功能(Virtual Network Function, VNF)的预测和迁移方法展开研究,其重要工作如下:

第一,网络边缘部署各种虚拟网络功能实例可以为终端用户提供不同的网络服务,提前部署适当数量的 VNF 不仅可以降低系统成本,也可以保障网络服务质量。针对当前 VNF 需求预测方法准确率较低且不适用于边缘网络的问题,本文针对边缘网络场景提出了一种基于支持向量回归(Support Vector Regression, SVR)与门控循环单元(Gate Recurrent Unit, GRU)模型结合的 VNF 需求预测方法。本文考虑到网络边缘流量具有突发性、自相似性及长相关性等特点,结合 SVR 和 GRU 两种模型的优点,利用计算复杂度较低的 SVR 和 GRU 模型分别提取网络服务历史时序数据的短期特征和长期特征,以提高 VNF 需求预测准确率,实现边缘网络中 VNF 的提前放置。通过仿真表明,本文提出的预测方法相较于长短期记忆(Long Short-Term Memory, LSTM)网络模型能够降低 15%~18%的误差,有更佳预测效果。

第二,考虑到网络中边缘服务器节点过载时,无法满足其覆盖范围下所有用户的需求,因此需要进行 VNF 的迁移或复制,保证网络服务质量,实现负载均衡。本文权衡 VNF 复制产生的成本和 VNF 迁移导致的网络服务质量下降,协同 VNF 的复制,提出了一种基于深度强化学习的 VNF 迁移方法。该方法首先通过基于空闲资源和负载的阈值分析,以筛选合适的目标服务器节点,并依据时延与成本的优化问题构建马尔科夫决策过程(Markov Decision Process, MDP),最后通过深度 Q 学习(Deep Q-learning)实现最优策略的选择。仿真结果表明,本文提出的 VNF 迁移方法与传统的 VNF 迁移方法相比,在保证服务功能链的时延需求前提下,能够以更小的

成本实现服务器的负载均衡。

关键词：网络功能虚拟化，边缘计算，预测，迁移，深度强化学习

Abstract

In the era of the Internet of Everything, with the emergence of a large number of smart terminals and various real-time network services, the current bloated network is not only difficult to support the needs of technologies such as Virtual Reality (VR) and smart medical care, but also in terms of network maintenance upgrades and expansion. Network Function Virtualization (NFV) technology emerged at the historic moment, aiming to implement various network functions running on proprietary hardware through software, which not only improves the scalability of the network architecture, but also makes the network more flexible and reduces system costs. It can use the Edge Computing (EC) paradigm to sink real-time network services to the edge of the network, thereby improving the quality of user service experience and reducing user-side costs. This thesis focuses on the prediction and migration methods of Virtual Network Function (VNF) in the edge network. Its important work is as follows:

First, examples of various virtual network functions placed at the edge of the network can provide end users with different network services. Deploying an appropriate number of VNFs in advance can not only reduce system costs, but also improve network service quality. Aiming at the problem that the current VNF demand forecasting method is low in accuracy and not suitable for edge networks, this thesis proposes a VNF based on the combination of Support Vector Regression (SVR) and Gated Recurrent Unit (GRU) models for edge network scenarios. Demand forecasting methods. This thesis considers that network edge traffic has the characteristics of burstiness, self-similarity, and long correlation. Combining the advantages of the SVR and GRU models, the SVR and GRU models with lower computational complexity are used to extract the historical time series data of network services. Short-term features and long-term features to improve the accuracy of VNF demand forecasts and realize the advance placement of VNFs in the edge network. The simulation results show that the prediction method proposed in this paper can reduce the error of 15% - 18% compared with the Long Short Term Memory (LSTM) network model, and has better prediction effect.

Second, when considering the overload of edge server nodes in the network, it cannot meet the needs of all users under its coverage, Therefore, it is necessary to migrate or copy VNF to ensure network service quality and realize load balancing. This thesis weighs the

cost of VNF replication and the degradation of network service quality caused by VNF migration, and cooperates with VNF replication, and proposes a VNF migration method based on deep reinforcement learning. This method first selects suitable target server nodes through threshold analysis based on idle resources and load, and constructs a Markov decision process based on the optimization of delay and cost, and finally through Deep Q Learning (Deep Q-learning) realizes the selection of the optimal strategy. The simulation results show that, compared with the traditional VNF migration method, the VNF migration method proposed in this thesis can achieve server load balancing at a lower cost under the premise of ensuring the delay requirement of the service function chain.

Keywords: network function virtualization, edge computing, prediction, migration, deep reinforcement learning

目录

摘要.....	I
目录.....	V
图录.....	VIII
表录.....	IX
第 1 章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 VNF 需求预测	2
1.2.2 VNF 迁移	3
1.3 主要研究工作.....	5
1.4 本文的组织结构.....	7
第 2 章 相关理论综述.....	8
2.1 网络功能虚拟化与边缘计算.....	8
2.1.1 网络功能虚拟化	8
2.1.2 边缘协同	10
2.2 网络流量预测方法.....	11
2.2.1 传统预测模型	12
2.2.2 支持向量回归	12
2.2.3 门控循环神经网络	13
2.3 VNF 迁移概述.....	14
2.3.1 冷迁移	15
2.3.2 实时迁移	15
2.3.3 VNF 迁移算法研究	15
2.4 深度强化学习.....	16

2.4.1 基于值函数的 Q 学习方法.....	17
2.4.2 深度 Q 网络算法.....	17
2.5 本章小结.....	19
第 3 章 边缘网络中 VNF 的需求预测.....	20
3.1 系统模型及问题描述.....	20
3.2 基于 SVR 与 GRU 的组合预测模型	21
3.2.1 边缘网络业务流量特征	21
3.2.2 SVR 模型.....	21
3.2.3 GRU 模型	22
3.2.4 SVR-GRU 模型	22
3.2.5 模型复杂度分析	24
3.3 VNF 需求分析.....	25
3.4 仿真结果与性能分析.....	26
3.4.1 实验设计	26
3.4.2 结果分析	28
3.5 本章小结.....	31
第 4 章 基于深度强化学习的 VNF 迁移.....	32
4.1 系统模型及问题描述.....	32
4.2 基于空闲资源和负载的阈值分析.....	33
4.3 目标优化模型.....	34
4.3.1 VNF 复制策略	34
4.3.2 优化目标	35
4.4 马尔科夫决策过程.....	37
4.5 基于深度 Q 网络的 VNF 迁移.....	41
4.6 仿真结果与性能分析.....	45
4.6.1 自身性能分析	46
4.6.2 不同方案间的性能对比	48

4.7 本章小结.....	48
第 5 章 结束语.....	50
5.1 论文总结.....	50
5.2 未来展望.....	51
参考文献.....	52
致谢.....	57
攻读硕士学位期间从事的科研工作及取得的成果.....	59

图录

图 2.1 网络演变过程.....	8
图 2.2 NFV 框架.....	9
图 2.3 边缘与边缘协同示意图.....	11
图 2.4 支持向量回归偏差示意图.....	13
图 2.5 GRU 内部结构	14
图 2.6 智能体与环境交互过程.....	16
图 2.7 Q 网络结构示意图.....	18
图 3.1 VNF 需求预测系统网络模型.....	21
图 3.2 实验仿真过程.....	28
图 3.3 不同模型预测效果.....	29
图 3.4 权重确定数据与 RMSE 的关系.....	30
图 4.1 VNF 迁移模型构建过程.....	33
图 4.2 基于空闲资源和负载的阈值分析流程.....	34
图 4.3 VNF 复制模型.....	35
图 4.4 基于 DQN 的 VNF 迁移算法框架	42
图 4.5 神经网络结构.....	44
图 4.6 SFC 数量对各性能的影响	47
图 4.7 时延和成本与 SFC 数量的关系	48

表录

表 3.1 SVR-GRU 组合模型的权重确定算法 24

表 3.2 SVR-GTU 与 LSTM 的空间复杂度对比 25

表 3.3 数据预处理格式 27

表 3.4 不同模型相关参数 28

表 3.5 不同模型 RMSE 指标对比..... 30

表 4.1 基于 DQN 的 VNF 迁移策略训练算法..... 43

表 4.2 网络场景的具体仿真参数 46

第 1 章 绪论

1.1 研究背景及意义

第五代移动通信(5th Generation Mobile Networks, 5G)系统以其灵活性和高效性为用户提供了超高吞吐量和超低时延的服务体验。作为 5G 的关键技术,网络功能虚拟化(Network Function Virtualization, NFV)和移动边缘计算(Mobile Edge Computing, MEC)相关研究越来越多,也受到了工业界的诸多关注^[1]。且如智能手机和平板电脑这样的移动设备已经成为主要的通信工具,与此同时,人脸识别、自然语言处理、互动游戏和增强现实等新兴的移动应用需要大量的计算资源来提高性能,从而导致较高的能耗。此外,它们还需要各种网络功能服务,包括防火墙、入侵检测系统和负载平衡器等,以保证应用程序的正确和安全执行^[2]。在边缘网络中放置不同类型的虚拟网络功能,能够满足不同网络服务的需求,边缘网络中不同节点覆盖范围下的需求随时间变化,如何有效的放置虚拟网络功能(Virtual Network Function, VNF)以及快速有效的迁移成为研究的关键。

随着全球数据流量以及连接至网络中的智能设备等剧增,边缘数据爆炸式的增长速度远超过集中式的云计算能力,海量的数据在网络边缘设备产生,并且这些数据传输到云中心会导致网络带宽负载严重,增大网络延迟^[3]。边缘计算主要将部分计算、存储等资源下放在网络的边缘,就近进行数据的处理,从而降低网络时延,节省带宽资源,以满足在低时延应用、实时网络业务、隐私保护等方面的基本需求。而且灵活的边缘架构可以与 NFV 很好的契合, NFV 可以有效降低系统成本,并能提升网络业务的响应速度,增强网络的可扩展性,网络运营商通过利用 NFV 技术,可以部署全面的、可管理的、分布式的云框架,将虚拟化优势最大化。

近些年来,互联网以及相关应用的快速发展,增大了网络中计算、存储和带宽等资源,但同时伴随的是各种网络实时业务、移动应用的涌现,这些业务及应用对时延的需求越来越高^[4]。在边缘计算环境中,应用程序和服务功能可以被放置在边缘节点中,从而减少服务延迟和后续网络流量。通过将应用程序和服务功能放置在接近用户的位置,边缘节点可以立即响应用户请求,使得网络延迟最小,而不用通过高延迟广域网(Wide Area Network, WAN)访问中心云资源。为了满足移动用户不

断增长的资源需求和严格的服务质量要求，网络服务提供商通常在移动边缘云上实例化一些经常需要的虚拟网络功能实例。在网络边缘实例化 VNF 实例可以缩短网络服务的访问延迟，节省创建新 VNF 实例的时间，提供具有不同类型的网络服务。NFV 将网络功能以软件的方式实现，不仅能够打破目前网络架构的僵化，提高网络的灵活性和扩展性，实现按需动态放置，也降低了资本支出(Capital Expenditure, CAPEX)和运营费用(Operating Expense, OPEX)。且 NFV 能够使得系统在维护升级方面非常便捷，以往运行在专有硬件平台的网络功能维护或升级时，大多时候也需要对专有的硬件进行维护或升级，这些升级所需成本是非常高的，例如 IPv4 网络升级到 IPv6 的过程已经持续了很长一段时间，但目前 IPv4 的使用仍然广泛。

为降低系统成本、端到端延迟等，5G 无线接入网将更加开放，使得网络运营商拥有自主管理权，通过在接入网引入网络功能虚拟化等技术，可以让接入网更加可靠和智能。随着网络的逐渐演变，网络中放置的 VNF 将会越来越多，特别是在资源有限的边缘网络中，放置 VNF 的类型和数量将会显著影响网络服务质量与资源消耗，放置的 VNF 数量不足时，会导致网络服务质量下降，放置的 VNF 过多时，会造成资源的浪费。通过高准确率的 VNF 需求预测可以提前部署合适数量的 VNF，能够保障网络服务质量，有效降低系统成本，而有效的 VNF 的迁移则能够实现网络的负载均衡，提升资源利用率。因此，对于 VNF 需求预测与迁移方法的研究是非常有必要的。

1.2 国内外研究现状

1.2.1 VNF 需求预测

通过在虚拟化平台中运行 VNF，如虚拟机(Virtual Machine, VM)、容器等，可以实现动态扩展并有效地分配网络资源^[5]。运营商通过网络流量预测技术能够预估当前网络资源的使用状况，可以实现网络资源的合理、高效分配^[6]。文献[7]考虑到用户请求业务的动态性和信息反馈延迟引起的虚拟网络资源分配不合理及缓存溢出概率增大等问题，提出了一种基于长短期记忆(Long Short-Term Memory, LSTM)神经网络的流量预测方法，主要依据不同网络切片的历史队列信息来预测未来一

段时间内的负载状况，以实现对网络流量的在线监测。文献[8]基于软化的灵活性和 NFV 环境中大量的监视数据，提出了一种新颖的学习模型，通过使用服务功能链(Service Function Chain, SFC)数据预测 VNF 资源需求。但是由于目前网络边缘流量凸显的复杂特征，单一的模型很难准确地刻画。

文献[9]使用序列到序列(Sequence-to-Sequence, Seq2Seq)学习范例和卷积长短期记忆网络来解决多服务移动流量预测问题，以便有效地提取移动网络流量的复杂时空特征，并高精度地预测城市规模内单个服务的未来需求。文献[10]使用软件定义网络(Software Defined Network, SDN)中基于流的转发思想从数据平面提取流量统计信息，基于流量的时间特性，利用时间相关理论对流量进行建模，提出了基于流的转发流量预测算法，以预测 SDN 流量。文献[11]提出了一种基于基线自回归滑动平均(Autoregressive Moving Average Model, ARMA)的移动服务需求预测方法，该方法以分钟为单位粒度使用相对较短的历史数据，并考虑了多种数据之间的相关性，以无监督的方式利用了每种服务流量需求的功能特性，但是这种方法并不完全适用于 VNF 需求的预测。

文献[12]提出了一种基于机器学习的新颖解决方案，用于在虚拟化环境中横向扩展访问和移动性管理功能(Action Message Format, AMF)资源，主要通过采集的移动网络流量数据，利用经过训练的神经网络来预测用户附加请求速率，可以预测处理即将到来的用户流量所需 AMF 实例的确切数量，但是只是考虑了一种 VNF 即 AMF 的预测，也没有考虑网络边缘流量具有突发性等特点。文献[13]为了寻求一种主动的方法来提前为过载的 VNF 提供新实例，提出了一种基于 LSTM 的网络流量预测模型，并根据估算的流量，设计了一种协同互补的资源分配算法，以减少资源碎片，提高利用率。文献[14]提出了一种在适当的时间预测所需资源以维持 NFV 真正弹性的方法，主要通过预测日前中央处理器(Central Processing Unit, CPU)的利用率实现 VNF 需求的预测，但是提出的解决方法是通过例如 CPU 使用率或内存使用率等系统级信息，而没有考虑到服务级的信息。

1.2.2 VNF 迁移

SDN 和 NFV 技术可以降低网络维护成本，提高可扩展性，在解决问题的同时，也造成了网络负载以及流量分布不平衡等新问题，解决这些问题的一种有效方案

是借助 VNF 的迁移。VNF 迁移是重新部署虚拟网络功能以提供更好的网络服务的关键步骤。文献[15]建议迁移整个虚拟机而不是 VNF，以简化开发工作量，由于该优化问题是不确定多项式(Nondeterministic Polynomial, NP)难问题，针对不同的网络场景，提出了两种启发式算法，倾向于实现良好的 VNF 迁移，在满足网络带宽要求的同时，并最大限度地减少迁移的时间成本，但是并未考虑资源成本因素。

文献[16]提出了一种基于深度信念网络(Deep Belief Nets, DBN)的 VNF 实时迁移算法，主要通过预测 VNF 未来的资源需求来指导迁移，解决了现有方法中缺乏有效预测的问题，并在 VNF 迁移中优化带宽利用率和迁移开销，实现低成本的迁移。文献[17]介绍了一种虚拟网络功能迁移的新颖优化模型，将 VNF 迁移过程分为多个步骤，消除了每个步骤固定迁移期限的限制，使迁移模型更加灵活，并将 VNF 迁移问题构建为混合整数线性规划(Mixed Integer Linear Programming, MILP)，由商用求解器进行求解，这种迁移模型虽然灵活，但是在问题求解上较为困难。文献[18]提出了一种成本模型来评估迁移成本，且由于具有满足计算和网络资源约束的最佳虚拟网络功能迁移问题是 NP 难的，针对性地提出了一种启发式算法来计算近似解，并通过仿真验证了算法的有效性。文献[19]提出一种启发式算法，考虑了 VNF 迁移成本和资源限制的因素，在实现网络负载均衡和改善服务的同时最大程度地降低了网络影响。针对优化模型中系统状态转移概率难以确定和状态、动作空间过大的问题，文献[20]提出了一种迁移策略，该策略可以在 SFC 请求强度变化时确定何时迁移 VNF 以及迁移到何处。以上文献考虑不同的因素提出关于 VNF 迁移的有效算法，但都未考虑到与 VNF 复制的结合。

文献[21]设计并比较了三种用于 VNF 分配和复制的优化方法，包括线性规划(Linear Programming, LP)模型、遗传算法(Genetic Algorithm, GA)和随机拟合放置算法(Random Fitting Placement Algorithm, RFPA)。但是其分析的方法都未考虑在满足时延的前提下，优化成本。文献[22]研究了复制中 VNF 放置以及复制的问题，并提出了用于优化 VNF 放置的线性规划模型，该模型在链接利用率和 CPU 资源利用率两个因素的最小化之间进行权衡，通过权衡 VNF 的复制与 VNF 的迁移，可以实现更好的负载均衡，提升网络服务质量。文献[23]提出了一个线性规划模型来研究 VNF 复制与迁移之间的权衡，本质是在网络服务质量和成本之间进行衡量，但是传统的线性规划模型求解困难。文献[24]提出了一种网络感知的 VNF 迁移机

制，以解决由于 VNF 处理速率不匹配的情况而导致的 SFC 性能下降和服务质量 (Quality of Service, QoS)下降问题，该系统框架是基于 Docker 平台在两种实际拓扑上实现和模拟的，仿真结果表明，VNF 迁移后可以提高服务性能和资源利用率。文献[25]针对混合云/雾场景中存在雾节点移动性可能会导致高延迟的问题，提出了一种启发式方法，可以在每个时隙中找到最佳雾节点，将应用程序组件迁移到更近的雾节点，以在可接受的时间内找到次优解决方案，并通过实验表明了所提出的解决方案提高了有效期和迁移次数。但是所提的启发式方法并未对时延和成本进行权衡，也未涉及 VNF 的复制。

1.3 主要研究工作

当前，大多数传统网络充满了各种专有的硬件设备，这些设备也称为中间盒，例如网络地址转换器(Network Address Translation, NAT)和防火墙等^[26]。网络功能虚拟化可以通过在商用服务器上运行的 VM 中实现网络功能，解决了硬件和软件紧密耦合以及硬件设备供应商单一的问题^[27]。近年来，随着网络的不断优化和更新，网络运营商借助 NFV，可以将一些网络业务下移至网络边缘，降低延迟及成本，例如家庭网关(Residential Gateway, RGW)，服务提供商可以把网络地址转换器、统一资源管理器(Uniform Resource Locator, URL)、防火墙等功能迁移至网络边缘，用户端只用提供简单的设备如机顶盒(Set Top Box, STB)即可。然而在网络边缘放置哪些 VNF 以及放置多少是关键，放置的数量过少，会导致网络服务质量的下降，放置过多的 VNF，则会造成资源的浪费，使得网络边缘节点有限可用的资源更少。因此，通过边缘节点历史信息预测未来 VNF 需求并提前进行放置能够保障网络服务质量，避免资源的浪费，降低成本。此外，由于网络边缘中部署有各种实时网络业务，当因突发事件或服务器维护升级时，边缘节点可能会过载，难以满足其覆盖范围下的用户请求。在边缘节点过载或链路过载，而周围其他节点可用资源较多时，可以通过 VNF 的迁移，将 VNF 迁移至周围其他拥有空闲资源的节点上运行，实现服务器的负载均衡，从而保障网络服务质量。因此，本文的主要研究工作如下：

1. 基于 SVR 和 GRU 的 VNF 需求预测

在边缘云上放置的各种虚拟网功能，可以满足具有网络功能需求的用户请求。放置的 VNF 数量过少，会导致网络服务质量不佳，放置的 VNF 数量过多，则会增

大资源的消耗,如果能够确定合适的 VNF 种类及数量,可以降低系统成本,提高资源利用率。因此本文主要研究 VNF 需求的预测方法,主要依据边缘节点覆盖范围下的不同网络业务历史流量数据,结合不同的时序预测模型,从而更加准确地预测未来一段时间内该区域范围内不同虚拟网络功能的需求,在降低成本的同时,保障用户服务体验质量。具体来说:

(1) 模型选取。由于网络边缘流量与核心网络内的流量特点并不完全相同,网络边缘流量特征具有突发性强、波动性大、长相关等特征,依据这些特征,选择合适的模型进行预测。

(2) 权重的确定。通过结合支持向量回归(Support Vector Regression, SVR)和门控循环神经网络(Gate Recurrent Unit, GRU)模型,主要通过分析不同模型预测误差,利用方差倒数法,赋予两种模型不同的权重,分别提取网络边缘流量的短期特征和长期特征,以提高预测准确率。

(3) VNF 需求分析。通过模型进行 VNF 的预测,进一步分析所需要放置的 VNF 类型和数量,从而提前在边缘服务器上部署不同的 VNF,保障网络服务质量。

2. 基于深度强化学习的 VNF 迁移

边缘节点资源是以虚拟资源池的形式存在,可以动态地将有限的资源分配给不同的虚拟网络功能实例,在单一服务器节点过载无法满足用户任务需求时,一种有效的解决方案是将服务器节点上的部分 VNF 迁移至周围具有空闲资源的节点上,实现负载均衡。本文综合考虑服务功能链时延与迁移成本,结合 VNF 的复制策略,选择合适的节点进行迁移,使保障网络服务基本质量的同时,可以以较小的成本实现服务器的负载均衡。具体如下:

(1) 基于空闲资源和负载的阈值分析。在需要进行 VNF 迁移或复制时,分析周围其他节点的空闲资源、当前负载状态以及预测的未来负载状态,并按照空闲资源大小排序,选择最合适的几个节点作为最后迁移的候选目标节点,加快对问题的求解。

(2) 马尔科夫决策过程构建。通过结合 VNF 的复制,分析需要解决的问题,即在保证服务功能链的基本时延需求的同时,以最小的成本实现服务器的负载均衡,将其转换为目标优化问题,并构建马尔科夫决策过程,最终将问题的解决转换为最佳策略的选择。

(3) 基于深度强化学习的 VNF 迁移。由于传统的整数线性规划难以解决 NP 难问题,而启发式算法容易陷入局部最优,因此,本文提出一种基于深度强化学习的 VNF 迁移方法,通过结合 VNF 的复制,权衡 VNF 迁移导致的服务质量降级和 VNF 复制造成的成本消耗,选择最优的策略,从而实现在保证网络服务质量的同时,实现服务器的负载均衡并有效降低系统成本。

1.4 本文的组织结构

本文总共分为 5 章,详细的结构安排如下:

第 1 章为绪论。首先介绍了目前网络架构的局限,从而引出网络功能虚拟化与边缘计算的研究意义。然后介绍了目前 VNF 需求预测和迁移相关的国内外研究现状,并对本文主要研究内容作了介绍。最后简要概述了本文的整体结构。

第 2 章是 VNF 需求预测方法与迁移方法相关理论综述。本章介绍了网络功能虚拟化的框架与边缘协同的理论,介绍了流量预测的相关方法、VNF 迁移目前存在的问题以及深度强化学习的相关理论,从而引出本论文的研究意义。

第 3 章提出了基于 SVR 和 GRU 的 VNF 需求预测方法。依据网络边缘流量凸显的数据特征且考虑到网络边缘资源的有限,选择能够提取服务历史数据短期特征的 SVR 模型和提取数据长期特征的 GRU 模型,并赋予不同的权重,从而提高预测准确率,最后通过仿真表明,该方法能够降低预测误差,提高预测准确率。

第 4 章提出了基于深度强化学习的 VNF 迁移。首先通过分析待迁移或复制目标周围节点的空闲资源和负载状态信息,并按照空闲资源大小排序筛选出合适的目标节点。然后协同 VNF 的复制,依据优化的目标,构建马尔科夫决策过程,并通过深度 Q 网络(Deep Q Networks, DQN)训练学习,选择最优的策略。最后通过仿真验证,所提方法能够在保证网络服务基本质量的同时,可以以较小的成本实现服务器的负载均衡。

第 5 章为结束语。该部分内容主要阐述了本文主要研究工作,并针对性地提出本文研究内容存在的一些不足之处。除此之外,依据本文研究存在的不足和当前研究趋势提出了关于未来研究方向的一些建议。

第 2 章 相关理论综述

本章首先阐述网络功能虚拟化与边缘协同技术，然后介绍常见的网络流量预测方法与目前 VNF 迁移存在的问题，最后介绍了深度强化学习相关理论。

2.1 网络功能虚拟化与边缘计算

2.1.1 网络功能虚拟化

网络功能虚拟化和边缘计算是加速 5G 网络的两种有前景的创新技术^[28]。网络功能虚拟化的实质是将基础硬件设施与运行的网络功能服务分离，使软件和硬件的运行能够相对独立，管理和升级也都更加便捷。以往网络功能运行在专有的硬件设备之上，软件与硬件联系紧密，难以分开管理，在系统需要更新或维护时，变得异常麻烦，可能需要暂停所有业务，而且设计专有硬件的协议并部署在相应的平台上非常困难。总之，随着服务需求的不断增加，服务供应商也需不断更新相关底层物理设备，网络功能虚拟化技术能够将软件与硬件进行解耦，可以在通用商用硬件上以虚拟机或容器的方式部署各种虚拟网络功能如防火墙、入侵检测(Intrusion Detection System, IDS)等，从而有效降低系统的成本，增强网络的可扩展性。网络演变过程如图 2.1。

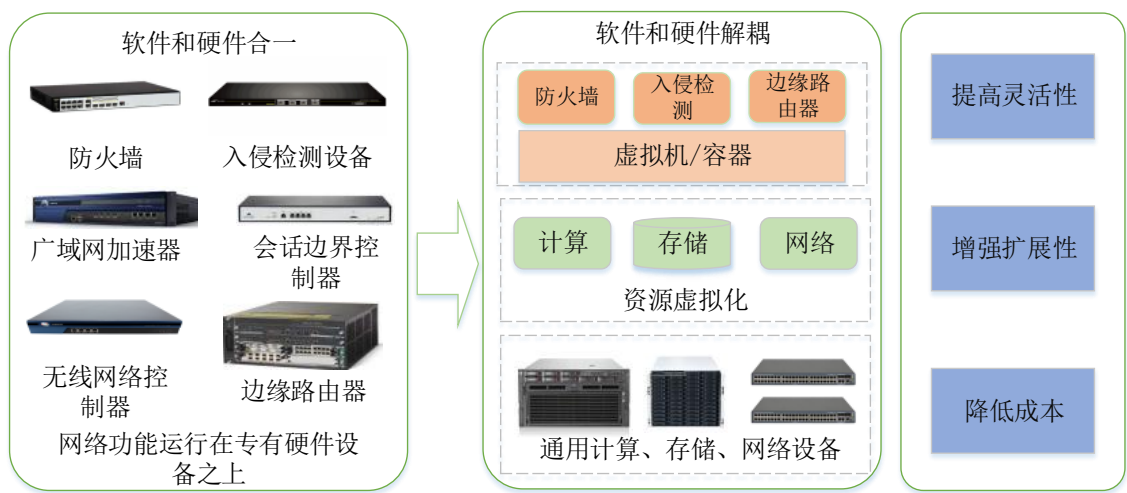


图 2.1 网络演变过程

NFV 体系结构由三个关键要素组成：网络功能虚拟化基础架构(Network

Function Virtualization Infrastructure, NFVI)、VNF 和 NFV 管理与编排(NFV Management and Orchestration, NFV MANO)^[29]。NFV 框架如图 2.2。

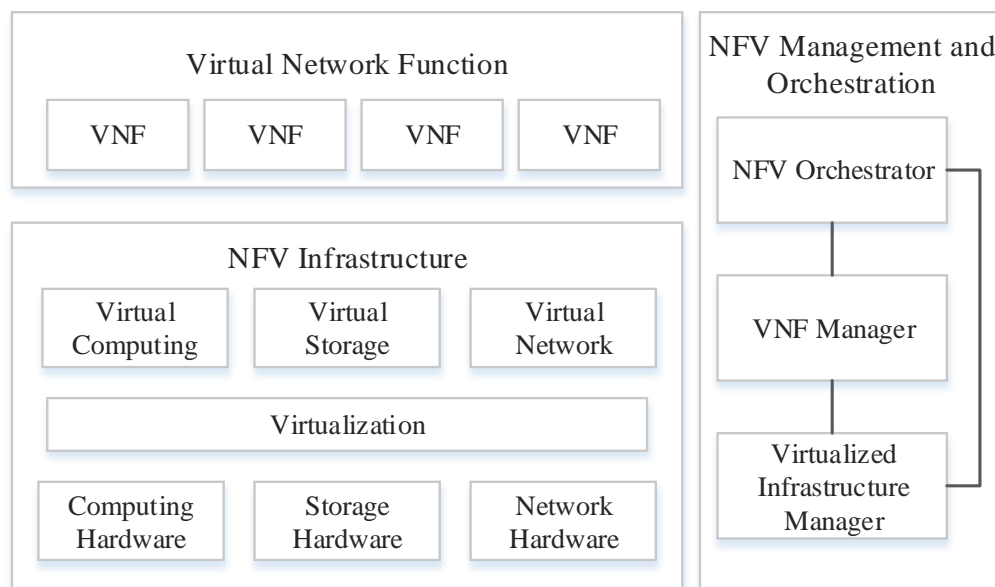


图 2.2 NFV 框架

NFVI 提供 VNF 运行所需要的硬件和软件环境。硬件环境主要是通用的计算、存储和网络设备等，软件环境包含虚拟机管理程序(hypervisor)或容器管理程序，管理 VNF 的运行。NFVI 通过将计算、存储和网络等资源统一在一个共享的虚拟资源池中，以供 VNF 使用。

VNF 是以软件的方式实现的网络功能，它与运行在专有硬件设备上的网络功能没有差别，是一个具体实现的功能，在 NFV 架构中，它运行在 VM 或者容器中。不同的 VNF 按照业务逻辑可以组合成不同网络服务。

NFV MANO 主要职责是在 NFV 框架内管理整个虚拟化上下文。具体而言，上下文包括虚拟化机制、硬件资源编排、VNF 实例的生命周期管理和模块之间的接口管理等^[26]。

借助 NFV 和软件定义网络(Software Defined Network, SDN)，可以在链中设计和配置 VNF，并灵活、高效地将它们部署在一个或多个网络服务器上，这种方法称为服务功能链^[30]。服务功能链上的虚拟网络功能，在逻辑上相连，为用户提供一个完整的网络服务，并且提供的网络服务可以很便捷高效地实现更新，如需要在服务功能链某处添加或删除一个 VNF，可以以较低的成本快速完成更新。

随着研究人员对 NFV 研究的不断深入，有不少研究人员研究了在边缘计算模

式下,如何结合 NFV 技术,从而更好地为用户服务。文献[31]介绍了 NFV 框架以及部署 NFV 的虚拟化平台的基本组件、物理服务器、系统管理程序和客户虚拟机,讨论了在移动网络中使用它的挑战和要求,并提出了一种标准以将虚拟化演进的分组核心的多种功能捆绑在单个物理设备或一组相邻设备中。文献[32]为了以低延迟处理大量数据和提高资源的利用率,在基于虚拟本地集线器(Virtual Local Hub, VLH)思想的基础上,提出了一个完整的边缘计算框架,该框架将 NFV 技术应用于物联网应用的边缘计算环境。NFV 技术应用在边缘网络已引起越来越多的关注。

2.1.2 边缘协同

边缘计算是一种利用网络边缘的计算、存储等资源来完成计算的新型模型^[3],旨在将部分计算、网络和存储等资源下沉在网络的边缘,从而分担云中心的服务压力。边缘计算利用融合计算、存储、应用核心能力等资源的平台,可以在终端附近完成对数据的处理,以降低时延,节省带宽资源,满足实时智能应用、网络业务以及如远程手术、自动驾驶等低时延应用的基本需求。

在物联网中,边缘计算资源主要由边缘设备和边缘服务器组成^[33]。由于海量终端设备及相关应用的涌现,需要部署的边缘服务器也逐渐增加,而在众多的边缘节点中,每个节点承担的任务也不尽相同,有些边缘服务器节点处于严重负载状态,而有些则可能处于极为空闲状态,无事可做,造成资源的严重浪费。通过边缘协同,可以加强服务器节点与周围节点以及与云中心之间的联系,实现资源的有效调度以及服务器节点的负载均衡,避免资源的过度浪费。

边缘计算中的协同主要有边缘与边缘之间的协同、边缘与云之间的协同,边缘与设备之间的协同。边缘与边缘之间的协同能够分担单一边缘节点的计算压力,可以实现负载均衡^[34]。

边缘与边缘协同:边缘与边缘之间的协同主要利用同处于边缘层的节点,使这些边缘节点之间相互配合,共同解决需要解决的问题。针对不同的场景,边缘节点之间的协作的方式也不完全相同。例如在计算卸载中,相对庞大的用户请求,单一的边缘节点资源是非常有限的,在单一的节点负载严重,无法满足当前节点覆盖范围下的所有用户请求时,可以利用周围其他节点的空闲资源,彼此之间相互配合,可以满足更多的用户请求。也可以通过边缘协同,利用多个节点的算力,共同完成

复杂的神经网络的训练。图 2.3 为边缘与边缘协同的示意图。

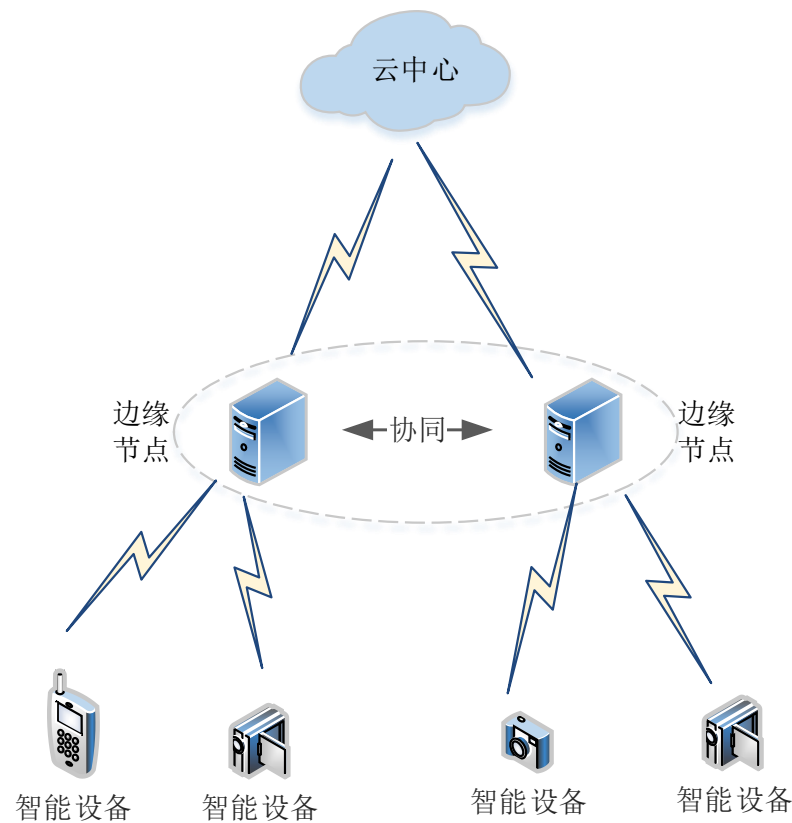


图 2.3 边缘与边缘协同示意图

边缘与云协同：边缘计算是相对云计算而言的，但不能取代云计算，边缘计算与云计算相辅相成，协同运作，边缘侧负责处理其覆盖范围下的计算、存储等工作，而云中心由于资源充足，负责一些大任务的计算，通过边云之间的协同，才能更有效地发挥它们的作用。

边缘与设备协同：边缘与设备之间的协同更多是在物联网场景中，万物互联时代，各种具有一定运算、存储等能力的智能终端涌现，因此，在一些任务中，智能终端设备可以与边缘云进行协作，共同完成，如由终端采集数据、而由边缘负责处理。

2.2 网络流量预测方法

对于网络流量的预测，目前已经有不少相关的研究，主要有传统时间序列建模的预测方法如自回归 (Autoregressive Model, AR) 模型、自回归移动平均 (Autoregressive Integrated Moving Average mode, ARIMA) 模型等。此外，也有许多

关于非线性的预测模型如支持向量回归、基于长短期记忆神经网络的预测模型等。传统模型虽然复杂度较低,但不能够准确刻画网络流量特征,预测准确率较低。由于目前网络的复杂性以及流量凸显的复杂特征,单一的模型对于预测结果的提升并不突出^[35],一种可行的方案是针对不同的场景,结合不同模型的优势,以提升预测准确率。

2.2.1 传统预测模型

在传统方法中,自回归移动平均模型,是目前比较常见的一种网络流量预测模型。 $ARIMA$ 模型可以表示为 $ARIMA(p,d,q)$, 是由自回归、滑动平均(Moving Average, MA)以及差分法三者融合而成的。 $ARIMA(p,d,q)$ 中 p 是自回归项数, q 是滑动平均项数, d 是数据差分的阶数^[36, 37]。

(1) 自回归模型: 在回归模型中, 目标值 y_t 是作为时间 t 的特征值的函数, 而与其他时间的 y 值不相关。在满足协方差平稳的序列中, 将 y_t 与前 p 个之前的 y 值关联, 利用变量本身的历史数据, 从而实现预测。

(2) 滑动平均模型: 将 y_t 与前 c 个误差项关联, 可以认为是前 q 个误差项的加权累加, 这种方式能够有效消除预测中的随机波动。

(3) 集成模型: 差分可以使时间序列平稳。

2.2.2 支持向量回归

支持向量机(Support Vector Machine, SVM)最开始主要用于解决分类的问题^[38]。近些年来, SVM 也被用于解决一些回归问题。对于样本 (x, y) , 传统回归模型的损失一般是通过计算模型输出 $f(x)$ 与真实输出值 y 之间的差值, 仅当 $f(x)$ 与 y 完全相等时, 损失才为零。而 SVR 则可以容忍 $f(x)$ 与 y 之间的差值有部分偏差, 只有在 $f(x)$ 与 y 之间的差值绝对值大于 ε 的时候才进行损失的计算, SVR 偏差如图 2.4 所示。

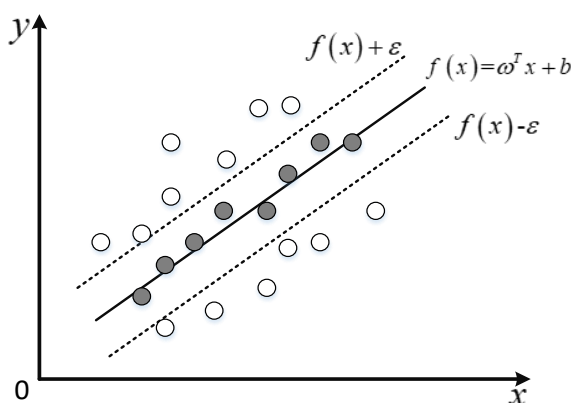


图 2.4 支持向量回归偏差示意图

SVR 以 $f(x)$ 为中心, 构建了一个 2ε 大小的间隔, 当数据落在此间隔之内, 说明预测是正确的。依据文献[39], SVR 的问题可以形式化为:

$$\min_{\omega, b} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^m \ell_{\varepsilon}(f(x_i) - y_i) \quad (2.1)$$

其中 ω 为法向量, ω 和 b 都是待确定的模型参数, C 是正则化常数, m 表示样本数目, ℓ_{ε} 表示 ε -不敏感损失函数, 可以表示为:

$$\ell_{\varepsilon}(z) = \begin{cases} 0, & \text{if } |z| \leq \varepsilon \\ |z| - \varepsilon, & \text{otherwise} \end{cases} \quad (2.2)$$

SVR 在少量样本的时 P 间序列预测中具有较好的性能, 泛化能力较强。

2.2.3 门控循环神经网络

循环神经网络(Recurrent Neural Network, RNN)是一种特殊的神经网络, 循环代表着同一网络结构被反复使用。相比于传统的神经网络, RNN 在隐藏层中加入了误差反馈机制, 即 RNN 当前隐藏层的输出包括上一时刻隐藏层的输出信息, 代表了 RNN 利用循环反馈机制保留了上一时刻的输出信息, 因此 RNN 具有短期记忆的能力, 从而使 RNN 模型在处理时间序列上以及自然语言处理方面具有很大的优势。

虽然循环神经网络能够记忆之前的数据特征, 但是由于反向误差传播的累叠, 会出现梯度爆炸或梯度消失的问题, 从而不能够记忆较远的神经单元, 为解决梯度的爆炸和消失问题, 长短期记忆网络^[40]和门控循环神经网络被设计出来, 它们能够解决 RNN 存在的长期依赖的问题^[41]。LSTM 主要通过单元内部设置遗忘门、

输入门和输出门来实现数据信息的控制，达到长期记忆的效果。

LSTM 虽然能够很好地处理长序列数据的影响，在具有长期依赖性的时间序列预测上有较为准确地预测，但是其内部结构单元有三个门，结构复杂，参数较多，很难直接确定，需要训练的时间长，也容易陷入局部最优。门控循环单元是 LSTM 的一种变体，在预测效果上与 LSTM 相差无几，而且其内部单元只有两个门，结构上较 LSTM 简单，计算复杂度低。GRU 内部结构如图 2.5。

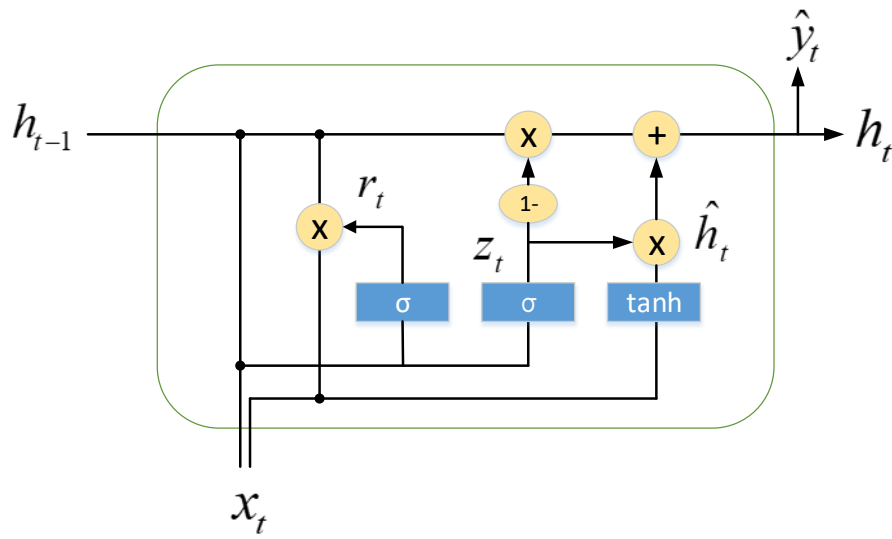


图 2.5 GRU 内部结构

依据相关文献[42]，GRU 单元内部相关的公式可表示为：

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (2.3)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \quad (2.4)$$

$$\hat{h}_t = \tanh(W_{\hat{h}} \cdot [r_t * h_{t-1}, x_t]) \quad (2.5)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t \quad (2.6)$$

其中 z_t 是门控信号，范围为 0~1， W_z 、 W_r 和 $W_{\hat{h}}$ 表示模型要学习的参数， r_t 表示重置门， $(1 - z_t) * h_t$ 表示对原本隐藏状态的“遗忘”， $z_t * h_t$ 表示对包含当前节点信息进行选择性“记忆”。

2.3 VNF 迁移概述

在启用 NFV 的基础架构中，其中大量的 VNF 用于服务网络服务，通过利用资源整合，可以最大程度地减少物理资源的使用^[26]。如今服务提供商在网络中部

署有各种 VNF，它们彼此间相互联系，为用户提供不同的网络服务。而在流量突增的情况下，服务器很可能会负载严重，从而导致服务质量急剧下降，难以保证用户的服务体验质量。在部署有虚拟网络功能的服务器需要维护或升级时，会导致服务的中断，无法为用户提供相应的服务。因此，为保证网络服务不被中断或中断的时间尽可能短，需要进行 VNF 的迁移。通过 VNF 的迁移，可以将 VNF 迁移至其他具有空闲资源的服务器上，实现服务器的负载平衡，在服务器进行升级时，也可以保障网络服务的运行。

2.3.1 冷迁移

服务提供商在网络中部署的虚拟网络功能可以运行在 VM 中，也可以运行在容器中。VNF 迁移问题主要由从传统的 VM 迁移问题演变而来的^[28]。因此，VNF 的迁移可以借助 VM，从而实现 VNF 从一个服务器到另外一个服务器的迁移，VM 的迁移有实时迁移和冷迁移。

冷迁移是指在迁移目标之前，保存当前的状态，将 VM 进行关机，之后再完成迁移，由于冷迁移需要先关闭 VM，迁移至目标节点之后还需要开启，完成这个过程会导致高的时延，因此，冷迁移并不适用于低时延的一些应用。但是，大多数部署在网络边缘部署 VNF 以降低时延为目的，因此，VNF 迁移一般都是实时迁移。

2.3.2 实时迁移

VNF 的实时迁移可分为复制前迁移、复制后迁移以及两者的结合^[43]。复制前迁移方式是先复制 VM 中内存内容，将这部分内容传输至目标节点之后，再将处理器的状态信息复制到目标节点。复制后迁移则是先将处理器的状态信息复制到目标节点，之后再复制 VM 内存中的内容到目标节点。

2.3.3 VNF 迁移算法研究

目前国内外许多研究人员已经对 VNF 迁移问题做了相关研究。文献[35]指出了当前研究 VNF 迁移需要注意的问题：(1) 迁移的时机问题，即如何确定在什么时间开始迁移；(2) 待迁移的目标选择问题，即在进行迁移时，如何确定需要迁移

的 VNF；(3) 迁移目标节点问题，即在将某个 VNF 进行迁移的时候，如何确定该 VNF 迁移到哪个节点。(4) 迁移的路径选择问题，即在进行迁移时，如何确定迁移过程中的路径。

现有的大多数研究都聚焦在优化 SFC 端到端时延、系统能耗以及两者的联合。如文献[44]主要考虑低时延网络中的 VNF 迁移问题，提出了一种 VNF 实时迁移的新颖算法，以在快速变化的资源状态下降低网络延迟。文献[45]通过对网络能耗最小化及 SFC 端到端时延最小化进行联合优化，提出了一种基于深度确定性策略梯度的 VNF 迁移方案。

VNF 迁移策略模型的求解被证明是一个 NP 难问题，由于网络环境的复杂性以及网络服务的多样性，导致 VNF 迁移策略的状态、动作空间较大，很难用线性规划和启发式算法求取最优解。

由于 VNF 的迁移会导致服务质量的下降，一种可行的方案是结合 VNF 的复制，通过 VNF 副本的快速复制，可以使计算分流，保障 SFC 的端到端时延。不过目前对协同 VNF 复制的 VNF 迁移研究较少。

2.4 深度强化学习

强化学习(Reinforcement Learning, RL)是通过智能体与环境进行交互，并从中学习的一种方式。强化学习的目标是找到一个最优的策略，使得智能体在与环境的交互中获取最大的奖励。智能体与环境交互的过程如图 2.6。

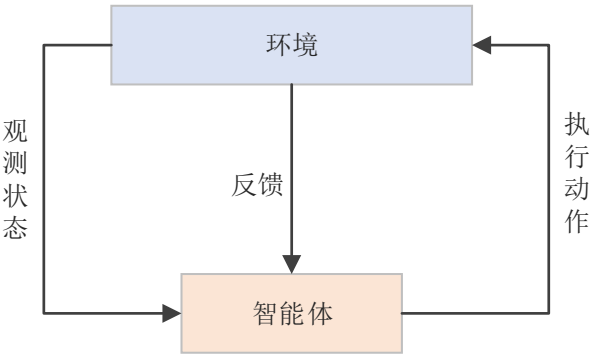


图 2.6 智能体与环境交互过程

具体来说，智能体监测当前环境的状态，并依据当前的环境状态按照某种策略选取一个动作，该动作执行后，当前环境会响应智能体执行的动作，同时智能体会得到一个奖励，并在不断地与环境交互过程中，学习到一个最佳策略。

2.4.1 基于值函数的 Q 学习方法

Q 学习(Q-learning)方法是一种基于值函数的强化学习算法。Q 学习方法是通
过创建一个 Q-table 来表示状态-动作值函数 $Q(s,a)$ ，并根据 Q 值选取可以获取最
大奖励的动作[46]。其中 Q-table 的每一列代表动作空间的动作，而每一行表示状
态。 $Q(s,a)$ 的值表示在每个的状态下，不同动作相应的累积奖励的期望，即：

$$Q(s,a)=E[\sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'} | s_t = s, a_t = a] \quad (2.7)$$

其中 s 表示环境的状态， a 表示执行的动作。Q 值越大，相应地执行该动作获取的
累积奖励就越大。若策略 π^* 为最优策略，则该策略学习最佳动作值函数 $Q^*(s,a)$ 输
出为：

$$\pi^*(s)=\arg \max_a Q^*(s,a) \quad (2.8)$$

最优值函数满足最优贝尔曼方程[47]，则最优值函数可以表示为：

$$Q^*(s,a)=E[r+\gamma \max_{a'} Q^*(s',a') | s,a] \quad (2.9)$$

其中 r 表示智能体在状态转移过程中得到的奖励， γ 表示折扣因子， s' 表示下一个
时隙任意可能的状态， a' 表示在 s' 状态时可执行的动作。Q 学习通过不断地依据
每次选择所对应的回报和下一个状态的所有 Q 值来更新 Q-table，更新公式为：

$$Q_{i+1}(s,a)=\leftarrow Q_i(s,a)+\lambda(r+\gamma \max_{a'} Q_i(s',a')-Q_i(s,a)) \quad (2.10)$$

其中 λ 是学习率，当 i 趋于无穷时， $Q(s,a)$ 将会收敛于最优状态-动作值 $Q^*(s,a)$ ，
实际中一般设置一个比较大的迭代次数，迭代后的估计值可以作为 $Q^*(s,a)$ 的近似。

2.4.2 深度 Q 网络算法

传统的强化学习算法可以处理简单环境任务，很难解决复杂任务环境的问题。
随着深度学习与强化学习的不断完善，强化学习和深度学习结合的研究愈来愈多。

传统的 Q 学习算法是基于表格来存储状态-动作值，对于状态较少的问题，可
以较为高效便捷地解决。但对于状态空间较大的问题，Q 学习中的 Q 值以表格的
方式进行存储，其维度将会急剧增加，可能出现“维数灾难”的问题，因此传统的
Q 学习算法难以解决状态空间过大的问题。DQN 算法的基本思想与 Q 学习算法一

致，不同之处在于 DQN 算法的 Q 值不是由状态-动作对来计算，而是通过神经网络来计算^[48]，DQN 属于基于值函数的深度强化学习。

深度 Q 学习在传统 Q 学习算法上做的改进主要有使用了经验回放机制，目标网络和缩小奖励值范围。

(1) 经验回放机制。智能体执行动作 a 之后，环境当前的状态 s 会转换为下一个状态 s' ，智能体会得到相应的一个回报 r ，将每次智能体执行动作之后得到的转移状态组 (s, a, r, s') 放入经验池中，训练时从经验池中随机选取样本进行学习，并更新相应的参数。

(2) 目标网络。深度 Q 学习利用神经网络作为状态-动作值函数逼近器的同时，也使用了一个目标网络来产生目标 Q 值。每经过一定的时间步，复制当前值网络的权重 θ 给目标网络，使得目标网络的参数与当前值网络参数一致，并通过最小化值网络输出与目标网络输出之间的误差来更新网络的权重。使用目标网络能够让目标 Q 值在一定时间内保持不变，可以降低当前值与目标值之间的关联性，在一定程度上提高了算法的稳定性和性能。

(3) 缩小奖励值范围。奖励的大小可能会影响学习的稳定性，为保障学习的稳定，应该将奖励值设置在较小范围。因此，在深度 Q 算法中将奖励值范围缩小 $[-1, 1]$ 区间内，保障学习时的稳定性。

输入为 2 维，输出为 4 维，隐藏层 3 层的 Q 网络结构如图 2.7。环境的状态作为 Q 网络的输入，依据输入环境的状态，通过遍历此状态下各种可执行动作的价值，并将其输出。

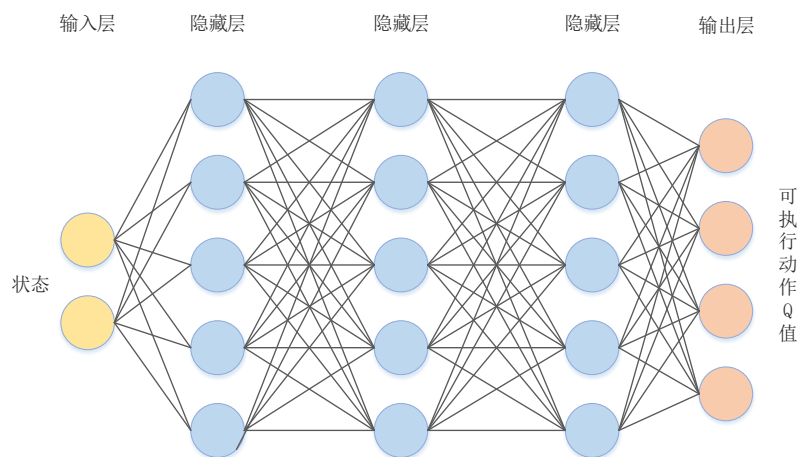


图 2.7 Q 网络结构示意图

2.5 本章小结

本章首先介绍了网络功能虚拟化技术和边缘计算中的协同技术。然后介绍了目前网络流量预测方法的现状以及相关的模型，之后又介绍了如今 VNF 迁移研究方面存在的一些问题。最后，根据本文研究的内容，对深度强化学习相关的一些理论作了介绍。

第3章 边缘网络中 VNF 的需求预测

3.1 系统模型及问题描述

万物互联时代,边缘计算和网络功能虚拟化等技术迅速发展,使得海量边缘智能终端设备以及相关网络业务涌现,同时也促进了网络逐步向智能化和虚拟化的方式转变。通过 NFV/SDN 等技术实现软件和硬件的解耦^[49],能够解决当前网络布局复杂、僵化的问题,提高网络的敏捷性和可扩展性,实现网络服务快速便捷的升级。且随着“边缘智能”、“智慧城市”等的逐步推动,服务提供商将愈来愈多的网络业务迁移至网络边缘,利用边缘基础设施虚拟化的计算、存储和网络资源进行数据的处理,能够降低计算时延,节省带宽资源,从而为终端用户提供更加优质的服务并有效降低系统成本。

当前,边缘的各类网络功能设备由各个设备独立实现,很难互相进行操作,无法满足基于某种商业规则控制用户的各类业务,例如机顶盒、接入网关设备和数据包深度监测(Deep Packet Inspection, STB)等,而边缘计算的兴起与网络功能虚拟化技术的迅速发展带来了一系列的改革,网络运营商借助 NFV 技术可以在网络边缘放置不同类型的 VNF,为用户提供相关的网络服务,以在降低系统成本的同时,可以提供高效的服务质量。但是,在网络边缘实例化各种 VNF 是需要时间的,放置相应 VNF 实例的数量是非常重要的,放置过多的 VNF 实例会造成系统资源的浪费,增加系统成本,而当放置的 VNF 实例过少时,会导致用户任务计算时延增大,降低网络服务质量。为能够实现 VNF 的可伸缩性,本章提出了一种新颖的 VNF 需求预测方法。

VNF 需求预测系统模型如图 3.1。在边缘节点中,依据不同网络服务的历史服务数据,通过构建一个 VNF 预测模型,提取历史数据的相关特征,预测未来一段时间内的不同 VNF 需要处理的流量,并进一步分析 VNF 的需求,从而可以提前放置相应的 VNF,实现 VNF 及网络服务的弹性部署,提高资源利用率、并降低系统的成本。

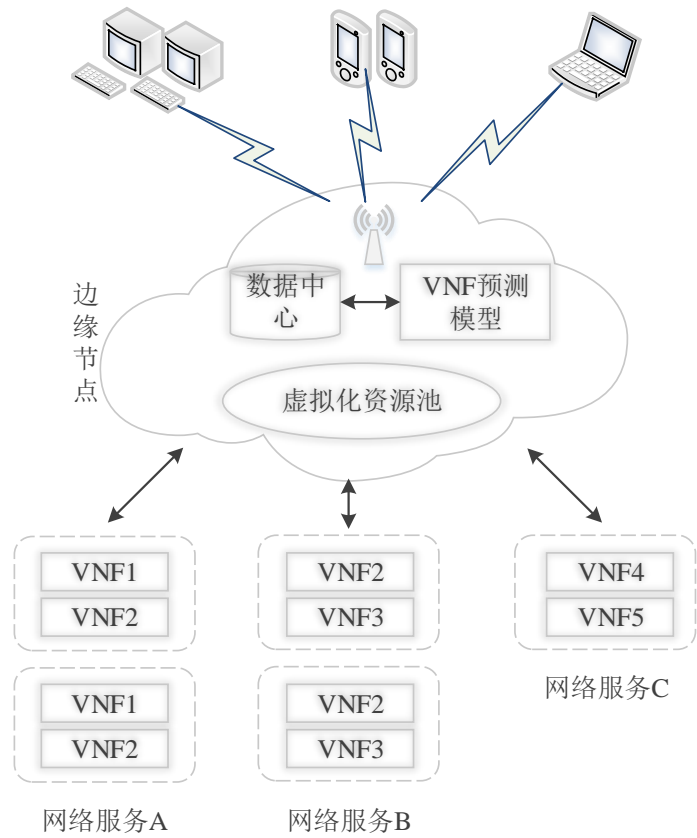


图 3.1 VNF 需求预测系统网络模型

3.2 基于 SVR 与 GRU 的组合预测模型

3.2.1 边缘网络业务流量特征

随着大量移动设备、各种实时网络业务的增多，导致边缘网络流量的不确定性。依据文献[50]、[51]的描述，边缘网络中的流量特征与接入的用户行为相关，随着网络中大量实时业务、新型应用的激增，增大了边缘网络中流量的不确定性，使得边缘网络中流量凸显出波动性大、突发性强的特点，且具有一定的自相似特征。自相似性是一种描述流量特点的统计特征，用于描述流量时间序列在不同时间尺度上存在突发性且不会随着网络的规模、拓扑及应用的变化而变化的特点^[50]。针对网络边缘流量表现出的特点，选取适合的模型进行 VNF 的预测。

3.2.2 SVR 模型

由于边缘实时网络业务流量的突发性、长相关性等特点，传统的如自回归、移

动平均以及自回归滑动平均模型(Autoregressive moving average model, ARMA)等线性预测模型虽然在计算复杂度上较低,但是不能够准确反映边缘真实网络业务流量非线性的特点,在预测精度上不如非线性预测模型。

支持向量回归可用于非线性的短期时间序列预测,且可不需要大量的样本就能够较为准确的预测,也不容易陷入局部最优解,泛化能力强,但是在分析大量的时间序列数据上,并不能够很好的提取到这些时间序列长期依赖的某些特征。

3.2.3 GRU 模型

RNN 是基于普通多层的反向传播(back propagation, BP)神经网络,再通过一个权重矩阵,使隐藏层的各单元间能够横向联系,能够记忆之前的神经单元的输出值信息,并将这些信息输入到当前的神经单元,从而使 RNN 模型在处理时间序列上以及自然语言处理方面具有很大的优势。

而 GRU 作为循环神经网络中的一种,本身就具备记忆功能,能够较好地分析大量的具有长期依赖性的时间序列,在神经网络层数可控范围内又能够消除 RNN 存在的梯度消失和爆炸的问题,且 GRU 相对 LSTM 来说,结构简单,适合在计算资源有限的网络边缘节点进行放置运行,但对于短期内的突发性特征不能够很好地处理。

3.2.4 SVR-GRU 模型

为提升预测效果,一方面通过 SVR 对少量历史数据进行学习,较好地提取网络业务流量的短期数据特征;另一方面,利用 GRU 模型处理长期时序数据,提取网络业务流量的长期特征,而且两种模型都具有较低的复杂度,适合在资源有限的边缘网络中运行。通过有效结合 SVR 和 GRU,为其赋予不同的权重,构建 SVR-GRU 模型,提取数据短期内的突发性特征和长期特征,以提升 VNF 需求预测效果。

若 $F_S(t)$ 表示支持向量回归的预测结果, $F_G(t)$ 表示门控循环神经网络的预测结果,则 SVR-GRU 模型预测结果 F_{S-G} 可表示为:

$$F_{S-G} = W_S \cdot F_S(t) + W_G \cdot F_G(t) \quad (3.1)$$

其中, W_S , W_G 分别为 SVR 模型和 GRU 模型的权重,且满足

$$\begin{cases} W_S + W_G = 1 \\ W_S \geq 0, W_G \geq 0 \end{cases} \quad (3.2)$$

采用方差倒数法也即误差平方和法, 旨在依据不同模型预测精度的不同, 确定相应模型的权重。预测模型精度越高, 则其在组合预测中所占有的权重越大, 反之, 越小。于是不同预测模型的权重可进一步表示为:

$$w_k = \frac{D_k^{-1}}{\sum_{k=1}^K D_k^{-1}}, \quad K = 2 \quad (3.3)$$

其中 D_k 为第 k 个模型的误差平方和, D_k 可表示为:

$$D_k = \sum_{t=1}^n (x_t - \hat{x}_t(k))^2, \quad k = 1, 2 \quad (3.4)$$

其中 x_t 为实际值, $\hat{x}_t(k)$ 为第 k 个模型的预测值, n 为预测的数据数量。若 D_{SVR} 和 D_{GRU} 分别表示 SVR 和 GRU 模型的误差平方和, 则进一步可计算 SVR 模型与 GRU 模型权重为:

$$\begin{cases} W_S = \frac{D_{SVR}^{-1}}{D_{SVR}^{-1} + D_{GRU}^{-1}} \\ W_G = \frac{D_{GRU}^{-1}}{D_{SVR}^{-1} + D_{GRU}^{-1}} \end{cases} \quad (3.5)$$

由于不同权重确定数据组计算出的权重会有不同, 为避免单组数据计算导致的偶然性, 通过计算多组权重的平均值, 将均值作为最终 SVR-GRU 模型的权重。若确定权重的数据组数为 N , SVR 模型与 GRU 模型权重进一步表示为:

$$\begin{cases} W_S = \frac{1}{N} \sum_{i=1}^N W_S^i \\ W_G = \frac{1}{N} \sum_{i=1}^N W_G^i \end{cases} \quad (3.6)$$

其中 W_S^i 表示由 i 组权重确定数据计算的 SVR 模型权重, W_G^i 表示由 i 组权重确定数据计算的 GRU 模型权重。

最终预测效果用均方根误差(Root Mean Squard Error, RMSE)表示, 它表示误差的平方期望值, RMSE 值越小则代表预测效果越好。

若未来 n 个时刻的预测值之和为:

$$\sum_{t=1}^n \varphi(t) = \sum_{t=1}^n (W_S \cdot F_S(t) + W_G \cdot F_G(t)) \quad (3.7)$$

则 SVR-GRU 组合模型预测的均方根误差为:

$$RMSE_{SVR-GRU} = \sqrt{\frac{\sum_{t=1}^n |\varphi(t) - y(t)|^2}{n}} \quad (3.8)$$

其中 $y(t)$ 为实际值, $\varphi(t)$ 为 SVR-GRU 组合模型预测值。

模型权重确定的具体过程如表 3.1。

表 3.1 SVR-GRU 组合模型的权重确定算法

算法 1: SVR-GRU 组合模型的权重确定算法	
1.	输入: 训练完成的 SVR 模型; 训练完成的 GRU 模型; N 组权重确定数据; 初始化索引 $i=1$;
2.	输出: 权重 W_S 和 W_G ;
3	根据 SVR 模型与权重确定数据进行预测得到预测结果 F_{SVR} ;
4.	根据 GRU 模型与权重确定数据进行预测得到预测结果 F_{GRU} ;
5.	由 SVR 模型预测结果 F_{SVR} 计算得到误差平方和 D_{SVR} , 由 GRU 模型预测结果 F_{GRU} 计算得到误差平方和 D_{GRU} ;
6.	由公式 (3.4) 和 (3.5) 计算出 SVR 模型的权重 W_S^i 和 GRU 模型的权重 W_G^i , 并保存相应的结果;
7.	若满足 $i=N$, 则停止循环, 执行步骤 9, 若不满足 $i=N$, 则执行步骤 8;
8.	令 $i=i+1$ 重新从步骤 3 开始执行;
9.	依据公式 (3.6) 计算出 SVR 模型的权重 W_S 和 GRU 模型的权重 W_G 并输出;

3.2.5 模型复杂度分析

对于神经网络模型的复杂度来说, 计算的复杂度影响着模型训练的时间, 模型复杂度过高, 则模型需要大量的时间进行训练, 无法实现快速预测。

由文献[52]、[53]可知, 一般情况下, SVR 使用精确搜索算法的时间复杂度为 $O(l^3)$, 使用不精确算法的时间复杂度则是 $O(l^2)$, 在实践中, 选择满足一定要求的不精确的搜索算法可以使时间的复杂度为 $O(l^2)$, 其中 l 为训练样本数量。RNN 执行 n 次序列操作后的总时间复杂度为 $O(n \times d^2)$, 其中 d 表示维度。由于 SVR 和 GRU 模型可以并行训练, 时间复杂度可以认为是 $O(n \times d^2)$ 。

对于空间复杂度来说, 若 m 为样本输入维数, 则支持向量回归空间复杂度一般为 $O(2 \times l \times (m+1) + 2 \times (l+1)^2 + 2 \times (l+1))$ 。LSTM 单元拥有输入门、输出门、遗忘门

和候选态,若隐藏层单元数目为 n ,则 LSTM 的空间复杂度为 $O(4 \times (n \times m + n^2 + n))$ 。GRU 内部单元则拥有比 LSTM 少的参数,通过维护一个参量控制信息的遗忘与更新,因此空间复杂度可表示为 $O(n \times m + n^2 + n)$ 。由于时间序列输入样本的维度 $m \ll l$,将常量 m 作为常数 1 消除,SVR 的空间复杂度可进一步表示为 $O(2 \times l^2 + 6 \times l)$,同理,LSTM 和 GRU 的空间复杂度分别为 $O(4 \times n^2 + 8 \times n)$ 和 $O(n^2 + 2 \times n)$ 。于是,SVR-GRU 组合预测模型的空间复杂度可表示为 $O(n^2 + 2 \times n + 2 \times l^2 + 6 \times l)$ 。由于 SVR 模型的训练样本数量较少,一般情况下 LSTM 模型隐藏层单元数目较于 SVR 训练样本数量近似相等或比其更大,此时,SVR-GRU 模型的空间复杂度会更低一些。SVR 训练样本数量为 l , n 为 LSTM 隐藏层单元数量,则 SVR-GRU 模型与 LSTM 模型空间复杂度对比如表 3.2。

表 3.2 SVR-GTU 与 LSTM 的空间复杂度对比

模型	空间复杂度	条件
LSTM	较大	$l \approx n$ 或 $l < n$
SVR-GRU	较大	$n < l$

3.3 VNF 需求分析

通过 SVR-GRU 模型对边缘节点未来一段时间内不同网络服务需要处理的请求进行预测,为 VNF 需求分析提供相应的数据支撑。

不同的网络服务由不同 VNF 组合,也即由不同的 VNF 按照业务逻辑组合成不同的网络服务功能链,且不同网络服务可能包含相同类型的 VNF。因此,利用预测出的不同网络服务需要处理的请求,能够计算出每种 VNF 需要处理的请求。

若集合 $S = \{s_1, s_2, \dots, s_m\}$ 为通过 SVR-GRU 模型预测出的 m 种网络服务需要处理的请求量,对于 m 种网络服务,每种网络服务由不同的 VNF 组成,若所有网络服务所包含的 VNF 类型集合为 $V = \{V_1, V_2, \dots, V_n\}$,则组成每种网络服务的 VNF 类型集合为 F_k , F_k 包含 n 种 VNF 中的一种或多种,而每种 VNF 单位时间内处理请求的数量为 $h = \{h_1, h_2, \dots, h_n\}$,则每种 VNF 在某段时间内要处理的请求数量 R_n 为:

$$R_n = \sum_{k=1}^m s_k | (V_n \in F_k), n = 1, 2, 3 \dots \quad (3.9)$$

在边缘节点资源足够时, 若 $v = \{v_1, v_2, \dots, v_n\}$ 为每种 VNF 的所需数量, 则每种 VNF 需求量为:

$$v_n = \frac{R_n}{h_n} = \frac{\sum_{k=1}^m s_k | (V_n \in F_k) |}{h_n}, n=1, 2, \dots \quad (3.10)$$

由于网络服务中同类型的 VNF 可能是多个, 与具体业务逻辑相关, 可以进一步通过不同网络服务功能链中同种 VNF 数量占比进行计算, 即:

$$\hat{v}_n^k = v_n * \frac{v_n^k}{\sum_{k=1}^m v_n^k | (V_n \in F_k) |}, n=1, 2, \dots \quad (3.11)$$

v_n^k 为网络服务 k 中类型为 n 的 VNF 数量。利用相应的分析结果可以在边缘节点提前部署 VNF, 以降低系统成本, 提高网络服务质量。

若边缘节点上的计算、网络和存储资源不足以支撑其覆盖范围下的所有用户请求, 则需将部分 VNF 放置在周围其他边缘节点, 才能保证所有任务能够被及时处理。

3.4 仿真结果与性能分析

3.4.1 实验设计

1. 实验设计

为验证组合预测模型预测的效果, 在本章实验中, 设置了传统方法以及 LSTM 模型与 SVR-GRU 模型预测的预测效果对比图以及相应预测模型的 RSME 指标对比等。

2. 数据集

模型训练的数据来自 Kaggle 平台提供的数据集, 数据是有关 4G 蜂窝流量数据, 当用户使用移动网络服务时, 附近的 4G 基站将为移动设备提供服务。一小时内小区服务的所有用户的总数据容量称为该小区在 1 小时内的流量。数据集收集的数据包含多个单元, 数据是单元每个小时的流量, 约一年的数据, 共有 365×24 个数据。流量的性质随时间的变化而变化, 且不同时间段的数据表现出不同的特征^[54]。每个基站单元可以看作是一个边缘节点, 每个小时内的流量具有网络边缘侧流量数

据波动性大，突发性强等特征，可以作为验证本章方法的数据集。

3. 数据预处理

由于流量数据来自多个基站单元，本章选取三个基站单元流量数据进行相关的仿真。三个月有 24×90 即 2160 个数据，一年有 24×365 即 8760 个数据。三个月的数据记为 D_s ，作为支持向量回归模型的训练及测试，一年的数据记为 D_L ， D_s 用于 SVR 模型的训练及测试， D_L 用于 GRU 模型的训练及测试。整理的数据格式如表 3.3。

表 3.3 数据预处理格式

time	Cell_001801	Cell_001912	Cell_001963
7:00	138.81636	99.50784	90.75102
8:00	165.12386	65.90548	85.78711
9:00	82.23684	72.23638	112.62764
10:00	283.14842	113.5628	431.33294
11:00	707.64202	471.26541	1064.21055
12:00	567.5536	655.32557	645.89437
13:00	386.35251	348.15859	760.08825

模型的训练集数据与测试集数据的比例为 4:1。此外，选取 12 组数据，每组一周数据，作为训练好的 SVR 和 GRU 模型的输入数据，从而进一步依据预测误差和方差倒数法计算出权重，确定模型的权重。

4. 仿真流程

在数据预处理完成之后，需要构建 SVR-GRU 模型并对该模型的预测效果进行有效分析，实验仿真过程如图 3.2 示。

对于每一个基站单元的数据，将预处理之后的数据分为两部分，分别为权重确定数据组和测试数据。其中权重确定数据组作为 SVR 模型与 GRU 模型的输入，依据权重确定数据，分别计算出两种模型相应的预测误差，然后通过方差倒数法得到多组权重值，分别取权重的均值作为最终两种模型的权重。测试数据是为验证与 ARIMA 和 LSTM 模型的对比，得到相应模型的预测结果，计算相应的 RMSE 指标并进行分析。

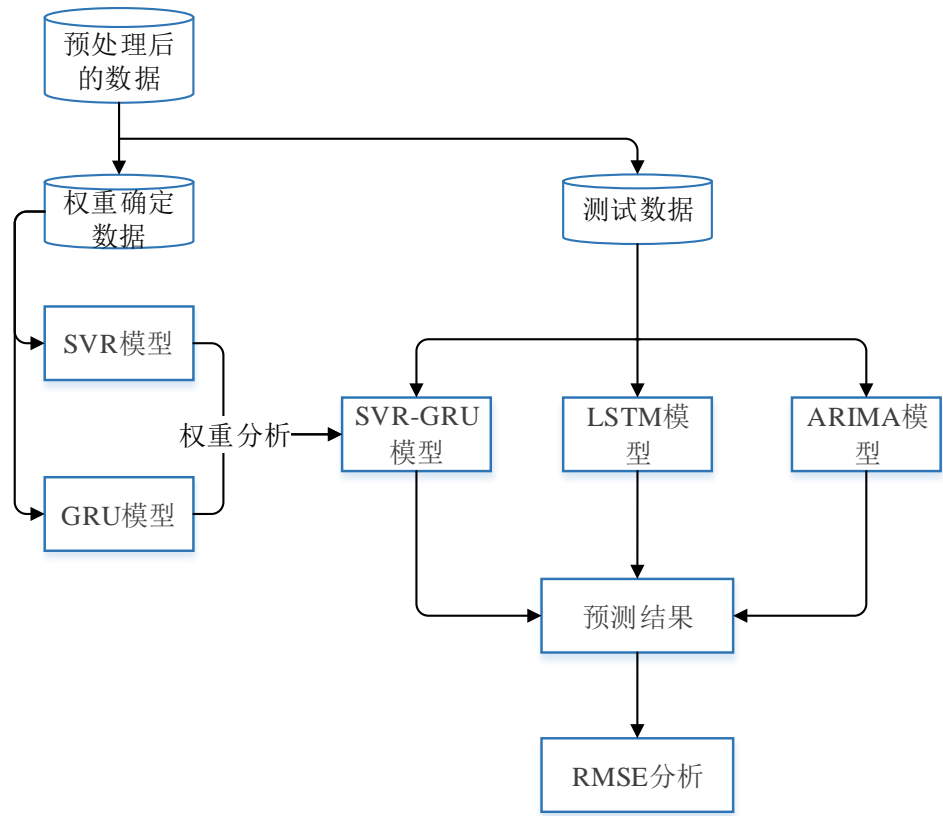


图 3.2 实验仿真过程

3.4.2 结果分析

文献[8]通过分析预测值和实际值对比了文中提出的 CAT-LSTM 和基本 LSTM 的模型，也分析了其准确性和误差。文献[7]通过实验分析了不同网络切片预测 MAPE 误差参数以及 LSTM 平均误差。文献[16]通过计算预测值与实际值之间的 RMSE 指标说明其预测精度。文献[51]基于 ARMA 的方法对资源需求进行了预测。本章通过对比不同模型的预测结果并计算得到的 RMSE 指标,验证本章提出的 SVR-GRU 组合模型效果。该仿真运行在 PyCharm 环境中，实验相关参数如表 3.4。

表 3.4 不同模型相关参数

模型	参数
ARIMA	p=2, q=1
LSTM	迭代次数=1000, timestep =24, batchSize=100
SVR	C=0.12, gamma=0.031, kernel=rbf
GRU	迭代次数=1000, timestep =24, batchSize=100

如图 3.3 是 SVR-GRU、LSTM 和 MARA 模型对 Cell_001801 单元流量的预测值与实际值对比, 由图 3.3 可以看出, 组合模型和 LSTM 模型预测效果接近, 在整体趋势上与实际值相近, 而 ARIMA 模型预测则有较大误差, 预测效果较差。

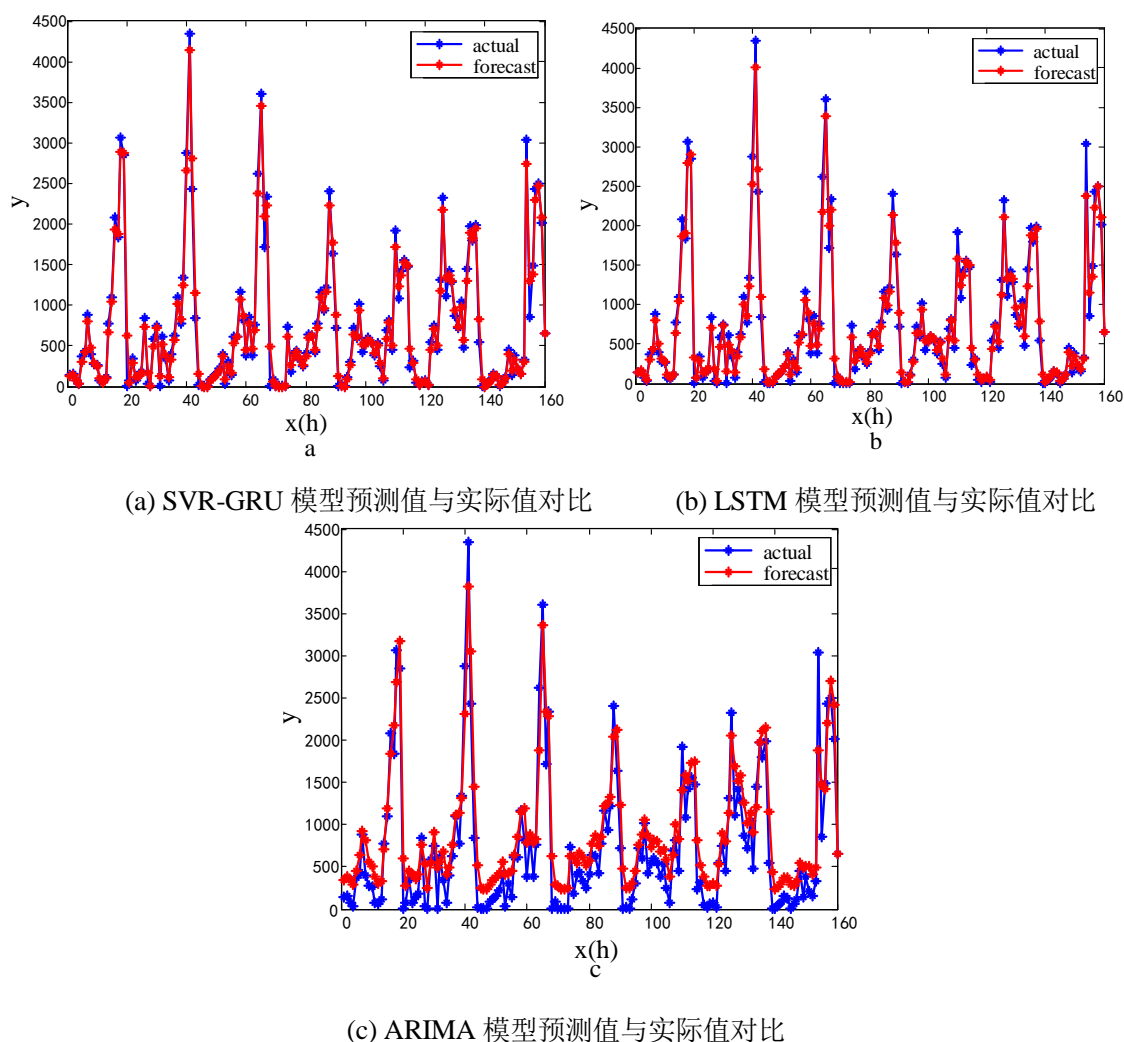


图 3.3 不同模型预测效果

RMSE 指标越小, 代表模型预测效果越好。如表 3.5 是由 10 组数据计算权重、其他测试数据相同的条件下计算得到的 RMSE 指标, 可以看出 SVR-GRU 模型的 RMSE 在三种算法中最小, 也即 SVR-GRU 模型预测的误差最小, 在这三种模型中预测效果最好, 其次是 LSTM, RMSE 比 SVR-GRU 模型要大 15% 左右, 预测效果最差的是 ARIMA 模型, 这是由于边缘网络流量的复杂特点, 线性预测模型难以准确预测, RMSE 指标较大。

表 3.5 不同模型 RMSE 指标对比

指标	RMSE		
	SVR-GRU	LSTM	ARIMA
Cell_001801	62.462	73.638	96.257
Cell_001912	45.723	54.259	70.128
Cell_001963	78.165	89.387	112.792

为探讨 RMSE 值与权重数据的关系，依次增多确定权重的数据组，分别对三个单元进行了实验，实验结果如图 3.4。

由图 3.4 可以看出，在其他条件相同时，SVR-GRU 模型的 RSME 参数随着权重确定数据的增加而一定程度的降低，也即预测效果会有所提升，在数据组数增加到 10 组之后，RMSE 指标会逐渐趋于平缓。实验结果表明，多的数据组确定权重，可以避免单组数据确定权重造成的偶然性，可以降低 RMSE 值，提升预测效果。

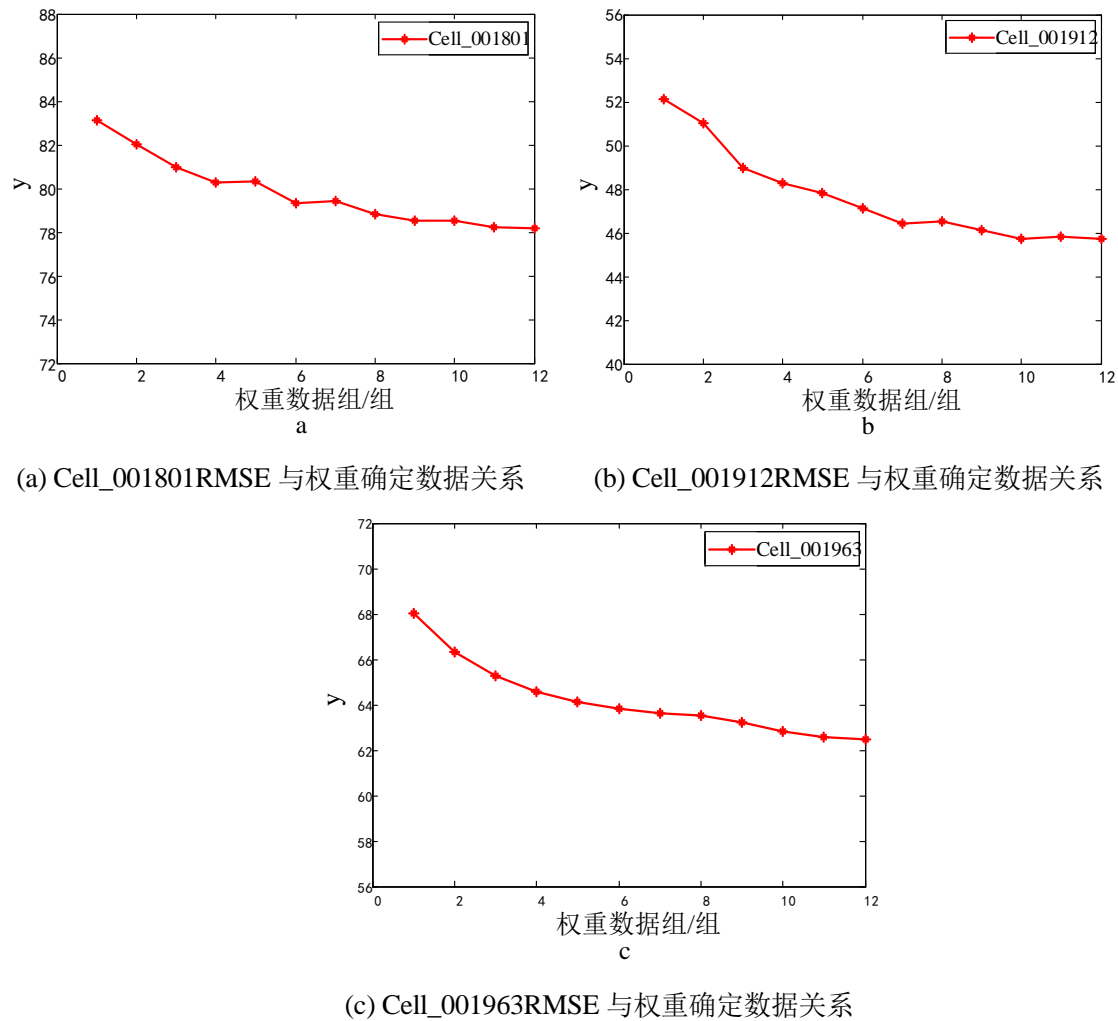


图 3.4 权重确定数据与 RMSE 的关系

3.5 本章小结

本章针对当前 VNF 需求预测方法准确率较低且不适用于边缘网络的问题,提出了一种适用于边缘网络的 VNF 需求预测方法。该方法考虑到边缘网络中网络服务流量的特点,利用相关数据集,选取计算 SVR 模型和具有记忆功能的 GRU 模型,分别提取历史数据的短期和长期特征,并通过方差倒数法为两种模型赋予不同的权重,将两种模型的优点有效结合,以提高预测准确率。通过仿真表明,本章提出的方法在边缘网络场景中较于 LSTM 能够减少预测误差,提升预测效果。

第4章 基于深度强化学习的 VNF 迁移

4.1 系统模型及问题描述

随着边缘计算、物联网等技术的迅速发展,网络应用更新换代速度加剧,新兴的移动多媒体服务已渗透到人们的日常生活中^[54]。而传统网络中网络功能大都基于专有硬件,部署和升级都要对硬件设施进行更新,一定程度上增加了运营商的开销。通过网络功能虚拟化技术能够实现网络功能的软件化,降低成本,并提高网络的灵活性和可扩展性。近年来,边缘计算、物联网等技术的发展,大量智能终端以及网络实时业务涌现^[55],运营商也逐渐尝试在网络边缘部署各种虚拟网络功能,以提高网络服务质量,降低业务时延和成本。

将 SDN 和 NFV 集成以用于将来的网络,有望具有集中化的网络管理、虚拟化的服务功能链、降低的成本和增强的服务质量等特点^[56]。然而,在 NFV 环境的网络中,由于服务功能链请求的到达时间与生存周期的差异性,随着时间的推移,部分物理节点将部署大量虚拟网络功能,其资源占用率达到瓶颈后,服务质量将会急剧下降。特别是在网络边缘,面对突发性强、波动性大的用户请求,单个边缘节点部署的不同虚拟网络功能负载严重很难满足用户的需求时,通过将部分 VNF 迁移或复制到周围具有空闲资源的服务器节点上,能够实现服务器的负载均衡,提升服务质量。

由于大量智能设备、不同实时网络业务的增加,造成网络服务请求的多样性以及网络环境的复杂性,从而导致 VNF 迁移策略的状态和动作空间很大,很难用启发式算法得到最优解。随着用户请求对网络时延的要求越来越高,在服务器负载严重的情况下,通过 VNF 的迁移能够实现服务器的负载平衡,但却难以及时保证用户的服务质量,通过 VNF 的复制,可以满足更多用户的需求,保障网络服务质量,但同时也会造成系统成本的增加。因此,本章结合 VNF 的复制,构建 VNF 迁移模型,在保证网络服务质量的前提下,以较小成本实现服务器的负载均衡。VNF 迁移模型构建过程如图 4.1。

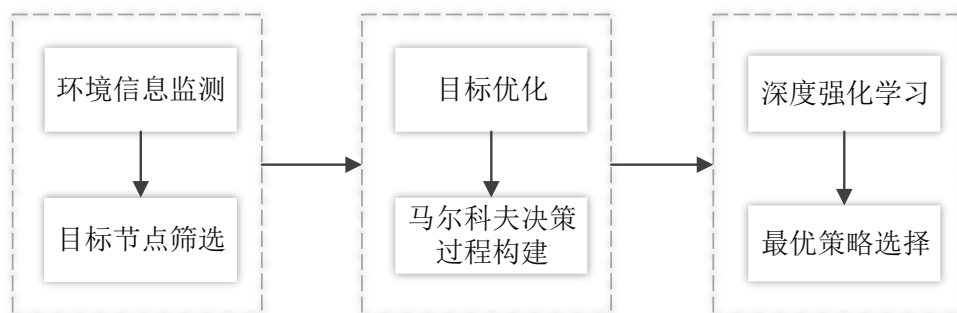


图 4.1 VNF 迁移模型构建过程

具体来说,先是收集当前节点与周围节点的信息,主要包含节点当前空闲资源、负载状态信息、以及短期历史的负载状态。然后,通过收集的信息,进行进一步的分析,获取有用的信息,筛选出迁移的候选目标节点。之后,依据目标优化问题构建马尔科夫决策过程,从而转换为策略选择问题。最后利用深度强化学习获取最优的迁移策略。

4.2 基于空闲资源和负载的阈值分析

在 SDN/NFV 网络拓扑模型中,通过控制器收集并监测边缘各个服务器节点的空闲资源、负载状态等信息,生成一个节点列表 N_{list} ,若某个服务器节点过载,提供网络服务的 VNF 不能够及时满足当前节点覆盖范围下的所有请求时,需要进行迁移或者复制到周围空闲资源的服务器节点上,保证用户的服务体验质量,实现负载均衡。为保证选取的节点能够满足迁移或复制所需的资源且实现负载均衡,本章提出了一种基于空闲资源和负载的三级阈值分析方法,用以选取合适的服务器节点。主要流程为:

(1) 当 VNF 需要迁移或复制副本时,首先预估所需的资源 E_{need} ,并将其作为第一级阈值,依次判断当前服务器节点的周围服务器空闲资源列表 E_{near} 中资源是否满足条件:

$$E_{near}^i \geq E_{need}, 0 \leq i \leq j \quad (4.1)$$

其中 E_{near}^i 表示第 i 个服务器上空闲资源大小, j 表示服务器空闲资源列表中服务器数量。若服务器节点不能满足该条件,则将服务器该节点剔除,剩余节点则继续进行下一轮判断。

(2) 依次计算服务器节点列表中的负载状态,若服务器节点的负载因子大于 L_1 , 则将其移除节点列表,符合条件的节点进入下一轮判断。定义负载因子 η_l 为服务器当前使用的资源比上节点所有资源为:

$$\eta_l = \frac{E_u}{E_{all}} \quad (4.2)$$

其中 E_u 表示服务器中已使用的资源, E_{all} 表示服务器所有的资源。

(3) 利用节点短期内的资源、链路使用等历史数据,通过支持向量回归进行预测未来节点的负载状态,若未来一段时间内的负载因子大于 L_2 , 将该节点移除列表。最终按照空闲资源大小进行排序,取前 N_s 个节点,作为最终的候选节点。基于空闲资源和负载的阈值分析流程如图 4.2。

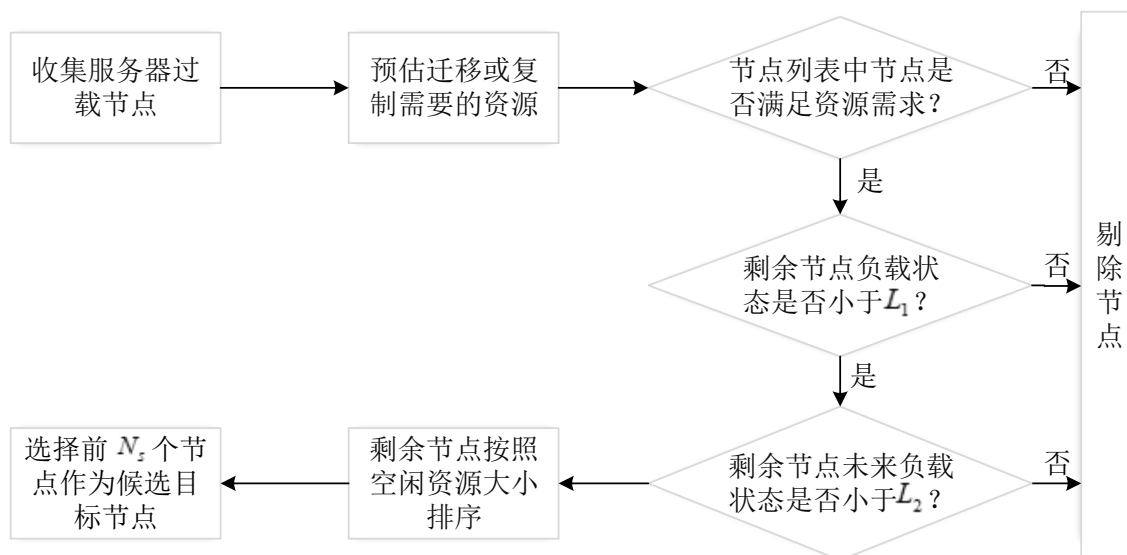


图 4.2 基于空闲资源和负载的阈值分析流程

4.3 目标优化模型

4.3.1 VNF 复制策略

若服务器节点中不同网络服务对应的虚拟网络功能不能够满足当前的业务需求,会导致 SFC 的端到端的时延增大,用户的服务体验质量下降,为保证用户的服务体验质量,需要利用额外的资源进行 VNF 的复制,副本的数量取决于负载的状态。

如图 4.3 为 VNF 复制模型,服务功能链需要依次经过节点 1 上的 VNF1 和节点 2 上的 VNF2 进行处理,在节点 1 负载时,无法保证用户服务体验质量,此时可通

过容器或虚拟机技术在节点 3 上复制 VNF1 的副本,实现负载均衡,保障服务质量。

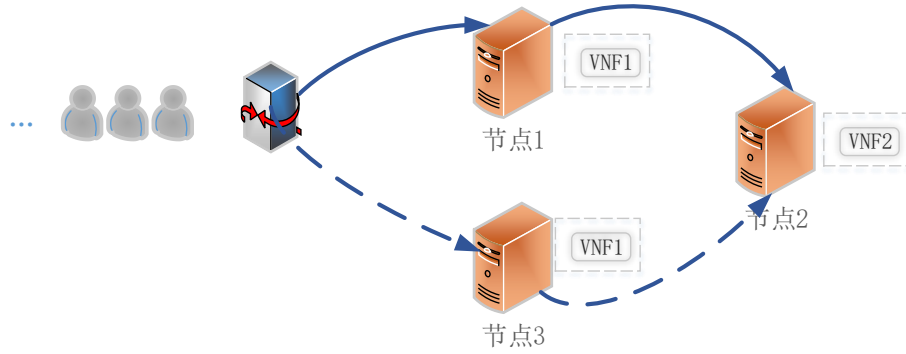


图 4.3 VNF 复制模型

4.3.2 优化目标

网络模型用一个无向图 $G(N^P, L^P)$ 表示, N^P 表示物理服务器节点的集合, L^P 表示物理链路的集合。任意的物理节点 $e \in N^P$, 任意一条物理链路 $l_{i,j} \in L^P$ 。用 $l_{c,d}$ 表示物理节点 c 到节点 d 之间的链路。本章主要考虑物理服务器节点的计算资源和存储资源约束, 用 X_e^{com} 表示服务器节点 e 中计算资源, X_e^{mem} 表示服务器节点 e 中存储资源。用 x_y^{com} 表示 VNF y 所需的计算资源, x_y^{mem} 表示 VNF y 所占存储资源, $B_y^{e1,e2}$ 表示 VNF y 从节点 $e1$ 迁移至节点 $e2$ 消耗的带宽资源。

将服务器节点每一种资源利用率作为服务器负载均衡的主要参考, 节点资源使用情况用为已使用的资源与总资源的比例。当计算资源或存储资源使用情况大于过载阈值时, 将会触发 VNF 的迁移或复制, 计算资源过载阈值用 U_c^x 表示, 存储资源过载阈值用 U_m^x 表示。

对于时延来说, VNF y 实时迁移的停机时间为 VNF 从一个物理节点到另外一个节点导致的服务中断时间^[57], 用 τ_y^{move} 表示。VNF 复制过程造成的时延主要是副本启动的时间, 可以忽略不计。对于成本来说, VNF 迁移成本主要是带宽资源的消耗, 而 VNF 复制成本主要是为副本分配的计算资源和存储资源, 用 y_{copy}^{com} 表示复制 VNF y 所需的计算资源, y_{copy}^{mem} 表示复制 VNF y 所需的存储资源。因此, 执行 VNF y 迁移或复制的成本 C_{all} 可表示为:

$$C_{all}^y = \delta C_{copy}^y + (1 - \delta) C_{move}^y \quad (4.3)$$

其中 δ 为 VNF 不同动作的标志, δ 为 0 表示进行的是 VNF 的迁移, δ 为 1 时表示进行的是 VNF 的复制。 C_{copy}^y 可以表示为:

$$C_{copy}^y = y_{copy}^{com} + y_{copy}^{mem} \quad (4.4)$$

C_{move}^y 为 VNF y 从一个节点迁移到另一个节点的开销, 参考文献[58]的定义, 迁移开销为使不发生流量丢失而被转发到控制器的网络数据包占用的控制器缓冲区大小, 考虑虚拟网络功能 y , 其服务的网络流包括 f_1, f_2, \dots, f_n , 流量速率分别为每秒 b_1, b_2, \dots, b_n , 迁移所耗费时间为 τ_y^{move} , 则迁移开销 C_{move}^y 可以表示为:

$$C_{move}^y = \tau_y^{move} \cdot \sum_{i=1}^n b_i \quad (4.5)$$

服务功能链的端到端时延主要为链中各个 VNF 的处理时延之和^[19], 且服务功能链的处理时延受物理服务器负载影响^[45], 定义服务器节点 e 中的 VNF y 在 t 时刻处理时延为:

$$\tau_y^h(t) = \frac{\tau_y^{pro}(t)}{\mu_y^e(t)} \quad (4.6)$$

其中 $\tau_y^{pro}(t)$ 表示 t 时刻 VNF y 对数据包的处理时延, $\mu_y^e(t)$ 表示为:

$$\mu_y^e(t) = \frac{\beta_y^e(t)}{(u_e^{com}(t))^2} \quad (4.7)$$

于是服务器节点 e 中的 VNF y 在 t 时刻处理时延可进一步表示为:

$$\tau_y^h(t) = \frac{(u_e^{com}(t))^2}{\beta_y^e(t)} \times \tau_y^{pro}(t) \quad (4.8)$$

其中 $\beta_y^e(t)$ 代表 VNF y 在 t 时刻对节点 e 计算资源的占用情况, 即 $\beta_y^e(t) = x_y^{com}(t) / X_e^{com}$, $u_e^{com}(t)$ 表示服务器节点 e 在 t 时刻的计算资源利用率, 若节点 e 中运行的 VNF 数量为 n , 则 $u_e^{com}(t)$ 可以表示为:

$$u_e^{com}(t) = \sum_{y=1}^n \beta_y^e(t) \leq 1 \quad (4.9)$$

由于网络服务质量与 SFC 端到端时延相关, 因此当 SFC 端到端时延能够满足基本需求也即 SFC 不超过最大可容忍延迟时, 才能够保障相应网络业务的基本服务质量, 若 SFC m 包含的 VNF 为 $F_m^S = \{v_1, v_2, \dots, v_y\}$, 则 SFC 的端到端时延 τ_{sfc}^m 为:

$$\tau_{sfc}^m = \sum_{i=1}^y \tau_i^h \quad (4.10)$$

本章主要解决的问题是以最小的迁移或复制成本，实现服务器的负载均衡，并保障网络服务质量。通过主要目标函数法，将对 SFC 端到端时延的优化与服务器资源利用率的优化作为已知条件，从而将多目标优化问题转化为单一的最小化成本优化问题。目标优化问题为：

$$\begin{aligned} & \min_{v \in V} \sum_{y=1}^y C_{all}^y \\ & s.t. \\ & c_1 : B_l^{ij} \leq B_l^{mn} \\ & c_2 : \tau_{sfc}^k + \tau_y^{move} \leq T_{SFC}^k, y \in F_k^S, k=1,2,3\cdots \\ & c_3 : u_e^{com} \leq U_c^X, e=1,2,3\cdots \\ & c_4 : u_e^{mem} \leq U_m^X, e=1,2,3\cdots \end{aligned} \quad (4.11)$$

其中 T_{SFC}^k 表示服务功能链 k 的最大容忍时延， V 表示网络中进行迁移或复制的 VNF 集合。条件 c_1 表示从 i 到 j 的虚拟链路带宽不能大于映射的物理链路带宽；条件 c_2 为表示网络中各条服务功能链端到端时延在 VNF 迁移过程中需要满足相应基本需求，不能超过最大容忍延迟； c_3 表示系统中各个服务器节点的计算资源利用率不能大于过载阈值 U_c^X ， c_4 表示系统中各个服务器节点的存储资源利用率不能大于过载阈值 U_m^X 。

由于资源约束下的虚拟网络功能迁移优化问题是 NP 难问题^[58]，VNF 复制和 VNF 迁移结合的目标优化问题同样在多项式时间内寻求最优解，是 NP 难问题。利用传统的方法求解比较困难，而启发式算法容易陷入局部最优解，而且需要考虑到不同情况时选择复制或迁移的行为，因此，本章依据目标优化问题，构建马尔科夫决策过程，通过深度 Q 网络，利用经验池随机替换数据和随机选取数据样本可以避免陷入局部最优，找到最优的策略。

4.4 马尔科夫决策过程

对网络系统进行进一步分析，系统具有马尔科夫性，即系统的下一个状态只与当前状态有关，而与之之前的状态无关。因此对系统构建马尔科夫决策过程，MDP 是决策的一种数学模型，对于具有马尔科夫性质的系统环境，能够模拟智能体可实现

的随机性策略与奖励，旨在依据当前系统所处的状态以及执行不同策略后环境的反馈，从可用的策略中选择一个最优策略。MDP 可以由四元组 (S, A, P, R) 描述， S 表示为状态空间，用以描述系统可用的状态集合； A 表示系统可以执行的动作空间，在执行某个动作之后，系统会进入其他的状态； P 是系统不同状态之间转移的概率； R 则是奖励函数，如果一组 (s, a) 转移到了下一个状态 s' ，那么奖励函数可记为 $r(s' | s, a)$ 。

1. 定义状态空间

网络中各个服务器节点在正常运行时，将系统在 t 时刻的状态空间 $S(t)$ 定义为：

$$S(t) = \{(U(t), F(t), H(t)) | U(t) \in [0, 1], F(t) \in \{0, 1\}, 0 \leq H(t) \leq h_{\max}\} \quad (4.12)$$

其中 h_{\max} 为系统中服务器节点的数量，是一个恒定值， $F(t)$ 是系统中服务质量好坏的标志， $F(t)$ 为 0 时表示服务质量满足基本需求， $F(t)$ 为 1 时表示服务质量不能够满足基本需求， $H(t)$ 为当前系统中在 t 时刻处于过载的服务器数量， $U(t)$ 为系统全局服务器节点的资源平均利用率，其表达式为：

$$U(t) = \frac{1}{h_{\max}} \cdot \sum_{e=1}^{h_{\max}} u_e(t) \leq 1 \quad (4.13)$$

$u_e(t)$ 为服务器节点 e 在 t 时刻的所有资源利用率，取值范围为 0 到 1，由于 $U(t)$ 的值随时间而连续变化，而在 DQN 算法中不能包含无限的状态和动作，因此需要对 $U(t)$ 进离散化处理。具体地，当 $U(t)$ 利用率离散间隔为 Δu ，则系统全局服务器节点的平均利用率的状态空间被离散成 $N_u = 1/\Delta u$ 个区间，系统中部署的服务器数量 h_{\max} 以及在 t 时刻过载服务器数量 $H(t)$ 本身是离散的，不需要进行离散化处理，因此服务器离散空间 $E = \{e_1, e_2, \dots, e_{h_{\max}}\}$ ，在 t 时刻过载的服务器离散空间 $E_o = \{e_1, e_2, \dots, e_{H(t)}\}$ 。

于是，系统的离散状态空间 S 可以表示为：

$$S = \{s_1, s_2, \dots, s_{N-1}, s_N\} \quad (4.14)$$

状态空间 S 一共拥有 N 个状态，其中 $N = 2 \times (N_u + 1) \times (h_{\max} + 1)$ 。由于系统中过载的服务器数量随时间变化，因此在 t 时刻状态为 $(N^t \cdot \Delta u, F^t, H^t)$ 的系统可以选择的候选状态集合 $\hat{S}(t)$ 为：

$$\hat{S}(t) = \{(N^t \cdot \Delta u, F^t, H^t) | N^t \in [0, N_u], H^t \in [0, h_{\max}]\} \quad (4.15)$$

2. 定义动作空间

动作空间是可执行的一系列动作的集合。由于 VNF 的迁移或复制动作是离散的，不需要进行离散化处理。且迁移或复制的目标节点已经进行相应的筛选，候选目标节点个数为 N_s 个，定义迁移目标节点空间为 D 为：

$$D = \{d_1, d_2, \dots, d_N\} \quad (4.16)$$

若系统中的部署的虚拟网络功能集合为 $V = \{v_1, v_2, \dots, v_{q-1}, v_q\}$ ，每个服务器节点部署的虚拟网络功能集合 V_e 为：

$$V_e = \{v_1^e, v_2^e, \dots, v_j^e\} \quad 0 \leq j \leq q, 0 \leq e \leq h_{\max} \quad (4.17)$$

对于过载服务器上每一个 VNF 都可以选择迁移或复制，且每次执行动作时只对一个 VNF 进行操作，因此 t 时刻的动作空间 $A(t)$ 为：

$$A(t) = \{a(t) \mid a \in q\} \quad (4.18)$$

其中 a 表示 VNF 的动作， q 表示 VNF 可执行的动作空间， q 定义为：

$$q(t) = \{(a_{v,e}^{\delta,d}(t)) \mid \forall v \in V_e, \forall e \in E_o, \forall \delta \in \{0,1\}, \forall d \in D\} \quad (4.19)$$

系统在 t 时刻进行决策时，由 $s(t)$ 到 $s(t+1)$ 的转移过程中，可执行的动作空间 $A_{t,t+1}$ 为所有过载服务器节点中运行的 VNF 可执行动作的集合，用表达式可表示为：

$$A_{t,t+1} = \{(a_{v,e}^{\delta,d}) \mid \forall v \in V_e, \forall e \in E_o, \forall \delta \in \{0,1\}, \forall d \in D\} \quad (4.20)$$

3. 定义状态转移矩阵

服务器节点在选择虚拟网络功能 v 执行动作时，执行动作 a ，并由状态 s_k 转移至下一个状态 s_{k+1} ，由于系统的转移只与当前的状态有关，不与历史其他状态有关，因此，从状态 s_k 转移到状态 s_{k+1} 的概率为：

$$p(s(t) = s_{k+1} \mid s(t-1) = s_k) = p(s(t) = s_{k+1} \mid s(t-1) = s_k, \dots) \quad (4.21)$$

由于初始状态之间的转移概率难以知道，因此采用 ε 贪婪策略，其表达式为：

$$\pi(a \mid s) \leftarrow \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|A(s)|} & a = A^* \\ \frac{\varepsilon}{|A(s)|} & a \neq A^* \end{cases} \quad (4.22)$$

其中 $\pi(a \mid s)$ 为系统环境在 s 状态下执行动作 a 的策略， ε 是一个很小的正数，通过人为确定， $A^* \leftarrow \arg \max_a Q(s, a)$ 表示在某个时刻所有动作中被系统认为最优的动作，

使得值函数 $Q(s, a)$ 最大。 ε 贪婪策略是选取使状态-动作值函数最大的动作的概率最大, 而执行其他动作的概率都一样, 都为 $\frac{\varepsilon}{|A(s)|}$ 。 $|A(s)|$ 表示可选动作空间的数量, 不为 0。由于系统在 t 时刻时可选动作数量为 $\sum_{i=1}^{H(t)} 2 \cdot |V_i| \cdot N_s$, 其中 $|V_i|$ 表示该集合中元素的数量, N_s 为筛选出的目标节点数目, 于是 $|A(s)|$ 的值为:

$$|A(s)| = \sum_{i=1}^{H(t)} 2 \cdot |V_i| \cdot N_s \quad (4.23)$$

4. 定义奖励函数。

在离散的状态空间中, 系统在进行决策时, 依据当前的状态能够确定后续可达状态空间和可执行的动作空间, 若系统在 t 时刻从可执行的动作空间选取一个动作, 执行选取的动作之后, 将会得到一个相应的奖励值。由于本章需要解决的问题是考虑 SFC 端到端时延因素, 以最小的成本实现服务器的负载均衡。因此, 本章设计奖励函数时主要考虑服务器利用率、服务功能链端到端时延与迁移或复制过程中消耗的成本方面的因素。设计的奖励策略为:

(1) 考虑网络服务质量, 网络服务质量由服务功能链端到端时延决定, 当各条服务功能链端到端时延可以满足基本用户需求时, 认为网络服务质量是有保障的。在系统中网络服务质量不能够满足基本需求时, 对其奖励一个足够小的负值 R^- , 令其为 -1。

(2) 在系统网络服务质量可以满足基本需求且系统无过载服务器节点时, 也即系统处于终止状态, 对其给予一个足够大的正值 R^+ , 令其为 1。

(3) 对于系统状态转移过程中耗费的成本主要是迁移或复制过程产生的成本, 对其给予关于成本的负相关函数, 将系统由状态 $s(t)$ 转移至 $s(t+1)$ 的成本耗费奖励函数定义为:

$$f(c_t) = -\xi \cdot \sum_{v=1}^y c_t \quad (4.24)$$

其中 c_t 是由状态 $s(t)$ 转移至 $s(t+1)$ 消耗的资源成本, ξ 为折损系数, 使得 $f(c_t)$ 的绝对值小于 1。

综上, 若 $S(t)$ 表示终止状态集合, $S_o(t)$ 表示 SFC 时延不满足需求的状态集合, 则由状态 $s(t)$ 转移至 $s(t+1)$ 的即时奖励函数为:

$$r(t) = \begin{cases} R^+, & \text{if } s(t) \in S(t) \\ R^-, & \text{if } s(t) \in S_o(t) \quad t=0,1,\dots \\ f(c_t), & \text{otherwise} \end{cases} \quad (4.25)$$

在考虑即时奖励的同时，也需要考虑未来的奖励，系统最终选择策略的目标是在某个策略执行完成之后在各个时刻的累计奖励值最大化，即未来的期望奖励最大。未来的预期奖励可表示为：

$$R(t) = \sum_{i=1}^{\infty} \gamma^i \cdot r(t) = \gamma \cdot r(1) + \gamma^2 \cdot r(2) + \dots + \gamma^i \cdot r(t) \quad (4.26)$$

γ 是折扣因子，范围是 0 到 1，考虑到状态转移过程越多，对未来的奖励将会越小，因此需要对未来的奖励进行一定的折扣。

系统环境在 t 时刻的状态为 $s(t)$ ，依据策略 π 从可执行动作空间中选取一个动作 $a(t)$ ，并通过值函数 $Q_{\pi}(s(t), a(t))$ 对采取的动作进行好坏的评价， $Q_{\pi}(s(t), a(t))$ 可以表示为：

$$\begin{aligned} Q_{\pi}(s(t), a(t)) &= E[R(t) | s_t = s(t), a_t = a(t)] \\ &= E[r(t) + \gamma \cdot r(t+1) + \dots | s_t = s(t), a_t = a(t)] \\ &= E[r(t) + \gamma \cdot Q_{\pi}(s(t+1), a(t+1)) | s_t = s(t), a_t = a(t)] \end{aligned} \quad (4.27)$$

因此，在状态为 $s(t)$ 时，最佳的动作可以表示为：

$$a^* = \arg \max_{a(t)} Q_{\pi}(s(t), a(t)) \quad (4.28)$$

4.5 基于深度 Q 网络的 VNF 迁移

目前大多数对 VNF 迁移的研究，都是考虑 SFC 端到端时延与能耗的优化。而很少研究将 VNF 的复制与 VNF 迁移结合，从而实现服务器的负载均衡与网络服务质量的权衡。本章通过协同 VNF 的复制，提出一种基于 DQN 的 VNF 迁移方法，实现在保障网络服务基本质量的同时，能够以较小的成本平衡服务器的负载，主要通过利用 DQN 算法实现最优策略的选择。具体的基于 DQN 的 VNF 迁移算法架构如图 4.4。

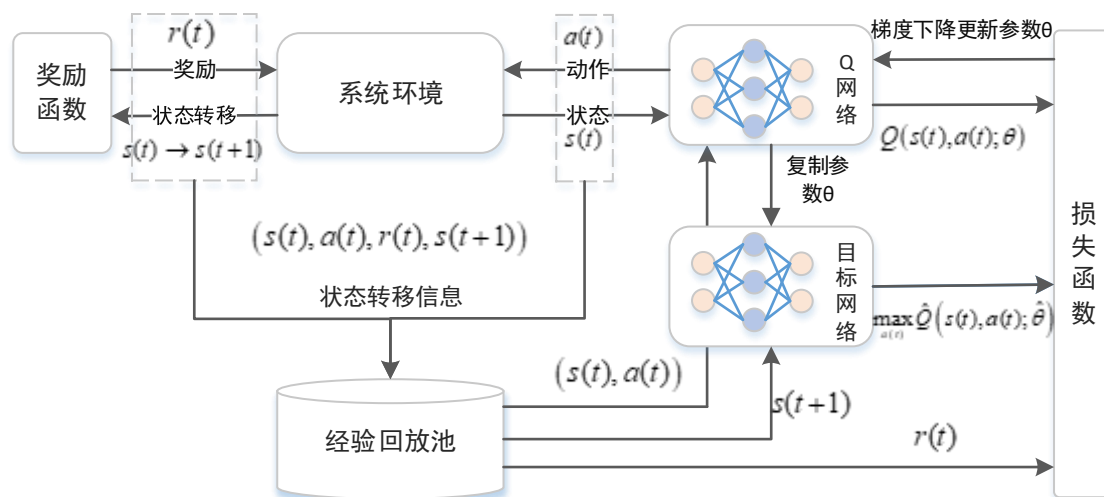


图 4.4 基于 DQN 的 VNF 迁移算法框架

观测到当前系统环境状态 $s(t)$ ，经过当前值网络也即 Q 网络，并根据 ε -贪婪转移策略，选择可执行的动作 $a(t)$ ，系统状态将会由状态 $s(t)$ 转移到状态 $s(t+1)$ ，并依据奖励函数得到相应的奖励 $r(t)$ 。然后将与环境交互得到的状态转移数据组 $(s(t), a(t), r(t), s(t+1))$ 放入到经验回放池中，在需要训练时随机选取一定样本数据来更新权值向量 θ 。

DQN 采用经验回放机制可以在一定程度上消除数据之间的相关性，状态转移样本数据是通过系统与环境之间进行互动而产生的。训练数据具体的生成过程主要为：

(1) 环境的探索。由于系统环境未知性，需要对网络系统环境进行充分探索，处于某个状态的系统进行决策时，通过使用 ε 贪婪 (ε -greedy) 策略来进行动作的选取，即以 $1-\varepsilon$ 的概率从可执行动作空间中选取令 Q 值最大的动作，以 ε 大小的概率选取动作空间中的其他动作。此外，由于 ε 的大小影响对环境状态探索的效果和速度， ε 值较小时，可以更好使用所有的训练数据，但是由于 ε 较小导致执行其他动作的概率较低，降低探索环境的效率，也容易在探索环境的初期过程中陷入局部最优；而 ε 值较大时，可以加快对环境的探索，但是会降低探索过程中选取到更佳动作的可能性。为探索到更多的未知环境信息，在环境探索过程中使 ε 逐渐减少，即在探索环境的初始阶段，使用较大的 ε 值，随着状态的转移也相应地减少 ε 值^[59]，从而在提高探索到最优动作可能性的同时兼顾探索环境的效率。

(2) 存储状态转移信息。探索环境的过程中，在系统选取并执行关于 VNF 的动

作 $a(t)$ 之后, 状态由 $s(t)$ 转换到 $s(t+1)$, 此时系统会得到相应的一个奖励 $r(t)$, 因此在系统环境状态转移的过程中主要涉及到 VNF 的动作信息 $a(t)$ 、系统获取的奖励 $r(t)$ 以及环境状态转移前后的状态 $s(t)$ 和 $s(t+1)$ 四个信息。在每次状态转移时, 都将产生一个关于状态转移的四元组信息 $(s(t), a(t), r(t), s(t+1))$, 该状态转移信息将会存储在经验回放池中, 当经验回放池容量已满, 采用随机替换数据的方式存入经验池中。

具体的基于 DQN 的 VNF 迁移策略训练算法具体过程如表 4.1。

表 4.1 基于 DQN 的 VNF 迁移策略训练算法

算法 2: 基于 DQN 的 VNF 迁移策略训练算法

1. **输入:** 深度学习的学习率 λ , 折扣系数 γ , 经验池容量大小 C , 训练样本数 M , 训练次数 N , 目标网络参数更新频率 K , 状态转移步数 T ;
2. **初始化:** 设置随机参数 θ , 初始化 Q 网络和目标网络和经验池 C ;
3. **输出:** 最佳策略 π ;
4. **for** 状态转移步数 $step = 1$ to T **do**
 5. t 时隙监测当前系统全局服务器利用率 $U(t)$ 、网络服务质量优劣标志 $F(t)$ 和系统环境中服务器过载节点的数量 $H(t)$, 并初始化状态 $s_0 = \{U(t), F(t), H(t)\}$;
 6. **for** $n = 0$ to N **do**
 7. 依据当前状态 s_n 和转移概率, 从可选动作空间中选择动作 a_n , 得到新的状态 s'_n , 并依据奖励函数给予相应的奖励 r_n ;
 8. **if** 经验池有可用容量 **then**
 9. 将状态数据组 (s_n, a_n, r_n, s'_n) 放入经验池中;
 10. **else**
 11. 随机替换经验池中的状态数据组;
 11. 从经验回放池中随机选择 M 个状态转移数据样本集合: $(s_k, a_k, r_k, s'_k), k = 1, 2, \dots, M$, 并计算得到 $y_k = r_k + \gamma \cdot \max_{a'_k} \hat{Q}(s'_k, a'_k; \hat{\theta})$;
 12. 通过梯度下降更新 θ 参数;
 13. **if** $step = K$ **then**
 14. 复制参数 θ 到目标网络, 更新目标网络中的参数;
 15. **end if**
 16. **end if**
 17. **end for**
 18. **end for**

网络在进行训练时，先随机从经验回放池中选择一段状态转移信息数据 (s, a, r, s') ，将系统状态 s 作为 Q 网络的输入，通过深度学习输出全部可执行动作对应的 Q 值，然后依据已选取状态转移数据中的动作 a ，获得相对应的 $Q(s, a; \theta)$ 。之后将系统状态 s' 输入到目标网络中，得到相应的目标值 y ，并依据 $Q(s, a; \theta)$ 和 y 之间的误差进行参数更新。

另外，每经过一定训练次数，就将目标网络的参数进行更新，更新目标网络参数 $\hat{\theta}$ 的方法是复制 Q 网络的参数，即令 $\hat{\theta} = \theta$ ，从而使得目标网络更新参数的同时又可以保持相对稳定，减少 Q 网络与目标网络输出值的关联性。

本章选取全连接前馈神经网络(Feedforward Neural Networks, FNN)模型实现对目标网络和 Q 网络的构建。FNN 主要由隐藏层、输入层以及输出层三部分组成。由于系统在 t 时隙状态 $\{U(t), F(t), H(t)\}$ 包含全局服务器节点的平均利用率、网络服务质量标志与过载的服务器节点数量三个数值，设置 FNN 的输入神经元个数为 3 个。系统在 t 时隙输出的动作数量与运行的 VNF 数量相关，若 N_v 为当前系统运行的 VNF 数量，动作数量最多为 $2 \cdot N_v \cdot N_s$ 个，其中 N_s 为筛选出的目标节点数目，神经网络的输出则是执行 VNF 复制或迁移动作 $a(t)$ 对应的 $Q(s(t), a(t))$ 。设计的人工神经网络结构如图 4.5。

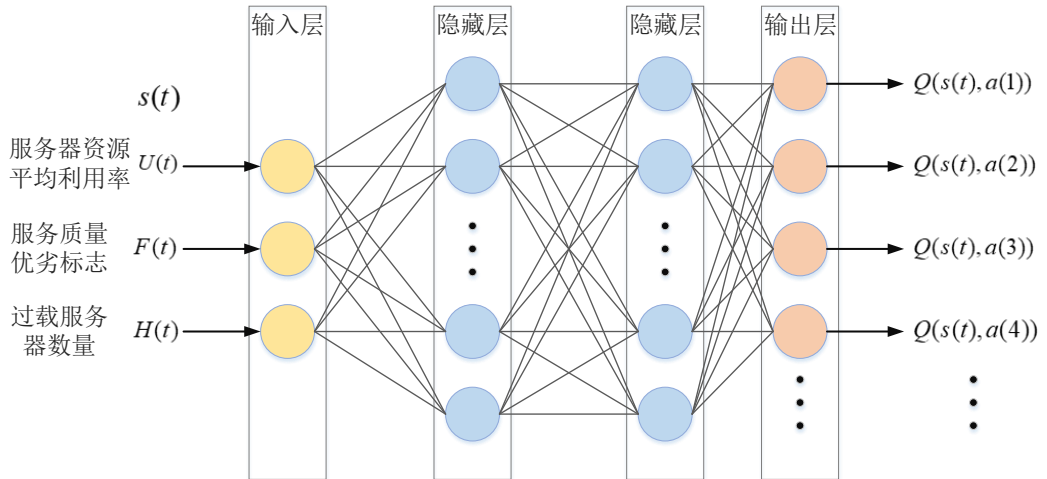


图 4.5 神经网络结构

将神经网络第 i 层的第 j 神经元的输入 y_j^i 定义为：

$$y_j^i = \sum_k w_{jk}^i x_k^{i-1} + b_j^i \mid j \in [1, N_{nodes}^i], k \in [1, N_{nodes}^{i-1}], i \in [2, N_{layers}] \quad (4.29)$$

其中 N_{layer} 为神经网络的层数， N_{nodes}^i 和 N_{nodes}^{i-1} 分别为第 i 层和第 $i-1$ 层的神经元数目，

w_{jk}^i 表示第 $i-1$ 层的第 j 个神经元连接至第 i 层的第 j 神经元的权重, b_j^i 为第 i 层的第 j 神经元的偏置值, x_k^{i-1} 为第 $i-1$ 层的第 k 神经元的输出。若 σ 为激活函数, 则第 i 层的第 j 神经元 x_k^i 可表示为:

$$x_k^i = \sigma(y_j^i) \quad (4.30)$$

FNN 对参数 θ 的优化主要是通过最小化目标网络与 Q 网络之间的损失函数来实现的。损失函数可表示为:

$$L(\theta) = E[(y(t) - Q(s(t), a(t); \theta))^2] \quad (4.31)$$

其中参数 θ 是除输入层之外神经元权重 w 和偏置值 b 的集合, 结合公式(4.29), 可表示为:

$$\theta = [w_{jk}^i, b_j^i] | j \in [1, N_{nodes}^i], k \in [1, N_{nodes}^{i-1}], i \in [2, N_{layers}] \quad (4.32)$$

在参数 θ 更新过程中, 利用梯度下降方法进行更新, 若 λ 为深度学习的学习率, 则更新公式可表示为:

$$\theta = \theta + \lambda [y(t) - Q(s(t), a(t); \theta)] \nabla Q(s(t), a(t); \theta) \quad (4.33)$$

公式(4.31)中 $y(t)$ 为目标值, 通过目标网络获得, 执行 VNF 的动作 $a(t)$ 由 Q 网络得到。由于最优策略通常是利用动作值函数 $Q^\pi(s(t), a(t))$ 得到的, $Q^\pi(s(t), a(t))$ 是评判在当前状态 $s(t)$ 时选取动作 $a(t)$ 的好坏, 可通过贝尔曼方程迭代得到, 表示为:

$$Q^\pi(s(t), a(t)) = r(t) + \gamma \sum_{a(t+1) \in A} P(s(t), a(t), s(t+1)) \cdot Q^\pi(s(t+1), a(t+1)) \quad (4.34)$$

其中 $P(s(t), a(t), s(t+1))$ 为状态 $s(t)$ 转换到 $s(t+1)$ 的概率, A 表示可执行动作空间, 于是最优动作值函数可表示为:

$$Q^*(s(t), a(t)) = \max_{a(t) \in A} \{ r(t) + \gamma \sum_{a(t+1) \in A} P(s(t), a(t), s(t+1)) \cdot Q^*(s(t+1), a(t+1)) \} \quad (4.35)$$

则在系统在状态 $s(t)$ 情况下执行最佳的动作 $a^*(t)$ 可表示为:

$$a^*(t) = \arg \max_{a(t) \in A} Q^*(s(t), a(t)) \quad (4.36)$$

4.6 仿真结果与性能分析

为验证本章提出方法的有效性, 本章将迁移和复制过程中耗费的成本以及 SFC

端到端时延作为对算法评估的标准,验证提出的 VNF 迁移算法性能。本章通过与文献[23]提出的基于线性规划的 VNF 迁移算法和文献[21]提出的遗传算法对比,以更好评估本章提出算法的有效性。本章的仿真平台主要基于 Python3.6,服务器参数主要通过参考文献[45]和[60]进行配置,每台服务器都具有相应的容量,成本以迁移和复制过程中消耗的服务器资源容量进行计算,具体的仿真参数如表 4.2。

表 4.2 网络场景的具体仿真参数

仿真参数	仿真数值
服务器节点数量 h_{\max}	32 个
服务器节点计算资源容量	300 与 350 之间随机分布
服务器节点存储资源容量	300 与 350 之间
数据包大小	5 到 15 之间均匀分布
VNF 的计算资源需求	1 到 20 之间均匀分布
VNF 的存储资源需求	1 到 20 之间均匀分布
每条 SFC 拥有 VNF 的数量	5 到 8 之间均匀分布
SFC 个数	20 到 40 之间依次增加
SFC 可接受的端到端时延 T_{SFC}^i	不大于 150
资源利用率过载阈值	0.75
服务器负载阈值 L_1 和 L_2	0.6, 0.75
折扣因子 γ	0.96
神经网络隐藏层个数	2
神经网络隐藏层神经元数量	512
神经网络的激活函数	RELU
ε -greedy 概率 ε 的初始值和最终值	0.99, 0.1

4.6.1 自身性能分析

针对本章所提的基于 DQN 的 VNF 迁移算法,讨论了本章算法只进行迁移、只进行复制和迁移复制结合三种情况对性能的影响,也讨论了 VNF 迁移和复制次数与 SFC 数量的关系,SFC 数量对各性能的影响如图 4.6。

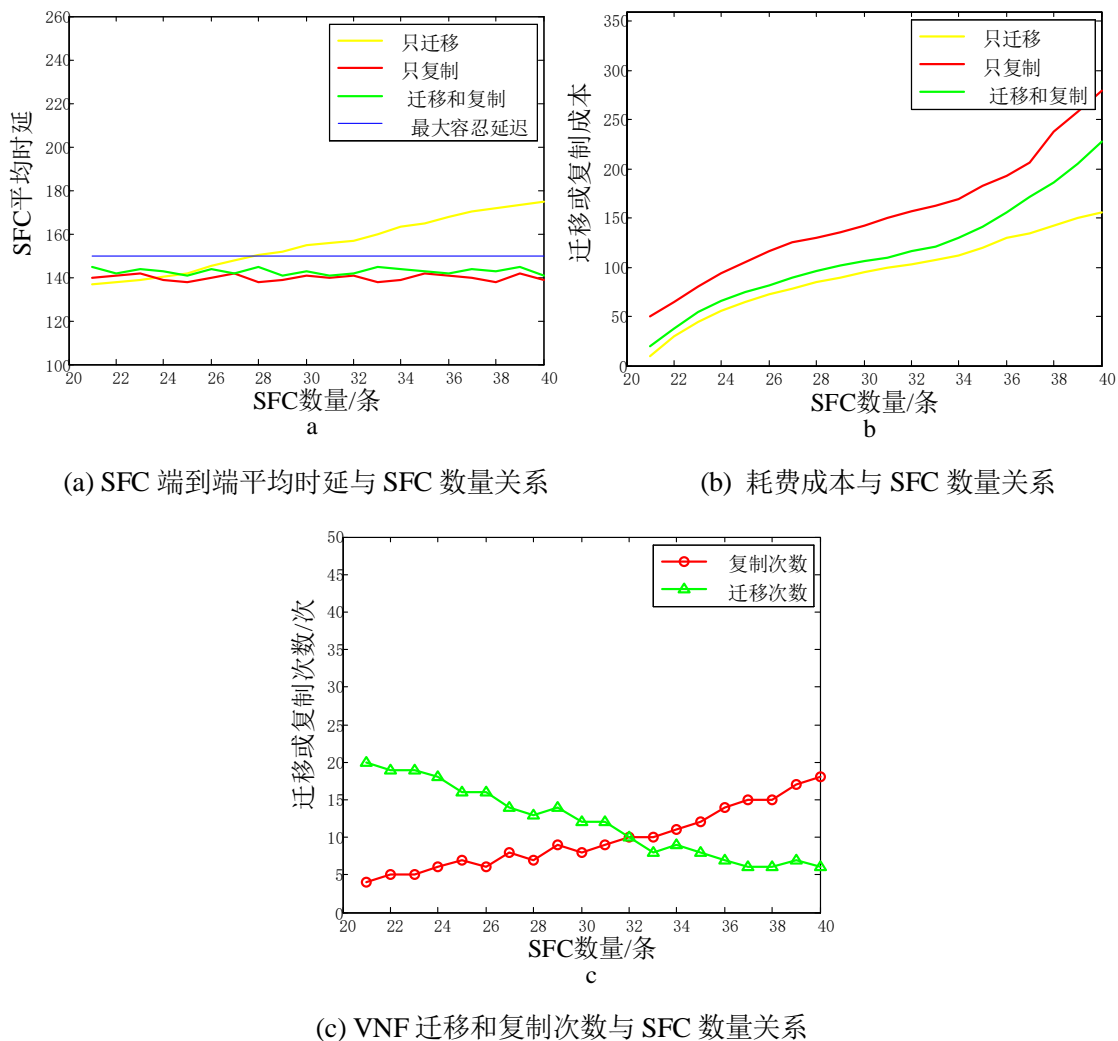


图 4.6 SFC 数量对各性能的影响

由图 4.6(a)(b)(c)可知, 随着 SFC 数量的增加, 只进行 VNF 迁移会导致 SFC 端到端时延逐渐增大并超出最大容忍延迟, 无法保障网络服务质量, 这是由于随着迁移次数的增加, 导致服务中断的时间增大, 从而使 SFC 端到端时延增加, 降低网络服务质量。而迁移与复制结合的算法和只复制算法都可以保障 SFC 端到端时延不会超过最大容忍延迟, 保障一定的网络服务质量, 这是由于这两种方法都结合了 VNF 的复制, 通过利用 VNF 的复制来保障 SFC 的端到端时延。由图 4.6(b)可以看出, 只进行迁移算法的成本是最低的, 而只复制算法的成本最高, 且明显高于其他两种情况, 这是由于 VNF 的复制会带来更多的成本, 而 VNF 迁移与复制结合能够在保障 SFC 端到端时延的同时, 降低一定的成本。通过图 4.6(c)可以看出, 随着 SFC 数量的增加, VNF 复制的次数将会增加, 而迁移的次数将会降低, 当网络中 SFC 数量增加时, 服务器利用率增加, 会导致 VNF 处理时延增加, 此时通过 VNF 的迁移并不

一定能够保障 SFC 端到端时延, 为保障网络服务质量, 将会通过更多的 VNF 复制来保障 SFC 端到端时延。

4.6.2 不同方案间的性能对比

为充分体现本章所提算法的性能, 还对比了其他两种算法即 LP 算法和 GA 算法, 评估三种算法在服务功能链数量变化时的时延和耗费成本的性能表现, 如图 4.7 为具体仿真结果。

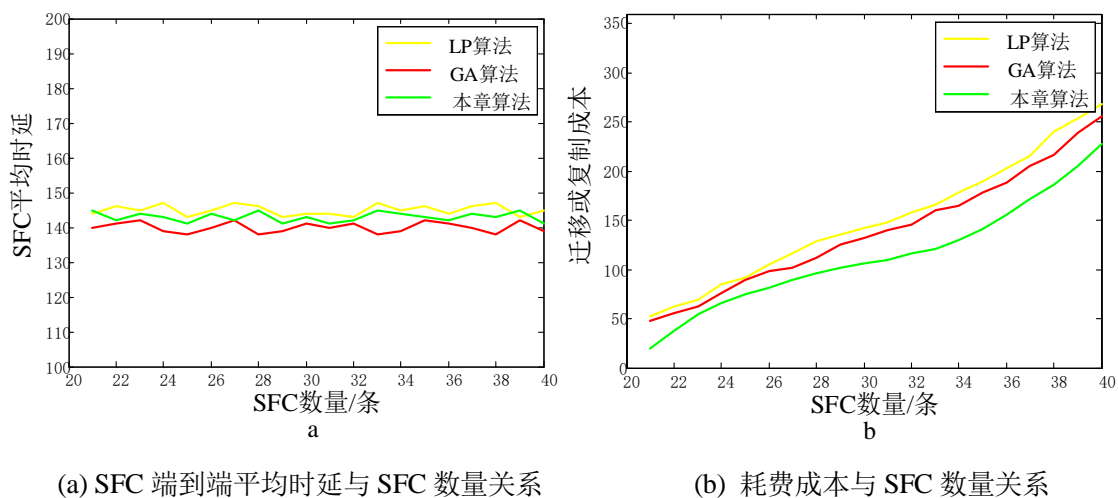


图 4.7 时延和成本与 SFC 数量的关系

从 4.7(a)(b)图中可以知道, 随着网络中服务功能链的增加, 本章提出算法、LP 和 GA 算法的服务功能链平均时延保持相对稳定, 在性能上本章算法与 LP 和 GA 算法相差无几, 都能够保障 SFC 端到端时延满足基本需求, 其原因是随着服务功能链的增加, 本章提出算法、LP 和 GA 算法都协同了 VNF 的复制, 可以在一定程度上保障网络服务质量。而对于耗费成本来说, 本章提出算法在三种算法中耗费最少, 且明显少于其他两种算法, 这是由于本章算法主要通过不同的状态下选择最优的动作, 可以以更小的成本实现网络负载均衡。

综上所述, 相比其他算法, 本章算法通过结合 VNF 的复制, 可以在保障 SFC 端到端时延的同时, 能够以更小的成本实现服务器的负载均衡。

4.7 本章小结

本章为了权衡 VNF 迁移造成的服务降级问题, 通过协同 VNF 的复制, 根据目

标优化问题，构建马尔科夫决策过程，从而使问题的求解转换为策略的选择并进行求解。为了简化问题的求解，本章提出了一种基于空闲资源和负载的阈值机制，筛选出合适的候选目标节点，最后利用深度强化学习得到最优 VNF 迁移策略。通过仿真表明，本章所提的 VNF 迁移方法可以在保障网络服务质量的同时，以更小的成本实现服务器的负载均衡。

第 5 章 结束语

5.1 论文总结

近年来,随着 5G 网络技术的发展,各种智能终端设备、实时网络应用以及虚拟现实、增强现实(Augmented Reality, AR)等低时延应用涌现,传统的网络已经无法承受随之而来的海量智能设备及数据。边缘计算、网络功能虚拟化、物联网等技术的发展,加快了 5G 网络技术的发展,也加速推动着网络向着下一代网络转变,网络的发展也从中心逐渐推向边缘。如今,随着服务提供商将愈来愈多的网络业务下沉至网络边缘,许多学者对于 VNF 的需求预测、放置和迁移等研究也越来越关注。本文针对边缘网络场景,重点对 VNF 的需求预测与迁移展开了研究。

在 VNF 需求预测方法研究中,由于边缘网络流量凸显的复杂特征,本文根据网络边缘流量凸显出的波动性大、突发性强等特点,通过结合不同模型的优点,提取数据的相关特征,利用方差倒数法赋予不同模型不同的权重。最后通过利用预测的 VNF 需要处理的网络流量,依据不同 VNF 的处理能力,进一步分析 VNF 的需求,进而可以提前部署相应的 VNF,保障网络服务质量,也可以避免资源的过多浪费。本文提出的预测方法相比 LSTM 预测模型,一定程度上可以减少误差,在预测准确率上有一定的提升。

在 VNF 迁移方法研究中,由于 VNF 的迁移会造成网络服务质量一定程度的下降,VNF 复制可以保障网络服务质量,但却会带来成本的增加。随着网络规模的扩大,网络边缘服务器节点越来越多,针对 VNF 进行迁移时目标节点过多的问题,提出了一种基于空闲资源和负载阈值的机制,分析各个节点当前的空闲资源、负载状态信息以及未来的负载状态信息,筛选出最为合适的几个节点,作为候选目标节点,可以在一定程度上降低问题的复杂度。之后依据目标优化问题构建马尔科夫决策过程,将目标优化问题的求解转化为策略选择问题。对于策略的选择,本文通过协同 VNF 的复制,将 VNF 的迁移和复制作为可执行动作,提出基于强化学习的解决方法,以实现最优策略的选择,仿真结果表明,所提的算法在保障一定网络服务质量的同时,能够以较小的计算和存储资源成本实现服务器的负载均衡。

5.2 未来展望

本文在边缘计算与物联网的大背景下，利用不同时间序列预测模型实现边缘网络中 VNF 需求的预测，可以降低误差，提高预测准确率。此外，对 VNF 迁移方法展开了研究，通过利用深度强化学习，实现 VNF 迁移与 VNF 复制的权衡，通过仿真验证，本文提出的迁移方法能够在保障网络服务质量的同时，有效降低成本。尽管研究取得了一定成果，但仍然存在一些不足，有待进一步完善和改进：

1. 下一代网络逐渐向更加智能化和虚拟化的方向转变，NFV 是一个宏大的架构，对传统网络部署方式是颠覆性的变化，通过将硬件资源虚拟化，以软件的方式实现网络功能，实现软硬件的解耦。本文利用组合模型对 VNF 需求进行了预测，但运行在云平台上的网络功能如何进行高效的自动化部署将是一个很复杂的问题，需要进一步深入研究。

2. 随着 5G 技术的广泛应用，各种新的应用孕育而生，网络功能虚拟化的应用也将更加广泛，但是虚拟化后网络功能并不能降低电信应用的可靠性要求，传统电信硬件通过特殊设计，可靠性通常较高，而虚拟化采用的通用商用设备可靠性相对降低了，如何通过提升软件可靠性来补偿硬件设备导致的可靠性降低，未来可进一步展开相关研究。

参考文献

- [1] 王侃, 赵楠, 李军怀, 等. 移动边缘计算网络中联合无线多播的服务功能链部署算法[J]. 通信学报, 2020, 41(10): 37-47.
- [2] Xu Zichuan, Liang Weifa, Huang Meitian, et al. Task offloading with network function requirements in a mobile edge-cloud network[J]. IEEE Transactions on Mobile Computing, 2018, 18(11): 2672-2685.
- [3] 施巍松, 孙辉, 曹杰, 等. 边缘计算:万物互联时代新型计算模型[J]. 计算机研究与发展, 2017, 54(05): 907-924.
- [4] 徐冉, 王文东, 龚向阳, 等. 网络功能虚拟化中延时感知的资源调度优化方法[J]. 计算机研究与发展, 2018, 55(04): 738-747.
- [5] Mijumbi R., Hasija S., Davy S., et al. Topology-aware prediction of virtual network function resource requirements[J]. IEEE Transactions on Network & Service Management, 2017, 14(1): 106-120.
- [6] 杜爽, 徐展琦, 马涛, 等. 基于神经网络模型的网络流量预测综述[J]. 无线电通信技术, 2020, 46(2): 216-222.
- [7] 唐伦, 周钰, 杨友超, 等. 5G 网络切片场景中基于预测的虚拟网络功能动态部署算法[J]. 电子与信息学报, 2019, 41(9): 2071-2078.
- [8] Kim H. G., Lee D. Y., Hong J. W., et al. Machine learning-based method for prediction of virtual network function resource demands[C]// IEEE Conference on Network Softwarization (NetSoft). Paris: IEEE Press, 2019: 405-413.
- [9] Zhang Chaoyun, Fiore M., Patras P. Multi-service mobile traffic forecasting via convolutional long short-term memories[C]// IEEE International Symposium on Measurements & Networking (M&N). Catania: IEEE Press, 2019: 1-6.
- [10] Wang Yuqing, Jiang Dingde, Huo Liuwei, et al. A new traffic prediction algorithm to software defined networking[J]. Mobile Networks and Applications, 2019: 1-10.
- [11] Ntalampiras S., Fiore M. Forecasting mobile service demands for anticipatory MEC[C]// IEEE 19th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM). Chania: IEEE Press, 2018: 14-19.
- [12] Alawe I., Ksentini A., Hadjadj-Aoul Y., et al. Improving traffic forecasting for 5G core network scalability: a machine learning approach[J]. IEEE Network, 2018, 32(6): 42-

- 49.
- [13] Xu Ran. Proactive VNF scaling with heterogeneous cloud resources: fusing long short-term memory prediction and cooperative allocation[J]. Mathematical Problems in Engineering, 2020, 2020: 1-10.
- [14] Bilal A., Tarik T., Vajda A., et al. Dynamic cloud resource scheduling in virtualized 5G mobile systems[C]// IEEE Global Communications Conference (GLOBECOM). Washington: IEEE Press, 2016: 1-6.
- [15] Jing Xia, Pang Deming, Cai Zhiping, et al. Reasonably migrating virtual machine in NFV-featured networks[C]// IEEE International Conference on Computer and Information Technology (CIT). Nadi: IEEE Press, 2016: 361-366.
- [16] Tang Lun, He Xiaoyu, Zhao Peipei, et al. Virtual network function migration based on dynamic resource requirements prediction[J]. IEEE Access, 2019, 7: 112348-112362.
- [17] Khai N. T., Baumgartner A., Bauschert T. Optimising virtual network functions migrations: a flexible multi-step approach[C]// IEEE Conference on Network Softwarization (NetSoft). Paris: IEEE Press, 2019: 188-192.
- [18] Jing Xia, Cai Zhiping, Xu Ming. Optimized virtual network functions migration for NFV[C]// IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS). Wuhan: IEEE Press, 2016: 340-346.
- [19] Zhou Xinhao, Yi Bo, Wang Xingwei, et al. Approach for minimising network effect of VNF migration[J]. Iet Communications, 2018, 12(20): 2574-2581.
- [20] Eramo V., Ammar M., Lavacca F. G. Migration energy aware reconfigurations of virtual network function instances in NFV architectures[J]. IEEE Access, 2017, 5: 4927-4938.
- [21] Carpio F., Dhahri S., Jukan A. VNF placement with replication for load balancing in NFV networks[C]// IEEE International Conference on Communications (ICC). Paris: IEEE Press, 2017: 1-6.
- [22] Carpio F., Bziuk W., Jukan A. Replication of virtual network functions: optimizing link utilization and resource costs[C]// 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). Opatija: IEEE Press, 2017: 521-526.
- [23] Carpio F., Jukan A., Pries R. Balancing the migration of virtual network functions with replications in data centers[C]// NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium. Taipei: IEEE Press, 2018: 1-8.

- [24] Yi Bo, Wang Xingwei, Huang Min, et al. Design and implementation of network-aware VNF migration mechanism[J]. IEEE Access, 2020, 8: 44346-44358.
- [25] Afrasiabi S. N., Kianpisheh S., Mouradian C., et al. Application components migration in NFV-based hybrid cloud/fog systems[C]// IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN). Paris: IEEE Press, 2019: 1-6.
- [26] Umrao B. K., Yadav D. K. Algorithms for functionalities of virtual network: a survey[J]. The Journal of Supercomputing, 2021, 2: 1-72.
- [27] Tang Hong, Zhou Danny, Chen Duan. Dynamic network function instance scaling based on traffic forecasting and VNF placement in operator data centers[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(3): 530-543.
- [28] Yi Bo, Wang Xingwei, Li Keqin, et al. A comprehensive survey of network function virtualization[J]. Computer Networks, 2018, 133(14): 212-262.
- [29] Kaur K., Mangat V., Kumar K. Architectural framework, research issues and challenges of network function virtualization[C]// 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). Noida: IEEE Press, 2020: 474-478.
- [30] Pandey S., Hong J. W. K., Yoo J. H. Q-Learning based SFC deployment on edge computing environment[C]// 21st Asia-Pacific Network Operations and Management Symposium (APNOMS). Daegu: IEEE Press, 2020, 220-226.
- [31] Hawilo H., Shami A., Mirahmadi M. NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC)[J]. IEEE Network, 2014, 28(6): 18-26.
- [32] Shih Y., Lin H., Pang A., et al. An NFV-based service framework for IoT applications in edge computing environments[J]. IEEE Transactions on Network and Service Management, 2019, 16(4): 1419-1434.
- [33] Chen Shichao, Li Qijie, Zhou Mengchu, et al. Recent advances in collaborative scheduling of computing tasks in an edge computing paradigm[J]. Sensors, 2021, 21(3): 779.
- [34] 张星洲, 鲁思迪, 施巍松. 边缘智能中的协同计算技术研究[J]. 人工智能, 2019, (5): 55-67.
- [35] 朱晓荣, 王一忠. 虚拟网络功能迁移研究综述[J]. 南京邮电大学学报(自然科学版), 2018, 38(1): 45-53.
- [36] 李思莉, 杨井荣. 基于遗传算法的改进时序预测模型研究[J]. 计算机技术与发

- 展, 2020, 30(11): 84-88.
- [37] Sun Youqiang, Wang Rujing, Sun Bingyu, et al. Prediction about time series based on updated prediction ARMA model[C]// 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). Shenyang: IEEE Press, 2013: 680-684.
- [38] 蔡韬. 基于城市大数据的细粒度空气质量预测与推测模型研究及应用[D]. 南京: 南京邮电大学, 2020.
- [39] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016: 133-140.
- [40] Schmidhuber J., Hochreiter S. Long short-term memory[J]. Neural Comput, 1997, 9(8): 1735-1780.
- [41] 杨丽, 吴雨茜, 王俊丽, 等. 循环神经网络研究综述[J]. 计算机应用, 2018, 38(S2): 1-6+26.
- [42] Chung J., Gulcehre C., Cho K H., et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. Eprint Arxiv, 2014.
- [43] Woesner H., Verbeiren D. SDN and NFV in telecommunication network migration[C]// European Workshop on Software Defined Networks (EWSDN). Bilbao: IEEE Press, 2015: 125-126.
- [44] Cho D., Taheri J., Zomaya A. Y., et al. Real-time virtual network function (VNF) migration toward low network latency in cloud environments[C]// IEEE 10th International Conference on Cloud Computing (CLOUD). Honolulu: IEEE Press, 2017: 798-801.
- [45] 贺兰钦. 基于深度强化学习的虚拟网络功能部署及迁移优化算法研究[D]. 重庆: 重庆邮电大学, 2020.
- [46] Jang B., Kim M., Harerimana G., et al. Q-Learning algorithms: a comprehensive classification and applications[J]. IEEE Access, 2019, 7: 133653-133667.
- [47] Guan Xin, Huang Yang, Dong Chao, et al. User association and power allocation for UAV-assisted networks: a distributed reinforcement learning approach[J]. 中国通信, 2020, 17(12): 110-122.
- [48] Li Taihui, Zhu Xiaorong, Liu Xu. An end-to-end network slicing algorithm based on deep Q-Learning for 5G network[J]. IEEE Access, 2020, 8: 122229-122240.
- [49] 刘大畅, 丁浩, 曾晶. 5G 核心网网络架构及关键技术分析[J]. 中国新通信, 2021, 23(1): 81-82.
- [50] 师圣蔓. 基于机器学习的网络流量预测与应用研究[D]. 北京: 北京邮电大学,

2019.

- [51] An Xiaoguang, Qu Liangliang, Yan Hui. A study based on self-similar network traffic model[C]// Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA). Guiyang: IEEE Press, 2015: 73-76.
- [52] 曾绍华. 支持向量回归机算法理论研究与应[D]. 重庆: 重庆大学, 2006.
- [53] Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30: 5998-6008.
- [54] Kurri V., Raja V., Prakasam P. Cellular traffic prediction on blockchain-based mobile networks using LSTM model in 4G LTE network[J]. Peer-to-Peer Networking and Applications, 2021: 1-18.
- [55] Huang Huawei, Guo Song. Proactive failure recovery for NFV in distributed edge computing[J]. IEEE Communications Magazine, 2019, 57(5): 131-137.
- [56] Li Junling, Shi Weisen, Zhang Ning. Delay-aware VNF scheduling: a reinforcement learning approach with variable action set[J]. IEEE Transactions on Cognitive Communications and Networking, 2021, 7(1): 304-318.
- [57] Ghaznavi M., Khan A., Shahriar N., et al. Elastic virtual network function placement[C]// IEEE 4th International Conference on Cloud Networking (CloudNet). Niagara Falls: IEEE Press, 2015: 255-260.
- [58] 夏竞. 基于网络功能虚拟化的网络资源优化与安全关键技术研究[D]. 长沙: 国防科技大学, 2018.
- [59] Zhang Miao, Zhang Qi, Lv Yisheng, et al. An AI based high-speed railway automatic train operation system analysis and design[C]// International Conference on Intelligent Rail Transportation (ICIRT). Singapore: IEEE Press, 2018: 1-5.
- [60] 阳勇, 孟相如, 康巧燕, 等. 拓扑与资源感知的虚拟网络功能迁移方法[J]. 计算机科学与探索: 2021: 1-11.

致谢

岁月不居，时节如流，研究生的三年时光转瞬即逝，三年的时间虽然不是很长，但却给我的人生留下了浓墨重彩的一笔。这一笔包含了几度春秋，也折射出了研究生生活的点点滴滴。雨水划过窗台，我似乎听到了导师对我们的谆谆教诲，也听到了大家共聚时的欢声笑语，往日那一幅幅画面也逐渐浮现在我的脑海中，清晰而又深刻。

很荣幸我自己能够成为重庆邮电大学的一名研究生，我非常感谢学校给予我的一切，也是学校才让我认识到了一群可爱又值得感恩的人。在我心中，我特别感谢我的导师黄宏程老师，黄老师是我非常尊敬和敬佩的一位老师，当初是黄老师的一通电话，我才有机会加入这个和睦又温暖的大家庭。不管是在工作还是生活中，黄老师都给予了我们莫大的帮助，给我们创造好的学习环境，关心我们的吃住行……总而言之，黄老师在工作中拼搏奋进，一直悉心指导着我们的学业，给我们授道解惑，带领我们共同进步，一步步壮大我们的大家庭，提升整个团队的实力。

在研究生的三年生涯中，我还特别感谢胡敏老师和寇兰老师。胡敏老师在生活中为人随和，在工作上非常认真负责，在学术上给予了我非常多的帮助，在每次的学术会议中，总是能够很耐心地指出我在学术研究上的错误和给出许多相关的宝贵建议，让我少走了许多弯路。寇兰老师和蔼可亲，很关怀我们大家庭中的每一个人，无论是学习中还是生活中，都给予了我们很多关心，让我们感受到大家庭的温暖。在这三年中，我由衷感谢三位老师给予我的指导和关心，祝愿三位老师身体健康，工作顺利！

此外，感谢王旭鹏、王俊岭、李熠、向薇、王婷婷、董梦媛等师兄师姐在生活和学习中对我的帮助。感谢陈玉明、李净、梁挺、廖强、田亚楠、周秀东和朱润笋同届好友三年的陪伴以及对我的关心，在这段令人记忆深刻的时间，我们一起钻研、互帮互助，共同进步，愿我们之间深厚的友谊长存。感谢我的师弟师妹们在我需要帮助的时候可以给予最大的帮助。感谢三位室友在日常生活中对我的关心和包容，也感谢遇到的每一个人。

感谢我的家人，是他们的支持和关切，我才能够一步一步前行，不断成长。

最后，感谢每一位参加论文评审的专家和老师，感谢你们在百忙之中抽出宝贵的时间审阅本论文！

攻读硕士学位期间从事的科研工作及取得的成果

参与科研项目：

- [1] “面向绿色生态城市的一体化综合管理服务系统与应用——面向绿色生态城市管理的多源泛在感知与多尺度物联融合组网技术研究”， 国家重点研发计划项目， 2019-12.

完成论文及受理专利：

- [1] 黄宏程, 鲍晓萌, 胡敏. 边缘网络中一种虚拟网络功能需求预测方法[J]. 电讯技术: 2021.
- [2] 黄宏程, 鲍晓萌, 胡敏, 等. 边缘网络中一种虚拟网络功能需求预测方法[P]. 中国专利: 202110279221.7, 2021.
- [3] 黄宏程, 鲍晓萌, 胡敏, 等. 一种基于边缘网络的虚拟网络功能迁移方法[P]. 中国专利: 202110285692.9, 2021.

获奖：

- [1] 重庆邮电大学研究生三等学业奖学金 (2018-2019)
- [2] 重庆邮电大学研究生三等学业奖学金 (2019-2020)
- [3] 第十届蓝桥杯重庆赛区 Java 程序设计大赛 A 组二等奖 (2019)
- [4] 重庆邮电大学研究生三等学业奖学金 (2020-2021)