

同濟大學

TONGJI UNIVERSITY

毕业设计（论文）

课题名称 面向复杂工程产品的大规模图网络优化技术

副标题 以汽车风阻系数预测为例

学院 软件学院

专业 软件工程

学生姓名 吴渭

学号 1951574

指导教师 饶卫雄

日期 2023 年 5 月 24 日

面向复杂工程产品的大规模图网络优化技术 ——以汽车风阻系数预测为例

摘要

针对工程产品的优化设计在现代工业领域中扮演着重要的角色。由于传统的 CAE 有限元分析技术难以应对日益复杂的工程问题，因此结合深度学习的工程产品仿真已经成为工业界的研究热点之一。

本文以汽车风阻系数的预测为例，探索优化汽车工程产品设计的深度学习技术。本文提出了一种基于图神经网络的汽车风阻系数预测方法，并将三维汽车建模转化为包含大量节点和边的图网络。该方法结合了图卷积神经网络方法和节点级采样等优化算法，通过对图网络中的节点和边进行采样优化，得到更加准确和高效的汽车风阻系数预测模型。同时，为了验证该方法的有效性，实验采用了实际的汽车外形数据进行训练和测试。结果表明，该方法在预测汽车风阻系数方面具有很高的准确度和鲁棒性，并且能够有效地处理复杂的汽车建模问题。该研究成果为工程设计领域提供了一种新颖的、高效的优化方法，对于提高汽车设计的准确性和效率具有一定的参考意义。

关键词：工程产品，汽车风阻系数，图神经网络，采样算法

Large-scale graph network optimization technique for complex engineering products ——a case study of predicting automobile drag coefficient

ABSTRACT

Optimized design of engineering products plays an important role in modern industrial fields. Due to the increasingly complex engineering problems, traditional CAE finite element analysis technology is insufficient, and thus combining deep learning with engineering product simulation has become a research hotspot in the industry.

This paper explores the application of deep learning technology to optimize the design of automotive engineering products, using the prediction of automobile drag coefficient as an example. A graph neural network-based method for predicting the automobile drag coefficient is proposed, which converts the three-dimensional automobile modeling into a graph network containing a large number of nodes and edges. The method combines graph convolutional neural network methods and node-level sampling optimization algorithms, optimizes the nodes and edges in the graph network, and obtains more accurate and efficient prediction results for automobile drag coefficient. To verify the effectiveness of the method, actual automotive flow field data were used for training and testing. The results show that the method has high accuracy and robustness in predicting automobile drag coefficient and can effectively handle complex automotive modeling problems. This research provides a new and efficient optimization method for the field of engineering design, and has a certain reference significance for improving the accuracy and efficiency of automotive design.

Key words: engineered products, Graph neural network, Sampling algorithm, Automotive drag coefficient

目 录

1	引言.....	1
1.1	课题背景与研究意义.....	1
1.2	国内外研究现状.....	2
1.2.1	三维数据研究表示.....	2
1.2.2	基于图神经网络的大规模图优化技术.....	7
1.3	本文所做的工作与创新点.....	9
1.4	本文组织架构.....	10
2	理论基础.....	11
2.1	三角形网格及 Delaunay 算法.....	11
2.1.1	三维多边形网格基础知识.....	11
2.1.2	基于 Delaunay 三角剖分的网格生成算法.....	12
2.2	神经网络与多层感知机.....	13
2.3	图与图网络.....	14
2.3.1	图.....	14
2.3.2	图网络与图卷积.....	14
3	数据、算法与模型.....	16
3.1	汽车模型网格数据.....	16
3.1.1	构建汽车网格数据集.....	16
3.1.2	数据集整理与预处理.....	17
3.2	模型设计.....	18
3.2.1	MeshGraphNets.....	18
3.2.2	节点采样算法 GraphSAGE.....	20
3.3	模型训练.....	21
3.3.1	噪声注入.....	21
3.3.2	数据标准化.....	21
3.3.3	损失函数与优化算法.....	22
3.4	本章小结.....	23
4	实验与结果分析.....	24
4.1	实验环境.....	24
4.2	实验设计.....	24
4.2.1	基本实验设计.....	24
4.2.2	泛化实验设计.....	25
4.2.3	对比实验设计.....	25
4.3	模型性能评估指标.....	27
4.4	实验结果与分析.....	28
4.4.1	基础实验.....	28
4.4.2	泛化实验.....	31
4.4.4	对比实验.....	32
4.5	本章小结.....	34
5	总结与展望.....	35
5.1	全文总结.....	35
5.2	未来展望.....	35
	参考文献.....	36
	谢辞.....	38

1 引言

1.1 课题背景与研究意义

在国务院印发的《中国制造 2025》通知中明确强调“推进信息化与工业化深度融合”，要“加快推动新一代信息技术与制造技术融合发展，把智能制造作为两化深度融合的主攻方向；着力发展智能装备和智能产品，推进生产过程智能化，培育新型生产方式，全面提升企业研发、生产、管理和服务的智能化水平。”^[1]

计算机辅助设计（Computer-Aided Design, 简称 CAD）^[2]和计算机辅助工程（Computer-Aided Engineering, 简称 CAE）^[3]是现代汽车设计和制造中至关重要的技术手段，它们在整个汽车设计和制造的各个阶段都起着至关重要的作用。在传统的 CAE 技术中，有限元法（Finite Element Method, FEM）^[4]是一种广泛应用于工程领域的数值计算方法，图 1.1 展示了常用的有限元方法分析过程。在汽车设计中，有限元法可以用于模拟各种不同的工况，例如汽车的静态和动态力学特性、汽车结构的耐撞性和安全性能等。有限元法可以将复杂的汽车结构划分成许多小的单元，通过求解这些单元的物理行为来预测整个系统的行为。这种方法的优点是在不破坏实际汽车的情况下进行各种测试，并可以通过更改模型的参数进行多种场景的仿真。

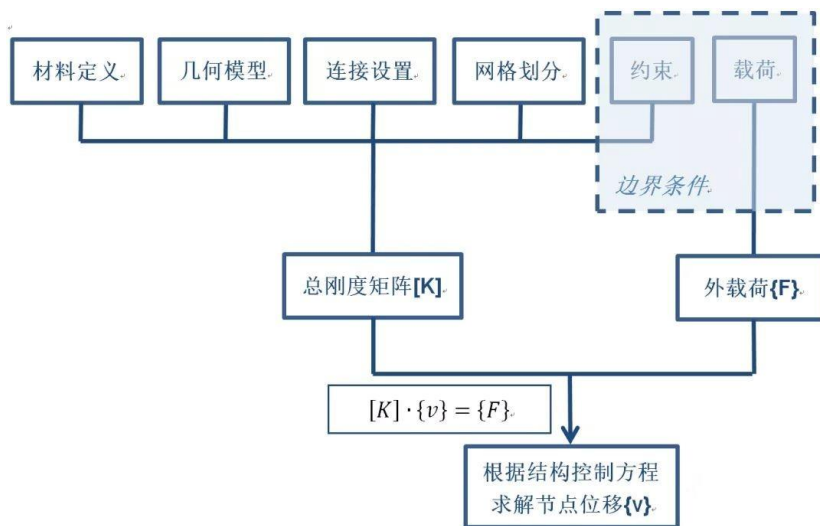


图 1.1 有限元分析过程

FEM 在汽车设计中的应用可以追溯到上世纪 60 年代，当时主要用于分析汽车车身的刚度和强度。随着计算机技术的不断发展，FEM 逐渐成为了汽车设计和制造中必不可少的工具之一^[5]。在汽车的设计和开发过程中，FEM 可以用于模拟和优化多种方案，从而加速产品研发的速度，减少实验和试验的成本，如图 1.2 所示。例如，在汽车碰撞模拟中，可以使用 FEM 预测汽车在碰撞中的行为，并通过模拟不同的碰撞情况来评估车辆的安全性能。在汽车车身轻量化方面，可以使

用 FEM 来分析材料的强度和刚度，优化结构设计，减少重量，提高车身刚度和强度^[6]。

然而，FEM 也存在一些不足之处。由于 FEM 是基于网格划分的方法，因此在处理复杂几何形状和流动问题时可能会遇到困难。此外，在处理非线性问题、非稳态问题和多物理场问题时，FEM 的计算效率可能较低。因此，研究人员开始探索结合深度学习的方法来解决这些问题。例如，在汽车风阻系数预测方面，传统的 FEM 方法需要将车辆的三维建模转换为网格模型，这需要耗费大量时间和计算资源。而基于深度学习的方法可以直接对车辆的三维模型进行处理，无需进行网格划分和离散化处理，大大提高了计算效率。因此，结合深度学习的工程产品仿真^[7]已经成为工业界的研究热点之一。

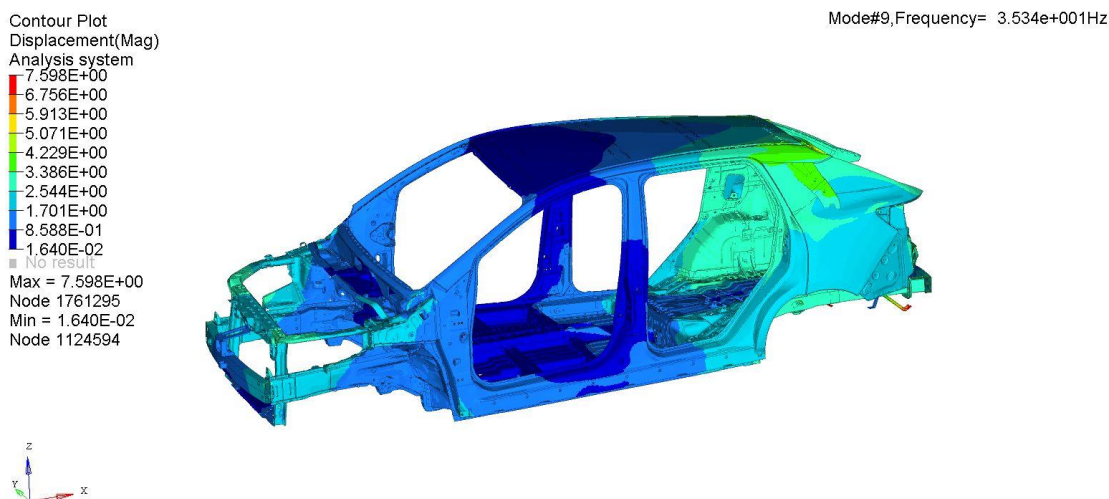


图 1.2 汽车车身的有限元仿真

然而，在实际工程应用中，深度学习技术也面临着一些挑战。首先，工程问题通常涉及到大量的数据和复杂的模型，需要考虑数据的规模和处理能力的问题；其次，工程产品的性能分析和优化需要考虑多种因素的综合作用，如结构、材料、流体力学等，需要深度学习模型能够处理这些多变因素的交互影响；最后，工程应用中需要对深度学习模型进行实时的优化和调整，以保证模型的准确度和效率。

1.2 国内外研究现状

1.2.1 三维数据研究表示

面向深度学习的三维数据表示是三维计算机视觉领域的重要研究方向，也是将深度学习应用至工程设计领域需要解决的首要问题。其目的是将三维场景表示为计算机可以理解和处理的形式，以便于进行深度学习任务的训练和推理。常见的三维数据类型包括体素、点云、多视图和三角网格等等。针对不同的应用场景和工业领域，选择合适的三维数据类型往往能够显著提高算法的性能和效率。

A. 体素数据

体素是一种基于三维网格的体积表示方法，通常用于表示实体物体或场景，如图 1.3 所示。体素表示可以通过光栅化方法将三维空间分割成一系列小的立方体，每个立方体称为一个体素。体素表示的优点是能够表示物体的几何和拓扑信息，并且适用于体积数据处理任务，如医学图像分析。Cicek 等提出的 3D U-Net^[8]模型使用卷积神经网络处理体素数据，在三维医学图像分割任务中取得了良好效果。3D U-Net 是基于 U-Net 的改进，它使用 3D 卷积和上采样来构建一个完整的 3D 图像分割模型，从而在处理三维数据时实现了较好的效果。该模型使用多个分辨率的特征图进行残差连接，同时还使用了深度监督方法来提高模型的稳定性和精度。此外，3D U-Net 模型采用了数据增强和随机扰动等方法来扩充训练数据，从而减少了标注数据的需求，使得模型能够更好地适应医学影像数据的特点。但由于采用了大量的卷积和上采样操作，模型的计算复杂度较高，需要较长的训练时间和更强的计算资源，且对于不平衡的数据分布容易出现偏差，需要额外的调整来解决这个问题。

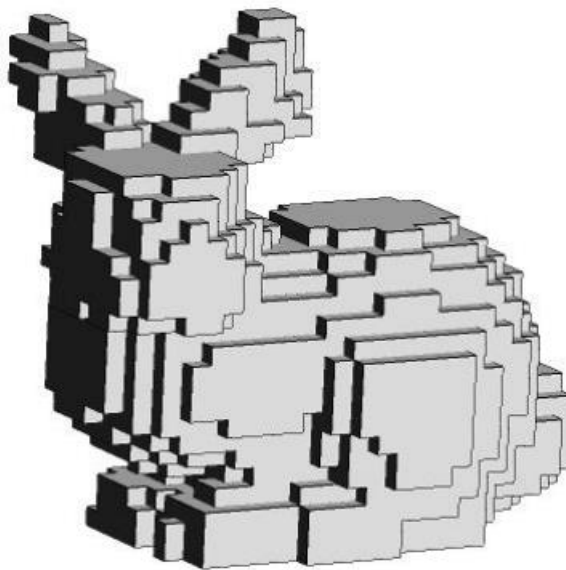


图 1.3 体素类型数据示例

B. 点云数据

点云是一种基于点集表示方法，通常用于表示散点型数据或非实体物体，如图 1.4 所示。点云表示可以通过点云采样或激光雷达扫描得到。点云表示的优点是能够保留物体的表面几何信息，并且适用于点云数据处理任务，如点云分割和配准。在深度学习中，点云数据通常使用基于图(Graph)的卷积神经网络处理。例如，Qi 等提出的 PointNet^[10]是一种处理点云数据的神经网络模型，它能够点云数据作为输入进行分类或分割任务。PointNet 使用一个基于对称函数的多层感知器（MLP）对每个点的特征进行编码，并且使用池化函数将全局特征表示提取出来，最后将这些全局特征表示映射到分类或分割结果。PointNet 的优点是可以直接处理点云数据，不需要额外的特征提取步骤，且具有很好的准确性和鲁棒性。它也可以处理不同形状和大小的点云数据，具

有很好的可扩展性。然而，PointNet 的运行时间较长，并且需要大量的计算资源。此外，它们对于较小的点云数据可能会产生过拟合的问题。因此，在实际应用中，需要仔细考虑模型的复杂性和计算资源的需求，以及对数据预处理和数据增强的需求。

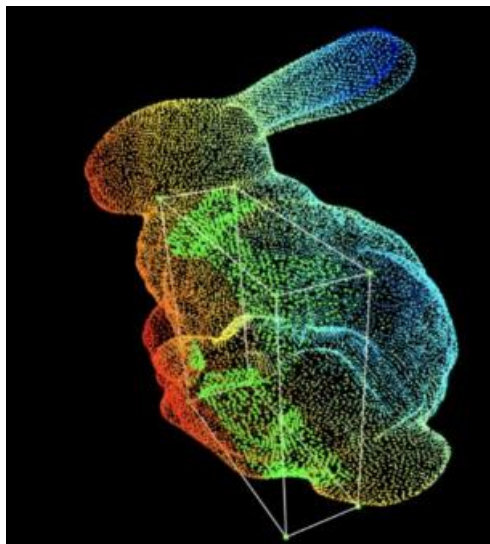


图 1.4 点云类型数据示例

C. 多视图数据

多视图是一种基于多个视角的表示方法，通常用于表示实体物体。多视图表示可以通过在不同视角下拍摄物体图像得到，并将这些图像组合成一个视角序列。多视图表示的优点是能够表示物体的几何和表面纹理信息，并且适用于多视角几何建模和物体识别任务。多视图数据通常使用几何神经网络(Geometric Neural Networks)进行处理。在三维形状分类和分割任务中，Wu 等使用了一种称为 3D-GAN (3D Generative Adversarial Networks)^[11]的方法，它能够从多视图图像中生成三维形状，并且使用了基于图的卷积神经网络对这些三维形状进行分类或分割。3D-GAN 模型结合了生成对抗网络和卷积神经网络，能够从多视图图像中学习到三维形状的特征表示。但由于生成对抗网络的不稳定性，3D-GAN 的训练过程可能会出现不稳定性问题。

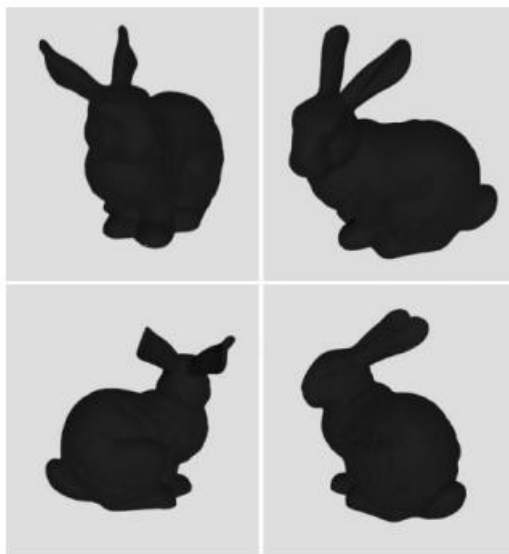


图 1.5 多视图类型数据示例

D. 三角网格数据

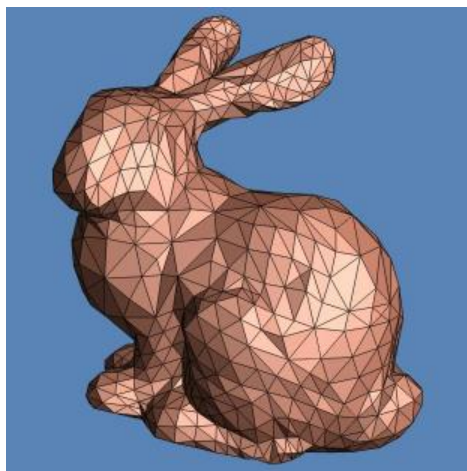


图 1.6 网格类型数据示例

三角网格是一种基于三角形的表示方法，通常用于表示实体物体的表面。三角网格表示可以通过将物体表面离散化成一系列三角形得到。三角网格表示的优点是能够表示物体的表面拓扑结构和纹理信息，并且适用于物体形状分析和流体力学分析。三角网格数据通常使用卷积神经网络或者图卷积神经网络进行处理。例如，在三维形状分类和分割任务中，Rana Hanocka 等提出的 MeshCNN[12]就是一种处理三角网格数据的神经网络模型。它使用了卷积神经网络对三角网格的邻域进行卷积，将其转换为特征表示，然后使用全连接层或池化层进行分类或分割。MeshCNN 模型采用了多层卷积和池化操作，从而逐步提取出越来越抽象的特征。在每一层中，MeshCNN 通过使用可变形卷积和局部池化操作，来处理三角形网格数据中的不规则性。这些操作使得

MeshCNN 能够对三维网格数据进行更加准确的特征提取，并且提高了网络的鲁棒性。相比于传统的基于体素或点云的方法，MeshCNN 更加灵活，可以直接操作三角形网格数据，从而减少了预处理的复杂度。

2021 年，Tobias Pfaff 和 Meire Fortunato 等人在 Alvaro Sanchez-Gonzalez 的研究工作的基础上，设计了基于网格的图网络仿真框架，提出了 MeshGraphNets 图网络模型^[13]。该模型可以训练在一个网格图上传递消息，并在正向模拟过程中适应网格划分。它可以准确地预测广泛的物理系统的动力学，包括空气动力学、结构力学和布料。该模型的自适应性支持学习分辨率无关的动态场景，并可以在测试时扩展到更复杂的状态空间，能够在复杂的环境下进行学习和预测，对多个物理场的模拟具有灵活性，且对于噪声和不确定性具有较好的鲁棒性，如图 1.7 所示。MeshGraphNets 的自适应性支持学习与分辨率无关的动力学，并且可以在测试时扩展到更复杂的状态空间。同时该方法也非常高效，运行速度比训练它的模拟方法快 1-2 个数量级，拓宽了神经网络模拟器可以操作的问题范围，并有望提高复杂科学建模任务的效率。

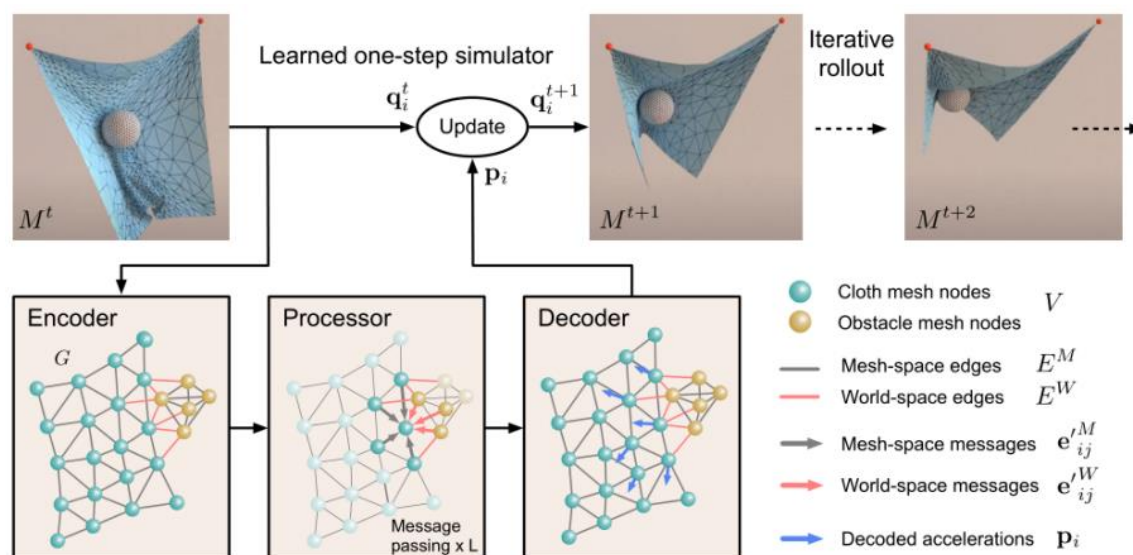


图 1.7 MeshGraphNets 模型原理图

国内外基于深度学习方法表示三维数据的研究现状总结如表 1.1 所示。不难发现，现存的工作大多以体素和点云等数据表示为基础。这是因为这些数据类型能够通过扫描仪等设备轻易地在真实世界中获取得到，而是因为这些规则数据能够自然地应用于卷积神经网络中进行训练。相比之下，高精度地网格数据往往需要通过 CAD 软件人工绘制，且网格地复杂性与不规则形也使其难以运用于卷积神经网络中。因此，为了充分利用网格表征三维形状的天然优势，本文沿着 MeshGraphNets 解决网格类型数据的复杂性和不规则性问题的思路，将网格类型数据自然地转换为图数据类型，应用于复杂工程产品的特征提取任务中，并以汽车模型的风阻预测为例，验证本课题模型的有效性。

表 1.1 三维数据仿真方法汇总

数据类型	模型名称	应用场景	存在问题
体素	3D U-Net	医学图像分割、物体识别	模型训练和推理速度较慢，需要更多的硬件资源
点云	PointNet	三维物体识别、语义分割	容易忽略点云的局部信息
多视图	3D-GAN	三维模型生成	对于不同类型的数据集的泛化性较差
三角网格	MeshCNN	计算机图形学和计算机视觉领域	对于拓扑结构不同的网格数据表现不佳、分割精度差
	MeshGraphNets	流体及空气动力学仿真	对于大规模的网格数据可能存在训练和推理速度较慢的问题

1.2.2 基于图神经网络的大规模图优化技术

图神经网络（Graph Neural Networks, GNN）是一种基于图数据的神经网络技术。GNN 将图中的节点和边视作输入信号，对其进行特征提取和表示学习，以便进行下游任务，如节点分类、图分类、链路预测等。相较于传统的深度学习方法，GNN 模型在处理图数据时能够保留图的拓扑结构和节点间的关系信息，从而更适用于处理图结构数据。Thomas 等人提出的 GCN（Graph Convolutional Network）^[14]将卷积的思想融入图神经网络，通过对图拉普拉斯矩阵的特征分解得到图傅里叶变换的基函数，继而将空域中不规则的图表示转换为谱域中规则的特征向量的线性组合，以此实现对图神经网络的卷积操作，从而进行后续的特征提取与聚合任务。然而，传统的图神经网络在进行数据训练与预测时，依赖于全局的图邻接矩阵和特征矩阵，并以此进行图的特征分解。由于图的特征分解的时间复杂度近似于 $O(n^3)$ ，在图节点数上千万的大型图中将耗费大量的计算时间和存储资源。为此，大量的工作聚焦于大型图的局部采样算法，用局部的特征采样来代替全局的特征表征，以期优化大型图训练任务的时间和空间复杂度。近年提出的采样算法按照采样层次，大致可分为节点采样、分层采样与子图采样三类。

A. 节点采样算法

节点采样算法往往从局部节点出发，按一定的采样策略，向邻域节点进行扩展，在降低图神经网络训练复杂度的同时，保留了图的局部结构信息。常见的节点采样算法有两种：Metropolis-Hastings 采样算法和随机游走采样算法。Metropolis-Hastings 采样算法是一种基于马尔可夫链蒙特卡罗（MCMC）的采样方法^[15]，其思想是通过不断迭代，从一个节点移动到另一个节点，最终得到节点采样结果。随机游走采样算法则是一种基于随机游走的采样方法，其思想是从一个起始

节点开始，沿着边随机游走，最终得到节点采样。

Hamilton 等人提出的 GraphSAGE[17]算法在图卷积神经网络的基础上，采用了随机游走的采样优化算法。GraphSAGE 在每一轮迭代周期中，不断从当前节点出发，向周边的邻域节点扩展，并聚合形成邻域的特征表示。GraphSAGE 算法能够较为准确地提取图中的局部特征信息，且因为无需对全局的拉普拉斯矩阵进行特征分解，该算法能够有效地推广到无监督学习中，并对未见节点进行训练，适用于图结构复杂且容易变化的应用场景，如社交网络。但逐层扩展的采样策略意味着采样节点数的指数级增长，在迭代次数较深时容易引起指数爆炸，进而影响训练效率。为此，VR-GCN (Chen et al, 2018d)[17]提出了一种基于控制变量的估计器。通过维护历轮迭代周期中采样节点的历史嵌入，来保持接受野的小范围，同时减小节点抽样的方差。然而，该算法要求存储历史隐藏嵌入，将消耗大量的额外内存，这在大型图的训练中是得不偿失的。

B. 分层采样算法

分层采样算法将节点划分为不同的层级，每一层级采样一部分节点作为下一层级的输入，并利用采样得到的节点信息进行模型训练。Chen 等提出的 FastGCN[18]错误!未找到引用源。基于图卷积神经网络和分层采样的思想。FastGCN 通过节点的度数对其进行分层，在每层采样固定数量的节点，由此避免了领域扩展的问题。此外，FastGCN 采用了对归一化邻接矩阵权重的重要性采样，对权重分布进行修正，使得每个邻接矩阵元素被采样的概率与其权重成正比，从而有效减小采样方差，提高模型性能。但是，由于 FastGCN 独立地对每一层进行采样，它无法捕获层之间的相关性，从而导致性能下降。为了更好地捕获层间相关性，ASGCN (Huang 等人, 2018)^[19]提出了一种自适应分层采样策略。其中，下层的采样概率取决于上层的采样概率。ASGCN 为了获得更好的层间相关性，仅从被采样节点的邻居中对节点进行抽样，而 FastGCN 利用所有节点之间的重要性抽样。

C. 子图采样算法

子图采样算法的基本思想是从大图中抽样一个子图，将其输入神经网络中处理，以有效降低数据规模。Cluster-GCN (Chiang et al, 2019)[20]首次提出基于高效图聚类算法提取小图聚类。Cluster-GCN 在子图层面构建小批量训练，每次迭代时仅从某子图的内部采样节点，从而避免了邻域扩展的问题。此外，作者提出了随机多分区方案，在迭代过程中随机地合并子图，以期捕获子图间的相关性。然而，Cluster-GCN 可能会改变数据集的原始分布，并引入一些偏差，且采样质量高度依赖于聚类算法的性能。为此 Zeng 提出了 GraphSAINT^[21]，该模型不使用聚类算法，而是引入了三个子图采样器结构来形成子图，分别是节点采样器、边缘采样器和随机游走采样器。此外，该算法还提出了损失归一化和聚集归一化来消除采样偏差。该模型的采样过程如图 1.8 所示

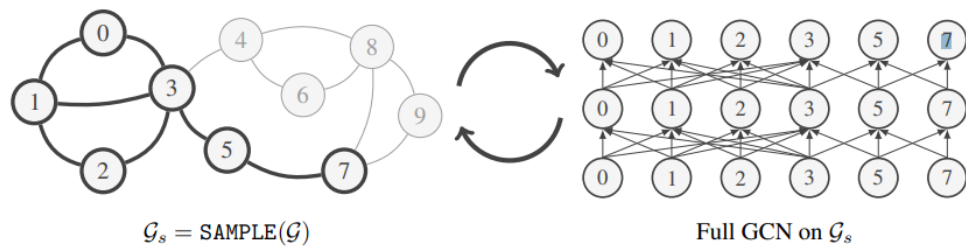


图 1.8 GraphSAINT 的采样过程

表 1.2 汇总了目前国内外主流的图网络采样方法。

表 1.2 图网络采样方法汇总

采样类型	模型名称	主要方法路线	存在问题
节点采样	Metropolis-Hastings	蒙特卡罗方法	收敛速度慢、高维问题困难
	随机游走	基于马尔可夫链的图算法	随机游走路径难以确定、对于大规模图的处理效率低
	GraphSAGE	随机采样与特征聚合	邻域扩展
	VR-GCN	变分自编码器与潜在变量学习	存储历史隐藏嵌入消耗大量内存
分层采样	FastGCN	节点分层与重要性采样	难以捕获层间的相关性
	ASGCN	基于概率的适应分层采样	注意力权重过于依赖初始化
子图采样	Cluster-GCN	子图聚类算法与随机多分区	改变了原始分布，高度依赖聚类算法
	GraphSAINT	子图采样器	难以处理孤立点

1.3 本文所做的工作与创新点

针对汽车风阻系数的仿真预测问题，本文基于现有的 MeshGraphNets 模型，研究如何运用图神经网络处理汽车网格数据，得到具有较好的可扩展性与稳定性的汽车风阻系数回归预测图网络模型。此外，本文应用 GraphSAGE 等采样算法，进一步提升图网络模型面对大规模图数据时的训练效率与可扩展性。本文的研究内容主要包括以下三个方面：

A. 汽车网格数据生成与风阻系数预测

结合前文所提国内外研究，传统的卷积神经网络针对几何体无法有效表示不同分辨率区域，且难以处理高维不规则的图结构数据；本文基于图网络模型 MeshGraphNets，提出了能够处理高精度汽车网格数据的汽车风阻系数回归预测模型

B. 基于采样的大型图优化算法

基于 MeshGraphNets 算法，针对大规模图数据集，提出了结合 GraphSAGE 算法的采样优化技术。通过加入随机采样和特征聚合，提高了原有图网络模型的可扩展性与训练效率。

C. 对不同方法策略进行对比实验

本文将数据分为训练集、验证集与测试集对模型的预测能力进行训练，并使用 MSE、MAE 等指标数据进行评估。在基本实验的基础上，使用不同的采样算法优化模型，对比其优化能力与训练精度，并在测试集上评估模型的泛化能力。

本文的主要研究思路和技术路线如图所示。

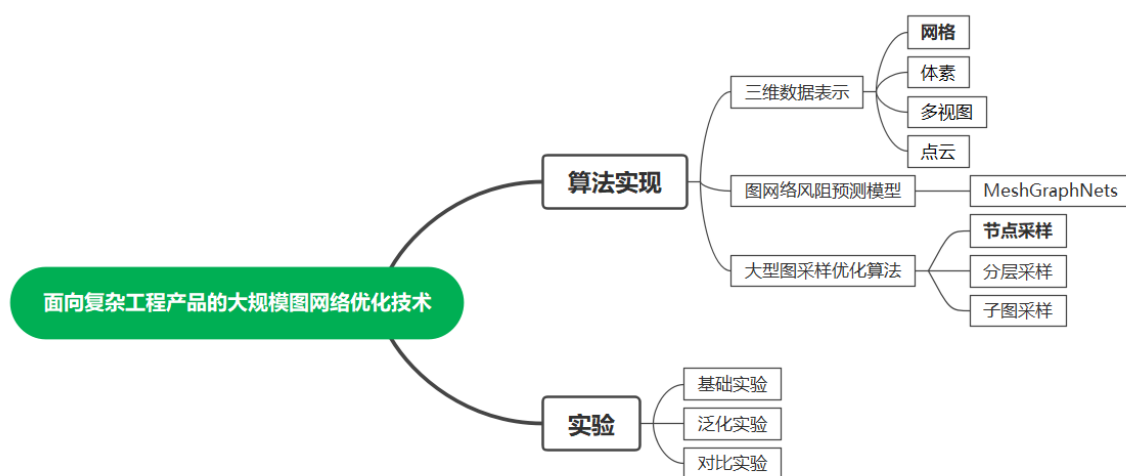


图 1.9 本文研究思路

1.4 本文组织架构

第 1 章：简述汽车风阻系数模拟仿真的意义，并分析了传统有限元方法效率的不足。介绍了国内外关于深度学习结合工程产品仿真研究的现状，分析现有大型图采样算法的可行性与优缺点。确定本文的研究内容与技术路线。

第 2 章：介绍了网格数据生成的工作原理。介绍了神经网络与图网络的结构与工作原理、模型结构与原理，这些理论知识为后续的实验提供理论支持。

第 3 章：基于网格生成算法处理汽车模型数据，划分三角网格并构建图网络结构。在预处理的数据集基础上提出基于图网络结构的风阻系数预测模型，以便进行接下来的实验。对于实验的超参数进行设置调试，进行噪声注入、标准化处理等提高模型鲁棒性的方法，对于模型进行优化。

第 4 章：针对提出的汽车风阻系数预测模型与节点采样算法，设计并实施基本实验，泛化能力，以及与其他采样方法的对比实验。根据实验结果分析分析模型的性能与算法优劣，以及相关超参数对于实验结果的影响。

第 5 章：总结本文的主要内容与结论，对后续研究进行展望。

2 理论基础

本章将介绍本课题研究过程中所涉及相关概念的基础背景知识与理论基础，主要介绍三维多边形网格的背景知识与定义（第 2.1 节）、神经网络与多层感知机（第 2.2 节），以及图与图网络（第 2.3 节）概念。本部分只针对课题中有所使用的相关基础内容进行介绍，并不涉及详细证明与展开。

2.1 三角形网格及 Delaunay 算法

2.1.1 三维多边形网格基础知识

三维多边形网格是指一个由多个三角形、四边形或其他多边形组成的三维网格。它是计算机图形学和计算机辅助设计中常用的数学模型。在三维多边形网格中，每个三角形或四边形被称为一个面，而每个点被称为一个顶点。三维多边形网格可以用于描述三维对象的形状，例如建筑物、汽车、飞机等等。它可以被表示为一个由顶点和面组成的数据结构，其中每个面由它的顶点索引列表表示^[22]。

三维多边形网格具有以下特点：

（1）它由顶点、面和边组成，其中顶点是一个三维空间中的点，面是由三个或四个顶点组成的平面几何体，边是两个相邻面之间的共享边界。

（2）它可以用于描述非常复杂的形状，并且可以通过分割面和添加新的顶点来实现更细的几何细节。

（3）它可以被用于进行三维渲染、物理模拟、建模和可视化等应用领域。

网格数据可以用多种方式表示，即以不同的方法存储顶点、边和面数据。其中顶点-顶点网格（Vertex-vertex Meshes）可以存储顶点信息和每个顶点周围相邻的顶点信息，常用于离散化表示的曲线和曲面；翼边网格（Winged-edge Meshes）可以存储顶点、边和面信息，其中每条边记录了它的两个相邻面的信息常用于多边形网格表示；面-顶点网格（Face-vertex Meshes）可以存储顶点信息和每个面对应的顶点索引信息，常用于三角形网格表示。

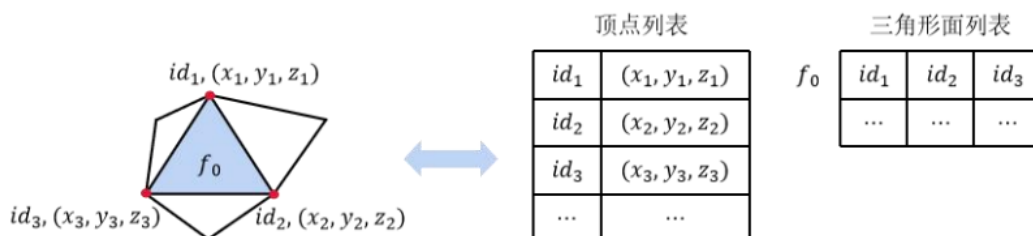


图 2.1 面-顶点三角形网格示意图

如图 2.1，展示了一种常用的面-顶点网格表示方式，我们可以将一个三角形网格表示为 $T = (P, F)$ ，其中：

P 为顶点集合，其中每个顶点 p 都包含其空间坐标 (x, y, z) ，并且每个顶点拥有独特的索引 i ，构成集合 ID ，对于 ID 中的每个 i ，可以通过一个映射 L 查询到对应的顶点 p ，即：

$$P = \{(x, y, z) | x, y, z \in \mathbb{R}\}, \forall p \in P, \exists id \in ID \text{ 使得 } p = L(id) \quad (2.1)$$

F 代表面的集合，其中每个面 f 包含其所表示的三角形的顶点 id ，即：

$$F = \{\{id_1, id_2, id_3\} | id_1 \neq id_2 \neq id_3, id_1, id_2, id_3 \in ID\} \quad (2.2)$$

若两个三角形面 $f1, f2$ 共享一条边，即 $|f1 \cap f2| = 2$ ，则称三角形面 $f1$ 和 $f2$ 是相邻的。

2.1.2 基于 Delaunay 三角剖分的网格生成算法

Delaunay 三角剖分算法是一种将点集转换为三角形网格的方法，该网格满足一些特定的几何约束，称为 Delaunay 约束^[23]。这个约束确保了所形成的三角形的角度都足够接近于 60 度，并且避免了出现奇异形状的三角形。

Delaunay 三角剖分算法的主要思路是基于点集来构建一个三角形网格，在这个三角形网格中每个三角形都满足 Delaunay 约束条件。具体来说，考虑平面点集 $P = \{p_1, p_1, p_2, \dots, p_n\}$ ，满足：

- a) 所有三角形的端点恰好构成集合 P 。
- b) 任意两个三角形的边不相交（要么重合，要么没有交点）。
- c) 所有三角形的合集构成 P 的凸包（convex hull）。

基于 Delaunay 三角剖分的网格生成过程如下：

- a) 计算区域的包围盒，生成两个超三角形作为初始三角网格。根据边界曲线生成边界点。
- b) 将边界点逐一插入到三角网格中。每一步插入后，用 Bowyer-Watson 算法得到新的 Delaunay 三角网格。
- c) 将第一步插入的辅助顶点删除（同时删除与其连接的三角形），得到关于全部边界点的 Delaunay 三角网格（亦即边界点的 Delaunay 三角剖分）。
- d) 在区域内部的三角形，根据一定的条件，将重心（centroid）插入三角网格，并根据 Bowyer-Watson 算法调整，得到新的 Delaunay 三角剖分。直到不再有需要插入的重心点，此时得到区域的完整 Delaunay 三角网格。

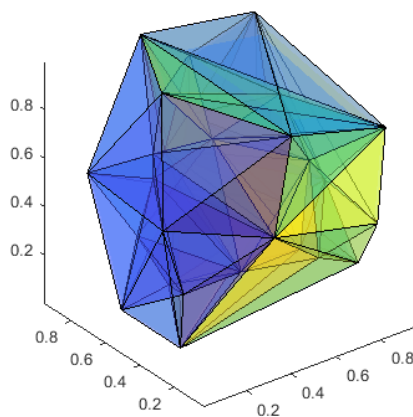


图 2.2 基于 Delaunay 算法的三角网格剖分示例图

2.2 神经网络与多层感知机

多层感知机（Multilayer Perceptron, MLP）是一种基于前馈神经网络（Feedforward Neural Network, FNN）的模型^[25]，由多个全连接的神经网络层组成。每一层包含多个神经元（或称节点），层与层之间的神经元之间全连接，每个连接都有一个权重。输入信号在网络中前向传播，每个神经元都会对输入信号进行一次非线性变换，并将输出信号传递给下一层的神经元，直至到达输出层，最后得到模型的输出。图 2.3 展示了一个典型的多层感知机模型。

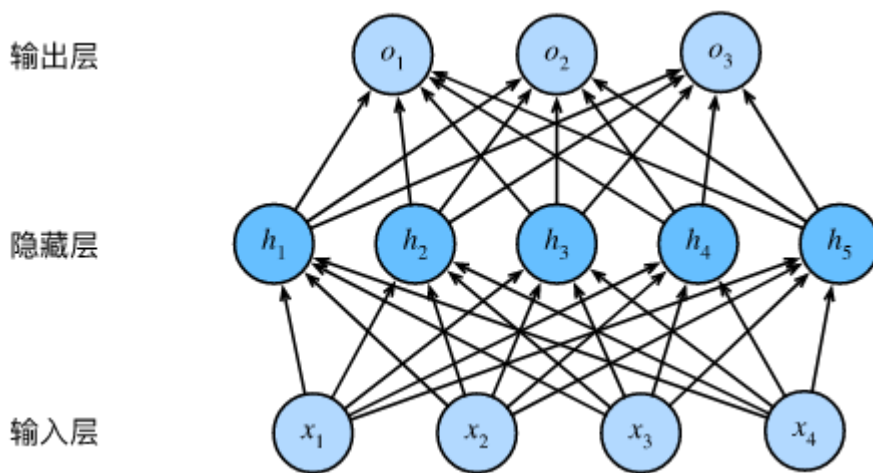


图 2.3 多层感知机模型示例图

MLP 的训练过程通常采用反向传播算法（Backpropagation, BP）来优化模型参数（即权重），使得模型能够对输入数据进行有效的拟合和预测。BP 算法根据训练数据和模型的预测结果来计算误差，并根据误差反向调整每个神经元的权重，以降低模型的误差。这个过程不断迭代，直至模型的误差降到一个可接受的程度。

在反向传播中，假设神经网络的输出为 y ，目标值为 t ，损失函数为 $L(y, t)$ ，其中 y 和 t 都是向量， L 是一个标量。反向传播的目标是计算 L 对神经网络参数 w 的偏导数 $\frac{\partial L}{\partial w}$ ，即损失函数对于每个参数的梯度。我们假设神经网络共有 n 层，第 i 层的输出为 $y^{(i)}$ ，其中 $y^{(1)}$ 是神经网络的输入。假设第 i 层的参数为 $w^{(i)}$ ，则神经网络的输出可以表示为：

$$y = f(n) \left(w(n), f(n-1) \left(w(n-1), \dots, f(1) \left(w(1), x \right) \right) \right) \quad (2.3)$$

其中 x 是输入数据， $f^{(i)}$ 表示第 i 层的计算过程。反向传播算法通过链式法则计算 L 对每个参数的偏导数，具体地：

$$\frac{\partial L}{\partial w^{(i)}} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial w^{(i)}} = \frac{\partial L}{\partial y(n)} \frac{\partial y(n)}{\partial y(n-1)} \cdots \frac{\partial y(i+1)}{\partial y(i)} \frac{\partial y(i)}{\partial w^{(i)}} \quad (2.4)$$

其中， $\frac{\partial L}{\partial y}$ 表示 L 对输出 y 的偏导数， $\frac{\partial y}{\partial w^{(i)}}$ 表示输出 y 对参数 $w^{(i)}$ 的偏导数。具体地， $\frac{\partial L}{\partial y(n)}$ 可以根据损失函数的定义计算， $\frac{\partial y(n)}{\partial y(n-1)}$ 可以根据 $f^{(n)}$ 的求导规则计算，而 $\frac{\partial y(i+1)}{\partial y(i)}$ 则可以根据 $f^{(i+1)}$ 的求导规则计算。

2.3 图与图网络

2.3.1 图

图（Graph）是一种数学结构，用来描述事物之间的关系。它由一组节点（Node）和一组边（Edge）组成，边连接两个节点，表示这两个节点之间存在一种关系^[26]。图可以用来表示各种实际问题，例如社交网络、道路网络、电力网络等等。图中的节点可以表示各种实体，例如人、城市、设备等等，而边可以表示它们之间的联系，例如人与人之间的关系、城市之间的距离、设备之间的连接等等。

如图 2.4 所示，图可以被表示为 $G = (V, E)$ ，其中 $V = \{v_1, \dots, v_N\}$ ， $E = \{e_1, \dots, e_M\}$ 。其中， V 是节点（Vertex）的集合， N 是节点的个数， E 是边（Edge）的集合， M 是边的个数。对于每一条边 e ，都包括发送方（sender）和接收方（receiver），分别用 s 和 r 代表

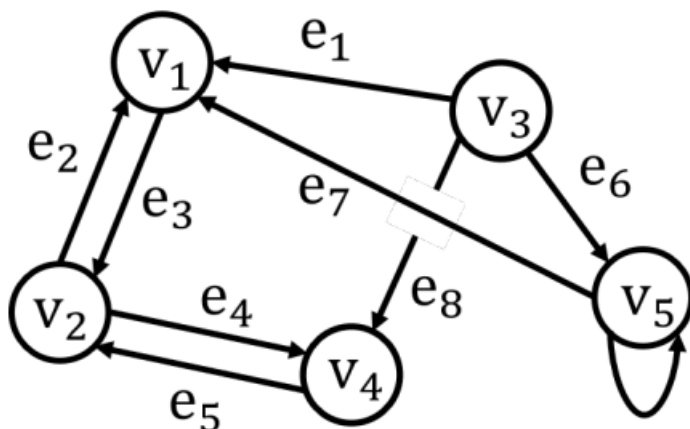


图 2.4 图结构示意图

2.3.2 图网络与图卷积

图网络（Graph Neural Network, GNN）是一种深度学习模型，用于处理图结构数据，如社交网络、蛋白质分子结构等。与传统的深度学习模型不同，图网络不仅能够处理固定大小的输入，还能够处理可变大小的图结构数据。图网络可以自适应地对不同的图结构进行处理，并生成节点或边的嵌入表示，以便进行节点或边的分类、聚类、预测等任务。

图网络可以分为基于图卷积网络（Graph Convolutional Network, GCN）的方法和基于图注意力网络（Graph Attention Network, GAT）的方法两种。GCN 基于卷积神经网络的思想，通过卷积操作对图上的节点进行信息聚合；而 GAT 则是通过自注意力机制对不同节点之间的关系进行建模，实现了对节点的自适应特征提取。

图卷积类似于传统卷积神经网络（Convolutional Neural Networks, CNN）在图像上的应用，但是不同于 CNN 将输入视为像素的网格，GCN 将输入视为节点和边的图形。

GCN 的目标是学习一个函数，将每个节点的特征向量与它的邻居节点的特征向量组合起来，

来预测每个节点的标签或其他任务。这个函数可以表示为一个矩阵形式，其中每一行代表一个节点的特征向量，每一列代表与该节点相邻的节点，矩阵元素代表节点之间的连接权重。

GCN 中的卷积操作不同于传统 CNN 中的卷积操作，它是基于邻居节点的特征向量进行计算，而不是基于像素的邻域。具体来说，GCN 中的卷积操作就是将每个节点的特征向量与邻居节点的特征向量加权求和，然后再应用激活函数。下面的公式展示了常用的图卷积网络分层传播规则：

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \quad (2.5)$$

其中， $\tilde{A} = A + I_N$ 是无向图 G 加自连接的邻接矩阵， I_N 是单位矩阵， $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ ， $W^{(l)}$ 是第 l 层特定的可训练权重矩阵。 $\sigma(\cdot)$ 表示激活函数，例如 $\text{ReLU}(\cdot) = \max(0, \cdot)$ 。 $H^{(l)} \in \mathbb{R}_{N \times D}$ 为第 l 层的激活矩阵。

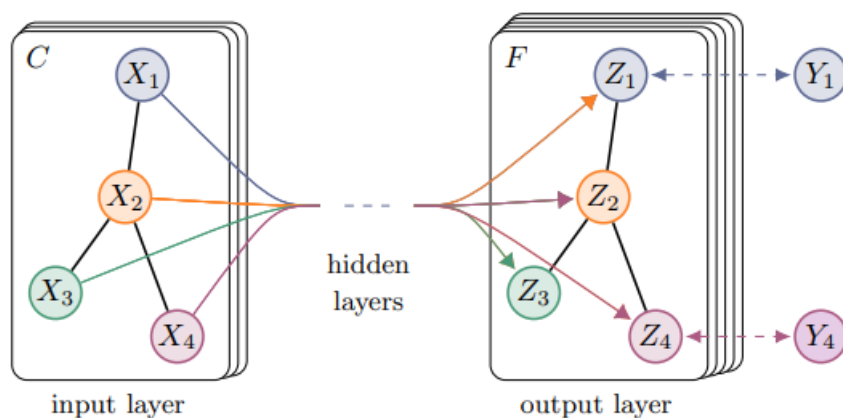


图 2.5 图网络结构示意图

3 数据、算法与模型

3.1 汽车模型网格数据

3.1.1 构建汽车网格数据集

本文采用的原始数据集为奇瑞汽车提供的两百个汽车模型 STEP 文件。它是一种中立的文件格式，可以用来在不同的 CAD 软件之间共享和交换 3D 模型数据。在 STEP 文件中，3D 模型被表示为一系列实体和属性。文件中包含了每个实体的几何形状、位置和方向等信息，还包括了实体之间的关系和约束等。STEP 文件还支持多种实体类型，包括曲面、体积、点云、线和文本等。由于 STEP 文件具有中立性和可扩展性，因此在跨平台和跨软件交换 3D 数据时被广泛应用。图 3.1 为使用 CAD 软件打开 STEP 文件的示例

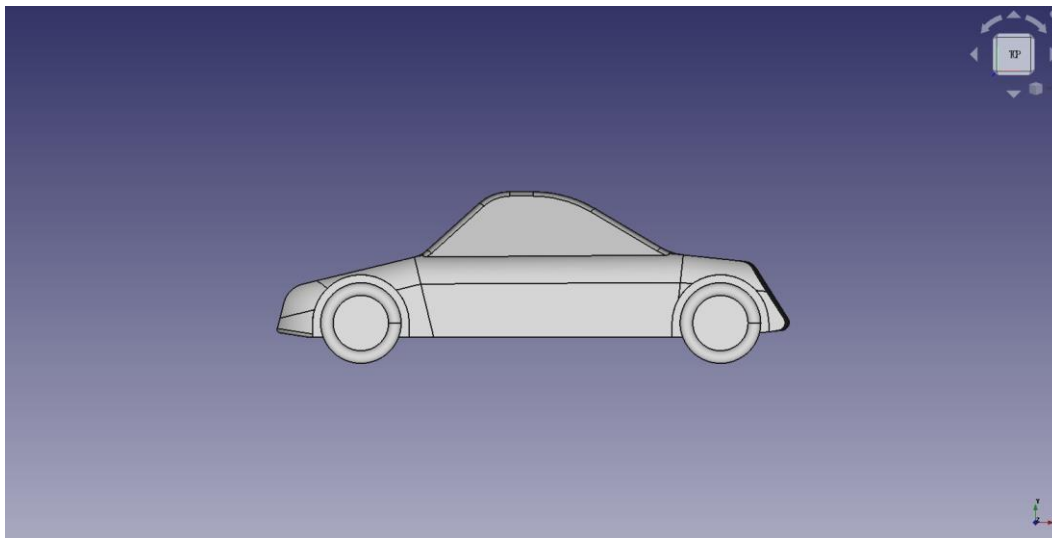


图 3.1 汽车模型 STEP 文件示例图

本文基于 gmsh 网格生成库构建汽车网格数据集，针对汽车 CAD 模型的外形及结构数据，利用 Delaunay 算法进行三角网格剖分，得到能够表征汽车流体结构特征的网格数据，进而利用图网络模型预测汽车风阻系数。本文所使用的网格生成参数如下表所示

表 3.1 gmsh 网格生成参数

参数名称	参数含义	参数值
Element size	单元尺寸	50
Mesh size grading	网格尺寸分级	1.0
Curvature	曲率	0.4
Optimize quality	是否启用质量优化	1
Smoothing.	是否启用平滑处理	1

图 3.2 为使用 gmsh 划分汽车网格后得到图形化示例数据：

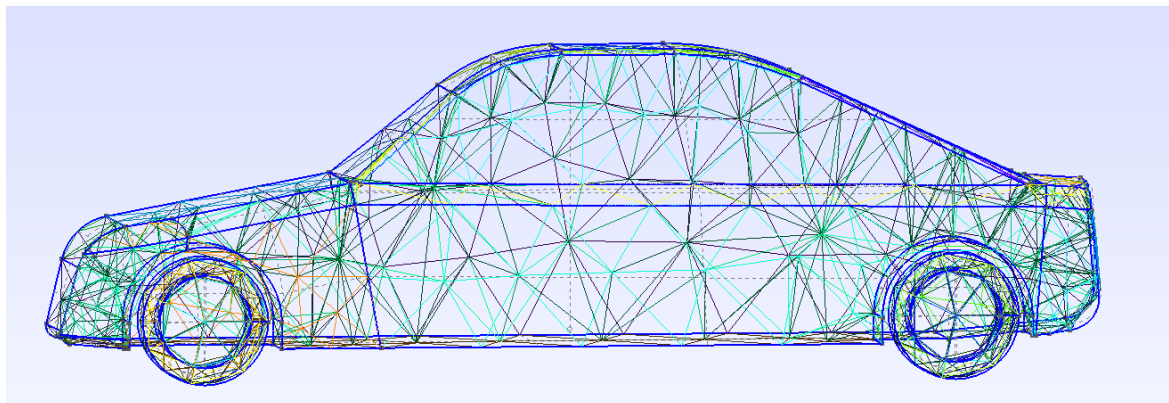


图 3.2 汽车网格示例数据

3.1.2 数据集整理与预处理

本文采用的 MeshgraphNets 模型基于图结构化的网格数据，因此为了使用该模型预测汽车风阻系数，需要对汽车网格数据进行预处理，在保留其结构特征的基础上将其转换为图结构数据。

具体来说，将三维网格的网格节点集 Nodes 与网格边集合 Connections 自然地转换为图数据中的节点和边，提取每个节点的空间三维坐标作为节点特征。边的特征包括起点和终点的节点序号，以及边的空间向量。边的空间向量是通过终点减去起点坐标的三维向量属性。此外，本文通过法向量法，基于网格单元的法向量计算每条网格边的迎风系数 $wind_c^{[27]}$ ，具体的计算公式如下所示：

$$wind_c = \frac{N_1 \cdot N_2}{||N_1|| * ||N_2||} \quad (3.1)$$

其中， N_1 和 N_2 是网格边 c 的相邻单元的法向量， $||N_1|| * ||N_2||$ 是 N_1 和 N_2 的范数数据格式如表 3.1 所示：

表 3.1 汽车图网格数据

特征	描述	范例
C_d	汽车模型的风阻系数	0.35972
Elements	网格单元集合	/
Nodes	网格节点集合	/
Connections	网格边集合	/
$Position_i$	节点的空间三维坐标	(0.0007847,0.0266234,0.0130512)
$Connection_i$	节点之间的连接属性，包括 sender（发送方）和 receiver（接收方）	{(0, 1)}

续表 3.1

$Position_c$	边的相对空间位置（向量方向），从起点指向终点	(0.00123520,0.0962283,1.5369145)
$Norm_e$	网格单元的法向量	(0, 0.65319749, 0.75733697)
w_c	边的迎风系数	0.56964

3.2 模型设计

3.2.1 MeshGraphNets

本文采用一种具有编码-处理-解码的基于图网络的预测算法模拟有限元分析，该算法包含三个步骤——编码器 ENCODER、处理器 PROCESSOR 与解码器 DECODER。步骤如下所示：

$$\mathcal{X} \rightarrow \mathcal{G}^0 \rightarrow \dots \mathcal{G}^M \rightarrow \mathcal{Y} \quad (3.1)$$

其中， \mathcal{X} 代表初始输入的图数据， \mathcal{G} 代表每个隐藏层的反复迭代更新后的图网络， \mathcal{Y} 代表输出的预测数据，在本文中即指汽车风阻系数，详细步骤如图 3.3 所示

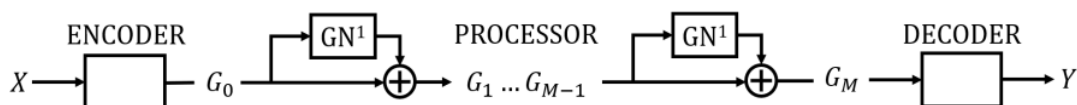


图 3.3 编码-处理-解码架构示意图

A. 编码器 ENCODER

ENCODER 是一个用于将输入网格数据转换为中间表示（latent representation）的神经网络模块。ENCODER 通过多层感知机将网格数据映射到高维空间，因而可以有效地提取输入网格的隐藏特征。这种结构具有较强的自适应性，能够自动提取和学习有用的特征。

在本文中，编码器接收的输入为 $N_v \times 3$ 维的节点特征张量和 $N_e \times 5$ 维的边特征向量，并通过一个两层的感知机将图网络特征映射为 128 维的隐藏向量。编码器架构如图 3.4 所示

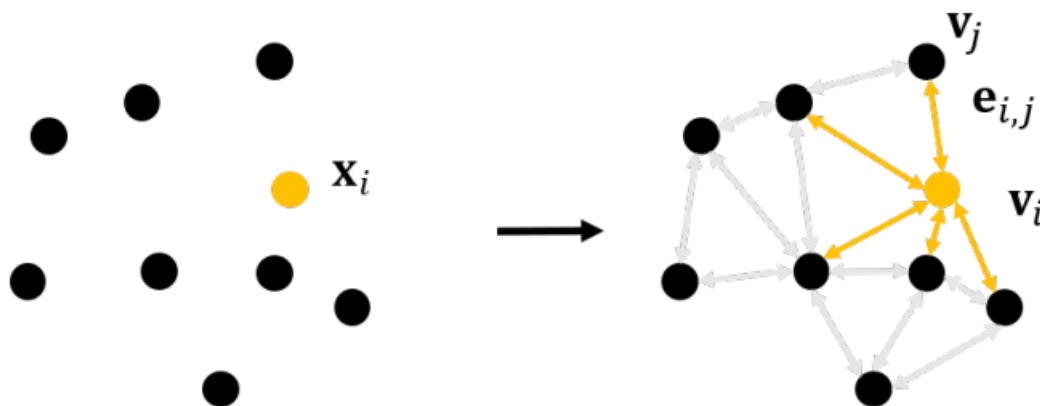


图 3.4 编码器示例图

B. 处理器 PROCESSOR

处理器主要负责接受编码器生成的特征向量，并通过消息传递机制不断为节点和边聚合邻域特征，更新节点嵌入。

如图 3.5 所示，本文的处理器架构使用具有相同编码结构的 M 个图网络结构。通过 M 个图网络学习消息传递机制，计算节点之间的交互信息，从输入图 G_0 ，迭代生成一系列更新的隐藏图 G_1, \dots, G_M 。 G_M 包含一组单独的网络参数，通过下列公式所示按顺序更新下一个图 G_{M+1} 的参数。

$$e'_{i,j} \leftarrow f^E(e_{i,j}, v_i, v_j), v'_i \leftarrow f^V\left(v_i, \sum_j e_{i,j}\right) \quad (3.2)$$

其中， $e'_{i,j}$ 代表节点 i 和节点 j 之间的边在图 G_{M+1} 的参数信息； v'_i 表示节点 i 在 G_{M+1} 的参数信息。

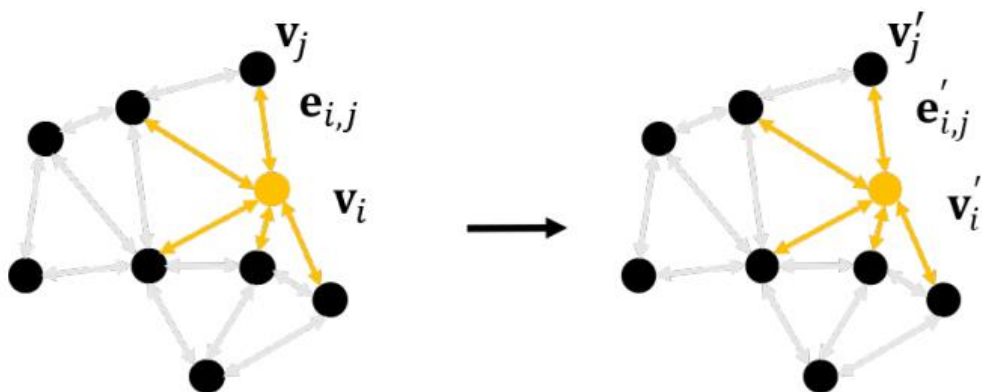


图 3.5 处理器示例图

C. 解码器 DECODER

如图 3.6 所示，DECODER 是一个用于将处理后的中间表示重新映射到输出网格的神经网络模块。DECODER 使用多层感知机架构对嵌入向量做映射处理，与 ENCODER 相似。它将中间表示转换为输出网格的节点嵌入，并使用卷积神经网络对其进行平滑处理。最后，神经网络通过多层感知机将图网络中的节点嵌入转换为图级的输出，即预测的汽车风阻系数。

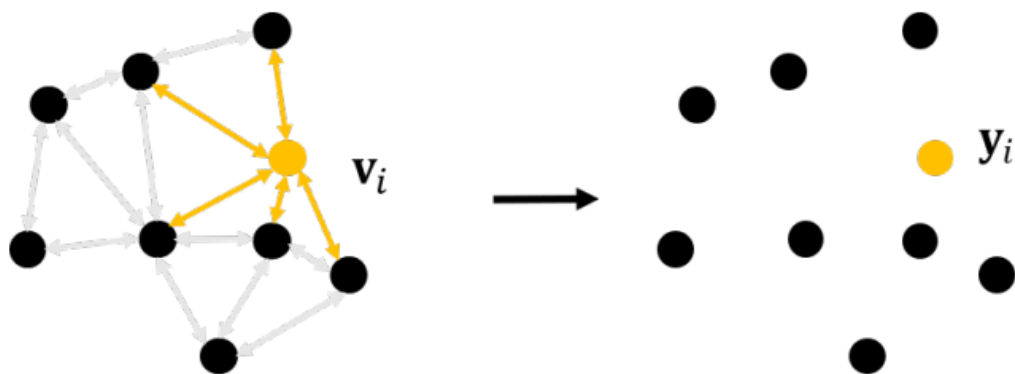


图 3.6 解码器示例图

3.2.2 节点采样算法 GraphSAGE

GraphSAGE 是一种图神经网络模型，用于学习节点嵌入（node embedding）。它的基本思想是，将每个节点的特征信息及其邻居节点的特征信息聚合起来，生成节点的嵌入表示。GraphSAGE 具有一定的灵活性和可扩展性，可以用于不同的图结构和应用场景。

传统的图算法通常采用基于矩阵计算或图遍历的方式进行计算，而 GraphSAGE 使用了深度学习的方法进行节点表征学习。GraphSAGE 通过神经网络模型学习节点的低维向量表示，以捕捉节点之间的结构和语义信息。此外，传统的图算法通常在全局范围内进行计算，涉及所有节点和边。而 GraphSAGE 采用了局部邻居采样的策略，仅选择每个节点的部分邻居节点进行聚合和更新。这样可以使得 GraphSAGE 在处理大规模图数据时更高效，并且可以学习到节点的局部结构信息。

GraphSAGE 的核心组件是聚合函数（aggregator），其作用是将节点的特征信息和邻居节点的特征信息聚合起来。GraphSAGE 提出了三种聚合函数：Mean Aggregator、Max Pooling Aggregator 和 LSTM Aggregator。

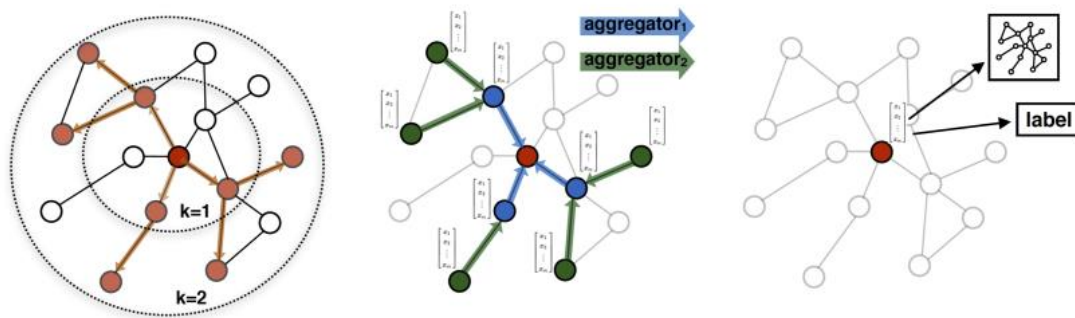


图 3.7 GraphSAGE 模型原理图

以下是 GraphSAGE 的详细过程：

- （1）初始化嵌入表示：对于每个节点，初始化其嵌入表示向量。可以将其初始化为一个随机向量，或者使用节点的原始特征向量。
- （2）聚合邻居节点信息：对于每个节点，使用聚合函数将其邻居节点的特征信息聚合成一个向量，表示邻居节点的特征信息。聚合函数的输入是邻居节点的特征向量，输出是聚合后的向量。
- （3）生成嵌入表示：将聚合后的邻居节点特征信息向量和节点自身的特征信息向量进行拼接，得到节点的嵌入表示向量。
- （4）迭代更新嵌入表示：对于每个节点，重复上述过程多次，直到得到最终的嵌入表示向量。在每次迭代中，使用上一次迭代得到的嵌入表示向量作为节点特征信息的输入。
- （5）损失函数计算：使用嵌入表示向量预测节点的标签或其他任务，并计算损失函数。可以使用梯度下降等方法优化损失函数，更新嵌入表示向量。

GraphSAGE 的伪代码如图 3.8 所示。

Algorithm 1: GraphSAGE embedding generation (i.e., forward propagation) algorithm

Input : Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth K ; weight matrices $\mathbf{W}^k, \forall k \in \{1, \dots, K\}$; non-linearity σ ; differentiable aggregator functions $\text{AGGREGATE}_k, \forall k \in \{1, \dots, K\}$; neighborhood function $\mathcal{N}: v \rightarrow 2^{\mathcal{V}}$

Output : Vector representations \mathbf{z}_v for all $v \in \mathcal{V}$

```

1  $\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V};$ 
2 for  $k = 1 \dots K$  do
3   for  $v \in \mathcal{V}$  do
4      $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\});$ 
5      $\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k))$ 
6   end
7    $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$ 
8 end
9  $\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$ 

```

图 3.8 GraphSAGE 算法伪代码

3.3 模型训练

3.3.1 噪声注入

噪声注入是一种在神经网络训练中向输入数据或模型参数中引入随机噪声的技术。它的目的是改变输入数据或参数的分布，以提高模型的鲁棒性、泛化能力和对未见数据的适应能力。此外噪声注入也可以视为一种正则化技术，有助于控制模型的复杂度。噪声的存在可以引导模型学习简化的表示，避免过度拟合训练数据。这有助于提高模型的泛化性能，并防止模型在训练集上出现过拟合的情况。

本文将噪声注入应用到了如下模块：

(1) 在模型参数更新过程中，为参数添加随机噪声。这可以有助于避免模型陷入局部最优解，并提高模型对不同训练样本的学习能力。本文通过在图网络参数更新时添加服从高斯分布的随机噪声 $N(0, \sigma_v = 0.02)$ 来改变参数的值

(2) Dropout: Dropout 是一种特殊形式的噪声注入技术，它通过在训练过程中随机关闭一部分神经元来减少模型的过拟合^[28]。关闭的神经元相当于从模型中删除，这样可以强制模型依赖于其他神经元的组合来进行预测，从而增强模型的泛化能力

3.3.2 数据标准化

本文在图神经网络中使用数据标准化，即对每个节点的输入特征进行规范化处理，使得输入的特征在一定程度上符合标准正态分布，同时，还通过反向操作将输出的标准化数据恢复为原始数据，以便将预测风阻系数与真实值进行比较。标准化的意义在于：

(1) 提高训练稳定性：标准化可以使得每个特征的尺度都相同，避免了某些特征对损失函数的影响过大，从而提高了训练的稳定性。

(2) 提高泛化能力：标准化可以降低特征之间的相关性，使得模型更容易学习到不同的特征，从而提高了模型的泛化能力。

本文采用 LayerNorm 对于特征进行归一化，如公式所示：

$$y = \frac{x - E(x)}{\sqrt{Var(x) + \epsilon}} * \gamma + \beta \quad (3.3)$$

其中， $E(x)$ 表示 x 的均值， $Var(x)$ 表示 x 的方差， ϵ 为固定常数，用于防止标准差为零的情况。

本文设定初始值为 $\epsilon = 1e - 8$, $\gamma = 1$, $\beta = 0$

3.3.3 损失函数与优化算法

A. 损失函数

本文采用 L2 损失函数，也称为平方误差损失（Squared Error Loss）或均方误差损失（Mean Squared Error Loss），是一种常用的回归任务的损失函数。它衡量了模型的预测值与真实值之间的差异的平方。L2 损失函数的计算公式如下：

$$L = \frac{1}{n} * \sum (y_{pred} - y_{true})^2 \quad (3.4)$$

其中， y_{pred} 表示模型的预测值， y_{true} 表示真实值。通过将预测值与真实值之间的差异求平方，并求和所有样本的差异，然后除以样本数量，得到 L2 损失。

L2 损失函数的特点是将差异的平方作为损失，因此对于差异较大的样本，其损失值会较大，对差异较小的样本，损失值会较小。这使得模型更加关注那些与真实值差异较大的样本，有助于优化模型对于这些样本的预测。但 L2 损失函数对离群值较为敏感，因为平方操作会放大离群值的影响，可能导致模型过度拟合离群值。

B. Adam 算法

本文使用 Adam 算法对模型学习率进行自适应调整。该算法结合了动量（Momentum）和自适应学习率方法，能够在训练过程中自动调整学习率，并具有良好的收敛性能和泛化能力。本文设定初始学习率 $l = 0.01$ ，衰减系数 $\delta = 0.99$

C. 早停法

早停法（Early Stopping）是一种用于防止模型过拟合的正则化技术，通常用于监督学习任务中。它基于验证集上的性能表现来决定何时停止模型的训练，以避免过拟合并获得更好的泛化能力。本文采用第三类停止标准，具体来说，假设 $E_{opt}(t)$ 是迭代次数 t 时取得的最好的验证集误差， $E_{va}(t)$ 是迭代次数 t 时模型在验证集上的误差，则有：

$$E_{opt}(t) = \min_{t' \leq t} E_{va}(t') \quad (3.5)$$

我们定义泛化损失（generalization loss）变量 $GL(t)$ ，它描述的是在当前迭代周期 t 中，泛化误差相比较目前的最低的误差的一个增长率，具体公式如下：

$$GL(t) = 100 * \left(\frac{E_{va}(t)}{E_{opt}(t)} - 1 \right) \quad (3.6)$$

本文规定，当验证集错误在连续 $patience$ 个周期内出现增长，即当泛化错误在连续 $patience$ 个周期内增长的时候停止训练（ $patience$ 默认为 10）。我们假设这样的现象表明了过拟合，这个停止标准可以度量局部的变化，即在训练阶段，允许误差可以比前面最小值高很多时候保留。

3.4 本章小结

本章通过三角剖分算法处理汽车 STEP 文件,得到汽车网格数据集。针对汽车风阻系数预测,提出了基于图网络结构的回归预测模型,将深度学习算法应用在风阻系数预测问题上。同时结合节点采样算法,针对大规模数据集进行训练效率与泛化能力的优化。此外,定义模型对于数据的预处理方法,如噪声注入以及标准化等方法。最后,设定了模型训练过程中采用的损失函数以及学习率优化等算法。

装

订

线

4 实验与结果分析

针对第 3 章构建的汽车数据集以及设计的基于图网络的汽车风阻系数预测算法，本章针对相关参数及算法设计了实验，并根据实验结果分析模型性能与改进方向，对比了不同算法的优化效果。

4.1 实验环境

本文的计算环境基于 Linux 操作系统和 Tensorflow1.15.0 深度学习框架搭建，CPU 为 Inter(R) Xeon(R) Platinum 8255C，显卡为 GeForce RTX 2080 Ti，具体实验环境与主要依赖库如下表所示。

表 4.1 本课题开发环境

类型	说明/版本	版本/型号	说明/版本
CPU	Intel(R) Xeon(R) Gold	6330	内存：80G，内核： 15G
GPU	NVIDIA	RTX 3090	显存：24G，CUDA： V11.1
操作系统	Ubuntu	V20.4	
开发语言	Python	V3.10	
IDE	PyCharm	V2023.1	Python 集成开发环境

表 4.2 本课题主要依赖库

名称	版本	说明
Numpy	V1.23.5	数值计算依赖库
Pytorch	V1.12.1+cu113	深度学习依赖库
Matplotlib	V3.7.0	绘图与可视化依赖库
Scikit-learn	V1.2.1	机器学习库

4.2 实验设计

4.2.1 基本实验设计

本文将两百个汽车模型数据集按 8：1：1 的比例随机划分训练集、验证集和测试集，即训练集包含 160 条数据，验证集和测试集包含 20 条数据。本文对于模型共训练 100 个周期，数据情况如图 4.1 所示。

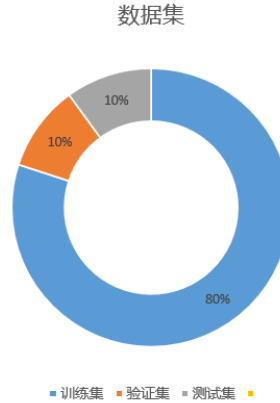


图 4.1 数据集划分

本实验记录每个周期模型在训练集和验证集上的 MSE、RMSE、MAE 等评估指标，用于评估模型的收敛情况和预测能力。

此外，本文分别对比了在相同的超参数设置下，不使用节点采样的 MeshGraphNet 模型与加入 GraphSAGE 采样算法后的 MeshGraphNets 的训练效果，以评估采样算法的优化能力

4.2.2 泛化实验设计

在工程实践中，CAE 仿真软件不需要学习就可以对任何汽车模型进行工程问题求解，因此这就要图神经网络对未见的汽车模型也能够进行风阻系数预测。在泛化实验中，笔者对本课题的神经网络模型进行了泛化能力实验，从数据集中随机选择 10% 的汽车网格模型作为测试集，测试集不参与模型的训练过程，仅作为泛化实验的参与数据集，以模拟神经网络模型面对全新汽车模型的场景。

4.2.3 对比实验设计

为对比检测 GraphSAGE 算法的采样优化效果，本文将采用《FASTGCN: FAST LEARNING WITH GRAPH CONVOLUTIONAL NETWORKS VIA IMPORTANCE SAMPLING》中提到的 FastGCN^[18] 算法以及《Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks》提到的 Cluster-GCN^[20] 算法作为对比。

A. FastGCN

FastGCN 是一种基于图卷积网络（GCN）的图神经网络模型，旨在加速 GCN 模型的训练和推断过程。在传统的图卷积网络中，由于邻居节点的采样和权重的随机性，梯度的估计存在较大的方差，这会导致训练过程不稳定，收敛速度慢。而 FastGCN 则对传统的 GCN 公式进行修改，从积分的视角引入无偏估计：

$$\tilde{h}^{(l+1)}(v) = \int \hat{A}(v, u) h^{(l)}(u) W^{(l)} dP(u), \quad h^{(l+1)} = \sigma(\tilde{h}^{(l+1)}(v)) \quad (4.1)$$

其中 $\tilde{h}^{(l+1)}(v)$ 表示第 $l+1$ 层的节点 v 的表示向量。 $\hat{A}(v, u)$ 表示节点 v 和节点 u 是邻域关系， $W^{(l)}$ 表示第 l 层的线性变换，即权重参数矩阵。

相较于传统的 GCN 公式，FastGCN 的公式中新增了 $P(u)$ 一项， $P(u)$ 表示节点的概率分布，即

节点 u 是以 $P(u)$ 的概率从潜在的无限的节点集 V 中采样出来的。由于难以精确计算节点的概率分布，原文提出使用蒙特卡洛法去计算上述积分公式的近似解，具体来说，就是在训练过程中对节点进行分层的重要性采样，下一层的嵌入函数由是前一层的积分变换得到的。

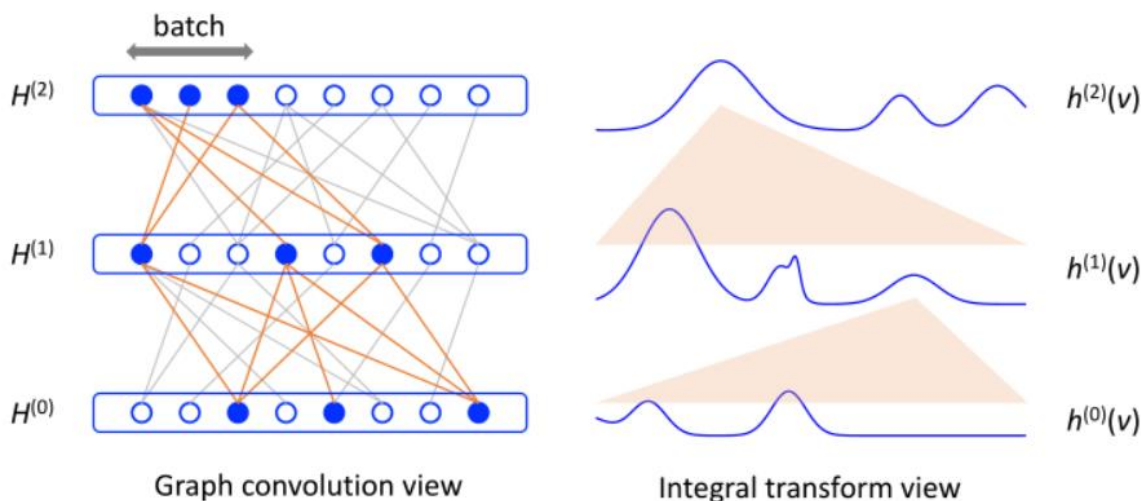


图 4.2 FastGCN 的卷积视图

重要性采样的思想是根据节点的重要性程度选择邻居节点进行计算，即更加关注那些对模型的训练和预测有较大影响的邻居节点。具体来说，FastGCN 中通过计算输入特征和邻接矩阵的归一化点积，得到节点之间的相似度，并基于归一化相似度的对数概率进行多项式抽样，从而得到被采样节点的索引。通过重要性采样，FastGCN 在保持模型性能的同时，减少了计算量，加快了训练速度。这种采样策略可以根据图的特点和任务需求进行调整，使得网络可以更好地适应不同的图结构和节点分布。

B. Cluster-GCN

Cluster-GCN 是一种用于图卷积网络的高效训练方法，旨在解决传统图卷积网络在大规模图上训练效率低下的问题。该模型通过将大规模图分割成多个子图，并在每个子图上独立地进行训练，从而实现高效的训练和推理过程。GCN 的核心原理包括图的聚类分割以及邻域信息的聚合：

(1) 图的分割：首先，将原始图分割为多个子图，每个子图包含一组节点。分割图的方法可以采用现有的图划分算法，例如 Metis 算法。通过分割图，可以将整个训练过程并行化，加快训练速度。

(2) 聚合邻居信息：在每个子图中，进行图卷积操作时，只考虑每个节点的局部邻居节点。为了获取每个节点的邻居信息，采用一种聚合策略。具体来说，每个节点首先聚合其邻居节点的特征，然后将聚合后的特征作为节点的输入。这样可以减少计算量，并提高模型的效率。比其传统的 GCN 算法，Cluster-GCN 仅在子图范围内进行特征的聚合，因而有效地避免了多层迭代导致的邻域扩展，如图 4.3 所示。

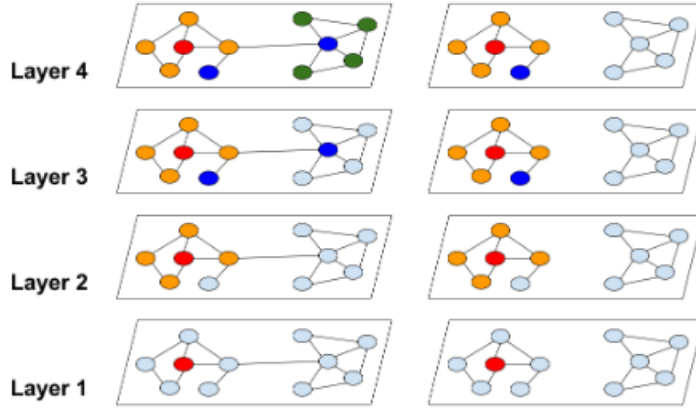


图 4.3 Cluster-GCN 与传统 GCN 的对比图

Cluster-GCN 的优点在于它能够将大规模图的训练问题转化为小规模子图的训练，从而提高了训练的效率和可扩展性。通过图的分割和子图训练，Cluster-GCN 能够在大规模图上进行高效的训练，并获得较好的性能。然而，Cluster-GCN 也存在一些问题，例如子图分割的策略选择、子图间信息传递的限制等，需要根据具体的应用场景和数据集特点进行调整和优化。

4.3 模型性能评估指标

本文研究汽车风阻系数的回归预测问题，将模型训练所产生的汽车风阻系数预测值与真实值做对比，进而求得模型的性能评估参数。以下是相关误差分析方法，其中 \hat{y}_i 为真实值， y_i 为回归预测值， n 为回归预测的数据个数。

(1) 平均绝对误差 (Mean Absolute Error, 简称 MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (4.1)$$

(2) 均方误差 (Mean Squared Error, 简称 MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.2)$$

(3) 均方根误差 (Root Mean Absolute Error, 简称 RMAE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (4.3)$$

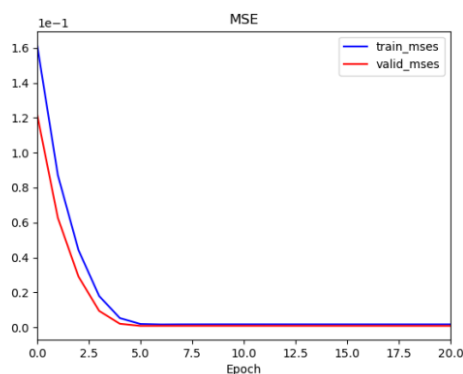
相较于平均绝对误差，均方误差和均方根误差由于引入了平方操作，因而对少量的数据异常将会更加敏感。综合多种评估指标，我们希望能够更好地评价模型在各种不同情况下的预测性能。

4.4 实验结果与分析

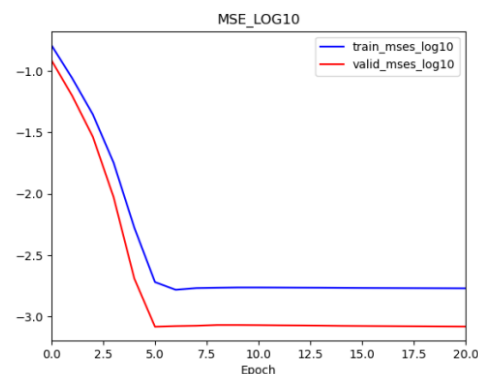
4.4.1 基础实验

A. MeshGraphNets 模型预测实验

本实验将全部数据分别按 80%、10%、10%的比例随机划分为训练集、验证集、测试集，损失函数采用 L2，即均方误差，并在 GPU 上训练 50 个 epoch。此外，由于先前并未有提出针对三维网格的汽车风阻系数预测任务，因此本文参考 GCN 中的方案，设计了一个面向汽车网格数据的图卷积网络 GCN 作为本文的基线（baseline）方法，用于与本课题的模型进行比较，以验证本课题模型的特征提取方法的有效性。实验结果如图 4.4 和 4.5 所示：

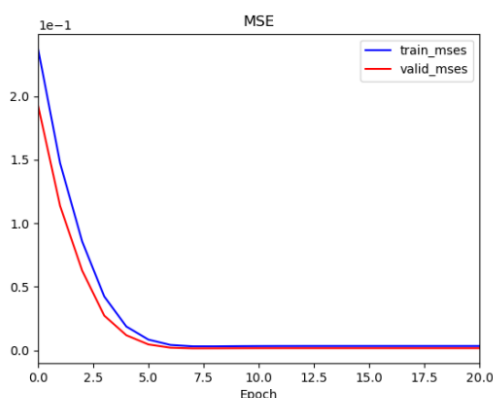


(a) 均方误差随训练步数示意图

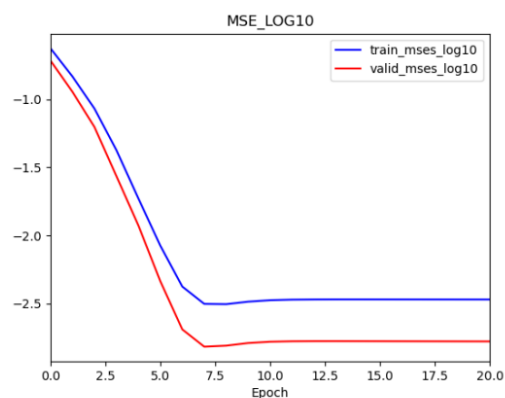


(b) 均方误差取对数随训练步数示意图

图 4.4 本课题模型的均方误差随训练次数变化图



(a) 均方误差随训练步数示意图



(b) 均方误差取对数随训练步数示意图

4.5 本课题基线的均方误差随训练次数变化图

图中结果表明，无论是本课题模型（MeshGraphNets）还是本课题基线（GCN），均在训练步数达到 20 次时趋于收敛，分析实验结果数据，我们得到以下结论：

（1）MeshGraphNets 和 GCN 模型对汽车网格数据的风阻预测均能达到比较好的效果，其中 MeshGraphNets 无论在训练集还是验证集上都有着更高的训练精度。相较于 GCN 方法，MeshGraphNets 的均方误差仅为前者的 0.38 倍，平均绝对误差仅为前者的 0.67 倍。这证明本文所采用的编码器-处理器-解码器架构在提取汽车模型特征方面有一定的优势

（2）在平均训练时间和预测时间上，MeshGraphNets 都比 GCN 方法要长 100%左右。由于 MeshGraphNets 在处理器架构中需要经过多轮的消息传递机制以聚合邻域特征信息，因此本结果也符合笔者的实验预期

表 4.3 基础实验定量评估指标表

方法名称	均方误差	均方根误差	平均绝对误差	平均训练时间	平均预测时间
本课题模型	0.000615	0.024799	0.031216	37.557256	0.026883
本课题基线	0.001619	0.040240	0.046574	19.999671	0.009715

B. GraphSAGE 采样算法优化实验

针对 MeshGraphNets 模型面向大规模图的训练时间问题，本文采用 GraphSAGE 采样算法对其进行优化，预期在保证预测精度的同时降低模型的训练时间。GraphSAGE 算法的核心是随机采样和特征聚合，因此本课题拟定基于不同的采样大小与聚合器对 GraphSAGE 的算法性能进行对比实验：

（1）采样大小：图 4.4 展示了示例汽车网格数据的节点邻域大小。图中大部分节点的邻域大小为 7，少部分节点的邻域大小为 5 和 8，因此本课题拟定分别基于 2、4、6 的采样大小对 GraphSAGE 算法进行对比实验，聚合器默认采用 MEAN。

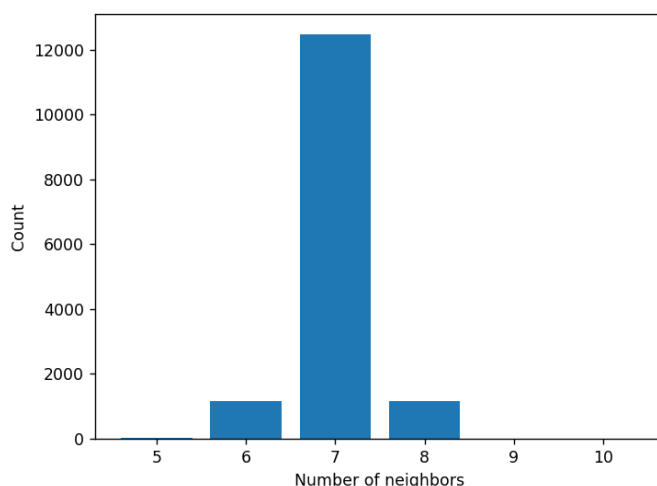


图 4.6 节点邻域大小示例图

图 4.7 展示了 GraphSAGE 采样大小对比实验的实验结果。本文分别在采样大小等于 2、4、6 时使用模型训练汽车网格数据，并记录各模型在验证集上的平均训练时间与平均均方误差。分析

数据可知，随着采样尺寸的增大，模型的平均训练时间不断增大，且平均均方误差减小，表明模型有着更好的预测精度。当采样大小 2 与 4 之间的精度差异，要远远大于采样大小 4 与 6 之间的差异。该实验结果比较符合笔者的实验预期，由于采样尺寸的增加会导致需要处理更多的节点和边，从而增加了模型的计算和训练时间。此外，更大的采样尺寸意味着需要更多的邻居节点特征进行聚合和更新，这也会增加计算的复杂性和时间成本。

平均均方误差减小，表明模型有着更好的预测精度：随着采样尺寸的增大，模型能够访问更多的节点和邻居信息，从而更全面地学习图的结构和特征。这导致模型能够更准确地预测节点的标签或执行其他图任务，因此平均均方误差减小。更大的采样尺寸提供了更多的信息，有助于提高模型的预测能力和精度。

采样大小的增加使模型能够更充分地捕捉图的全局结构和特征。当采样尺寸较小时，模型只能访问到较少的邻居节点信息，导致模型预测的精度较低。但是一旦采样尺寸增加到一定程度，模型已经能够获取足够的邻居信息，继续增加采样尺寸对模型的精度影响不大。因此，误差曲线随着采样大小的增加而逐渐趋于平缓。

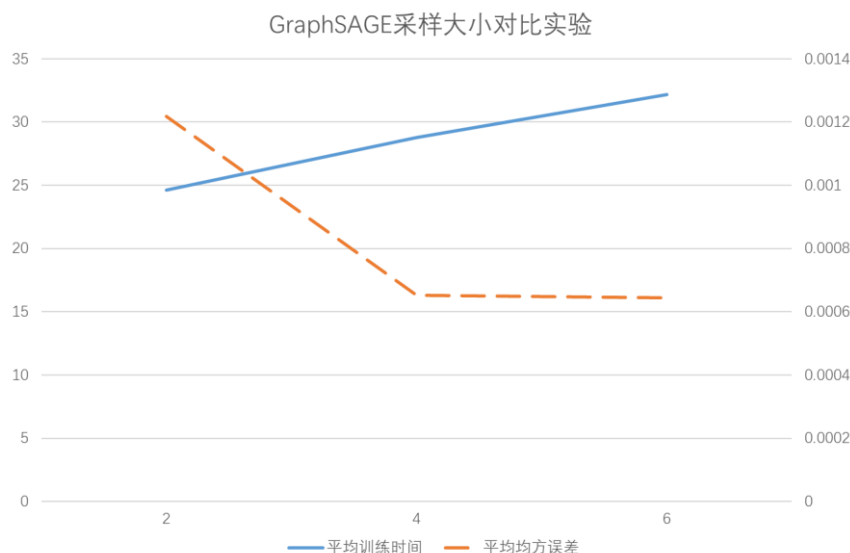


图 4.7 GraphSAGE 采样大小对比实验

（2）聚合器：本课题基于 GraphSAGE 原论文的实验路线，采样了平均聚合器 MEAN 和最大池化聚合器 MAX 分别作为特征聚合器，以检测其对采样特征提取的处理能力。

此外，为了防止在比较 GraphSAGE 聚合器时无意中进行“超参数调整”，我们对所有 GraphSAGE 变体遍历相同的一组超参数（根据验证集上的性能选择每个变体的最佳设置），可能的超参数值集是通过早期验证测试确定的。

基于采样所示大小实验的结果，本课题在采样大小为 4 的前提下，对比了两种聚合器和原模型的性能指标，如图 4.8 所示。

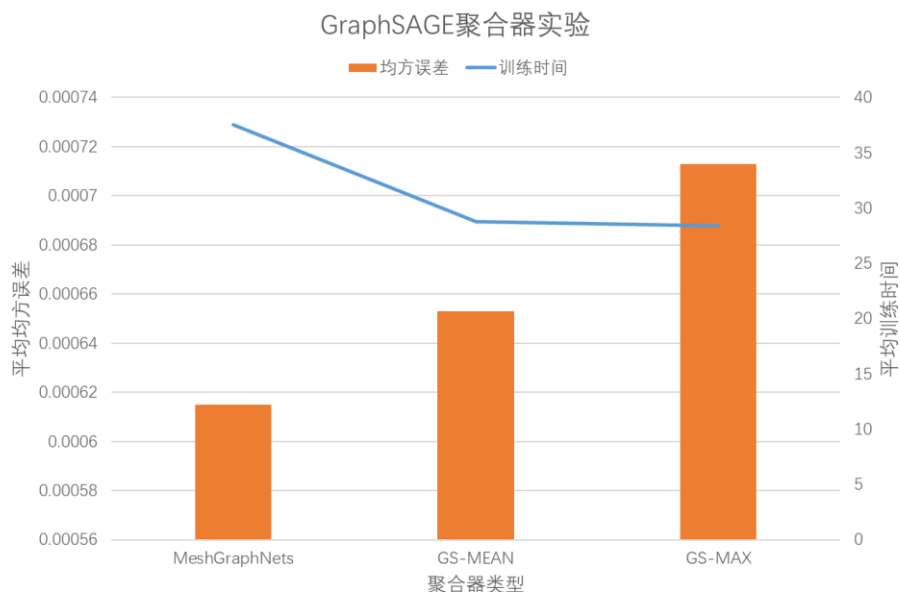


图 4.8 GraphSAGE 聚合器对比实验

表 4.4 聚合器实验定量评估指标表

模型	聚合器	均方误差	均方根误差	平均绝对误差	平均训练时间	平均预测时间
MeshGraphNets		0.000615	0.024799	0.031216	37.557256	0.026883
GraphSAGE	MEAN	0.000653	0.025553	0.035941	28.76173	0.022852
GraphSAGE	MAX	0.000713	0.026702	0.040854	28.39152	0.021425

表 4.4 给出了两种聚合器的性能指标对比。相较于不加采样的原始 MeshGraphNets 模型，两种 GraphSAGE 聚合器均对模型的平均训练时间和预测时间有一定的优化效果，且两者时间相近。在训练精度上，采用 MEAN 聚合器的 GraphSAGE 要有着更为接近原始模型的预测精度，其均方误差仅增长 6% 左右，而采用 MAX 聚合器的 GraphSAGE 模型增长了约 16% 的均方误差。这可能是由于采用 MEAN 聚合器能够更好地融合邻居节点的信息，提供更准确的预测结果，而采用 MAX 聚合器可能会导致一些邻居节点的重要信息丢失，从而降低了预测精度。

4.4.2 泛化实验

本文使用 GCN 方法作为基线模型，并分别使用原始的 MeshGraphNets 与加入节点采样的 MeshGraphNets 在测试集上进行泛化实验，其中采样使用 MEAN 聚合器，采样大小为 4。实验结果如表 4.3 所示，其中基线方法设计与第 4.2.1 小节中相同，可以看到本课题的神经网络模型在泛化实验场景下整体性能仅有略微地下降，模型泛化能力较好。

经过采样的 MeshGraphNets 在预测时间上有所降低，均方误差有所提升，符合基础实验结果，说明本文得到的预测模型与实验结论具有较好的泛化性。值得注意的是，采样后的 MeshGraphNets

模型在平均绝对误差上反而略有降低，这种现象可能暗示了原模型在某些样本上存在较大的偏差，即对某些特定情况下的预测结果较为不准确。这可能是因为模型在处理某些特殊样本或者数据集的某些特征时存在局限性或缺陷，导致在这些情况下预测的绝对误差较大。而经过采样后的预测模型能够在一定程度上避免这种由局部异常值而导致的误差。

表 4.3 泛化实验定量评估指标表

方法名称	均方误差	均方根误差	平均绝对误差	平均预测时间
GCN	0.002358	0.048559	0.058817	19.999671
MeshGraphNets	0.000579	0.024062	0.025863	0.029201
MeshGraphNets+GraphSAGE	0.000680	0.026086	0.017495	0.024564

4.4.4 对比实验

为了比较评估 GraphSAGE 算法的采样优化效果，本文引入 FastGCN 算法和 Cluster-GCN 算法作为对比算法。FastGCN 算法和 Cluster-GCN 算法均为拥有充足的理论基础并被广泛应用于工程实践的算法。且三者分别对应三种不同的采样层次，即节点采样、分层采样与子图采样。本文希望通过对比实验，更好地检测本文所提出的模型针对汽车网格数据的采样效果。通过对比实验结果，我们也可以更好地理解不同算法的适用范围和潜在优化空间，为进一步改进图神经网络算法的设计和应用提供有价值的参考。

表 4.4 对比实验定量评估指标表

方法名称	均方误差	均方根误差	平均绝对误差	平均预测时间	平均训练时间
GraphSAGE	0.000653	0.025553	0.035941	0.022852	28.76173
FastGCN	0.000636	0.025214	0.020394	0.042215	34.41847
Cluster-CGN	0.000668	0.025838	0.017730	0.024564	32.57351

表 4.4 展示了三种采样算法的各项性能评估指标结果。可以看出，三者在训练精度上相差不大，但 GraphSAGE 算法有着最短的平均预测时间和训练时间。FastGCN 在均方误差和平均绝对误差上均有着较好的表现，但采样效率一般。Cluster-GCN 算法有着最低的平均绝对误差，但均方误差却是三者中最高的，且该算法的收敛周期也是最长的。三种算法的数据对比如图 4.9 所示。

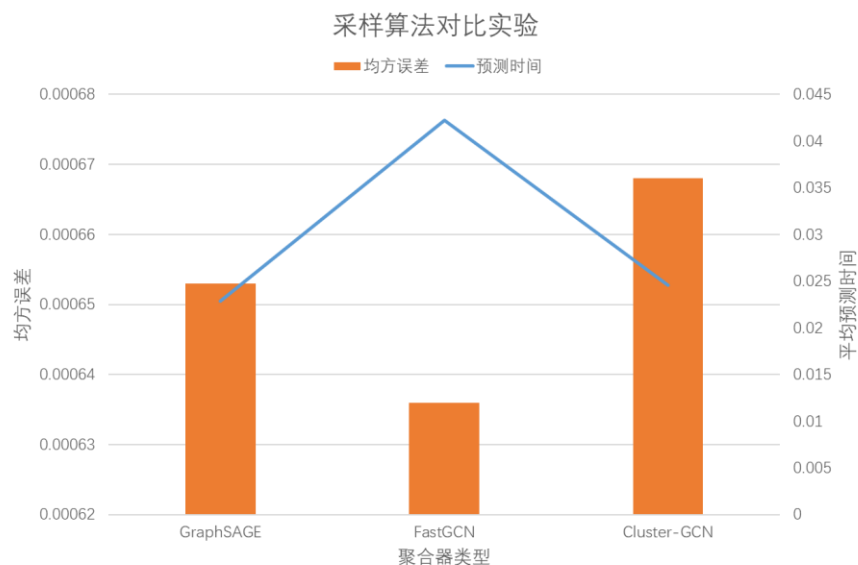


图 4.9 采样算法对比实验结果图

总结来看，在各种采样算法的对比中，GraphSAGE 算法有着可观的训练精度和优秀的训练效率，这可能得益于其高效简洁的随机采样策略。基于重要性采样的 FastGCN 有着最优的预测精度，但由于采样处理过程的复杂化，反而降低了训练效率。Cluster-GCN 算法基于子图内部的特征采样，也能够较好地提取汽车的局部结构特征信息，但也因此拖延了模型收敛的速度。

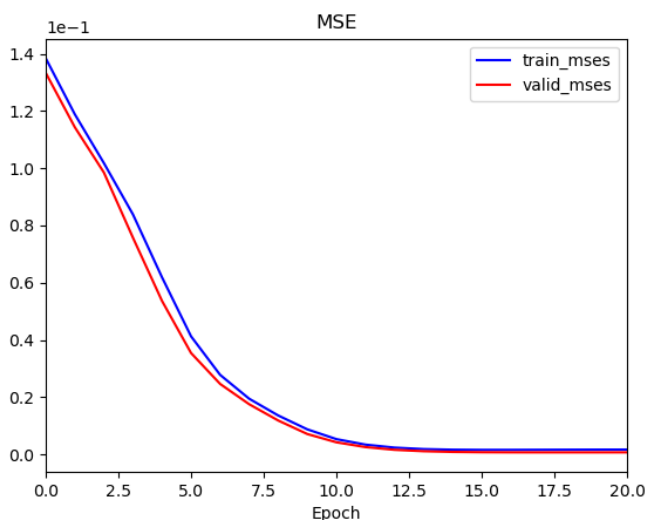


图 4.10 FastGCN 模型的均方误差随训练次数变化图

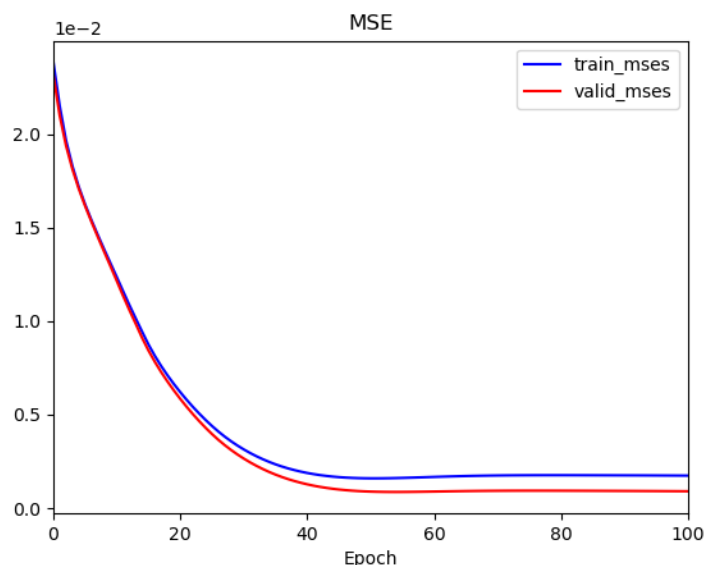


图 4.11 Cluster-GCN 的均方误差随训练次数变化图

4.5 本章小结

本章展示了实验的环境配置与模型结果评估的标准。针对第 3 章所生成数据集以及提出的汽车风阻预测模型，设计了三种实验：（1）基础实验，该预测模型的均方误差 MSE 达到 0.000615，相较于基线模型有较好的提升。且经过采样处理后，模型在保证训练精度的同时，能够大幅降低训练与预测时间。（2）泛化能力测试实验，本节基于测试集对风阻预测模型的泛化能力进行评估，MSE 仅为 0.000579，较基线模型有大幅提升，表明模型的泛化能力优秀。（3）对比实验，将本文所用的节点采样算法与分层采样和子图采样算法做比较，结果表明本文采样的节点采样算法有着优秀的预测精度与训练效率。

5 总结与展望

5.1 全文总结

本文对汽车模型进行网格化处理，并使用图神经网络模型训练汽车网格数据，预测汽车风阻系数。本文的主要工作包括：1、基于 GMSH 软件处理汽车模型文件，得到表征汽车结构的 wangge 数据。2、本文提出一种基于三维网格图神经网络模型，通过编码器-处理器-解码器架构提取汽车结构特征信息，进而预测汽车风阻系数。3、在图网络模型中加入节点采样算法，通过随机采样和特征聚合优化模型训练速度，同时保证预测精度。4、最后，本文设计了基础实验、泛化实验和对比实验。通过本文模型与基线模型的对比，各种采样算法的对比，最终证明了本文所提出的汽车风阻系数预测模型拥有较优性能，在收敛速度和模型误差两个方面均优于现有算法。

5.2 未来展望

相较于传统的风洞模拟实验和有限元分析方法，使用图神经网络预测汽车风阻系数能够大大降低汽车模型设计的优化成本，对于汽车制造业有莫大意义。本文所设计的汽车风阻系数预测模型较好地满足了实验预期，但在预测精度和时间上仍有进一步优化的空间：

（1）适应汽车网格数据的扩展。相较于本文所使用的数据集，工业界所使用的汽车模型数据往往更加复杂多变，有着更为精细巧妙的结构。且实际的生产场景下往往要求极高的网格精度，这将大大增加数据的复杂程度，因此对模型的特征提取能力和优化能力有着更高的要求。

（2）优化模型的采样算法。本文所采用的节点级采样算法的随机采样在面向简单的图数据场景中有着可观的优化能力，并且相对于复杂的分层采样和子图采样算法有着一定优势。但随着数据精度的提高和训练标准的提升，模型的采样能力也会面临更大的挑战。

（3）面向其他领域的工业应用场景。本文所使用的模型能够有效处理简单的汽车模型结构，但是否能够对相似的工程产品进行模拟仿真预测，仍待进一步研究。由于基于数值分析的有限元分析能够普适性地模拟各种工业生产应用场景，因此本文模型的泛用性仍待进一步提升。

参考文献

- [1] 中国新闻网. 国务院关于印发《中国制造 2025》的通知[EB/OL]. (2015-05-08). http://www.gov.cn/zhengce/content/2015-05/19/content_9784.htm
- [2] 刘林, 张瑞秋主编. 计算机辅助设计[M]. 广州: 华南理工大学出版社, 2015: 1-11
- DES MARAIS D J, STRAUSS H, et al. Carbon isotope evidence for the stepwise oxidation of the Proterozoic environment[J]. Nature, 1992, 359: 605-609.
- [3] 王自勤. 计算机辅助工程(CAE)技术及其应用[J]. 贵州工业大学学报: 自然科学版, 2001, 30(4): 4.
- [4] 曾攀. 有限元分析及应用[M]. 北京: 清华大学出版社, 2004: 362-36
- DES MARAIS D J, STRAUSS H, et al. Carbon isotope evidence for the stepwise oxidation of the Proterozoic environment[J]. Nature, 1992, 359: 605-609.
- [5] 吴磊, 庞毅, 潘俊潮. 基于有限元分析的新能源汽车侧面柱碰结构优化研究[J]. 西部交通科技, 2022(01): 187-190. DOI:10.13282/j.cnki.wccst.2022.01.051.
- DES MARAIS D J, STRAUSS H, et al. Carbon isotope evidence for the stepwise oxidation of the Proterozoic environment[J]. Nature, 1992, 359: 605-609.
- [6] 袁洁. 有限元分析技术在汽车零部件设计中应用的策略与路径[J]. 时代汽车, 2019(20): 88-89.
- DES MARAIS D J, STRAUSS H, et al. Carbon isotope evidence for the stepwise oxidation of the Proterozoic environment[J]. Nature, 1992, 359: 605-609.
- [7] 陈泽彬. 基于深度学习的有限元计算研究[D]. 哈尔滨工业大学, 2020. DOI:10.27061/d.cnki.ghgdu.2020.003450.
- DES MARAIS D J, STRAUSS H, et al. Carbon isotope evidence for the stepwise oxidation of the Proterozoic environment[J]. Nature, 1992, 359: 605-609.
- [8] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 424-432). Springer.
- [9] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 234-241). Springer.
- [10] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 652-660). IEEE.
- [11] Wu, J., Zhang, C., Xue, T., Freeman, B., & Tenenbaum, J. (2016). Learning a probabilistic latent space of object shapes via 3D Generative-Adversarial Modeling. In Advances in Neural Information Processing Systems (NIPS) (pp. 82-90).

- [12] Hanocka, R., Giryas, R., & Fleishman, S. (2019). MeshCNN: A Network with an Edge. ACM Transactions on Graphics (TOG), 38(4), 86.
- [13] Pfaff T, Fortunato M, Sanchez-Gonzalez A, et al. Learning mesh-based simulation with graph networks[J]. arXiv preprint arXiv:2010.03409, 2020.
- [14] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.
- [15] Robert C, Casella G, Robert C P, et al. Metropolis–hastings algorithms[J]. Introducing Monte Carlo Methods with R, 2010: 167-197.
- [16] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[J]. Advances in neural information processing systems, 2017, 30.
- [17] Chen J, Zhu J, Song L. Stochastic training of graph convolutional networks with variance reduction[J]. arXiv preprint arXiv:1710.10568, 2017.
- [18] Chen J, Ma T, Xiao C. Fastgcn: fast learning with graph convolutional networks via importance sampling[J]. arXiv preprint arXiv:1801.10247, 2018.
- [19] Huang W, Zhang T, Rong Y, et al. Adaptive sampling towards fast graph representation learning[J]. Advances in neural information processing systems, 2018, 31.
- [20] Chiang W L, Liu X, Si S, et al. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks[C]//Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019: 257-266.
- [21] Zeng H, Zhou H, Srivastava A, et al. Graphsaint: Graph sampling based inductive learning method[J]. arXiv preprint arXiv:1907.04931, 2019.
- [22] (美) 约翰·F.休斯 (John F. Hughes), (美) 詹姆斯·D.福利. 计算机图形学原理及实践 (进阶篇第 3 版) [M]. 北京: 机械工业出版社, 2021: 635-668
- [23] Lee, D. T., & Schachter, B. J. (1980). Two algorithms for constructing a Delaunay triangulation. International Journal of Computer & Information Sciences, 9(3), 219-242.
- [24] Watson, D. F. (1981). Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. The Computer Journal, 24(2), 167-172.
- [25] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016
- [26] Barnes J A, Harary F. Graph theory in network analysis[J]. Social networks, 1983, 5(2): 235-244
- [27] Bridson, R. (2008). Fluid simulation for computer graphics. AK Peters/CRC Press.
- [28] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.

谢辞

四年光阴荏苒，转眼已经走到毕业的路口，回顾四年来在同济的点点滴滴，只觉得心中感慨良多。大学四年的学习生活中，体验了形形色色的经历，结识了来自天南海北的朋友，一路上不乏挫折与悔憾，但末了心中只余感激和希望。我想要首先感谢我的导师饶卫雄老师。从初期的确认选题到文献调研，从模型设想到实验搭建，饶老师用自己的专业知识和实践经验给予了我莫大帮忙。饶老师常结合自己的人生阅历，为我们提供恳切真诚的为人处世的经验之谈，让我受益良多。我还要感谢帮忙指导我毕业设计的林滢师兄以及一同学习奋斗的小组成员，若非与他们的交流互助，我实难完成繁多复杂的实验工作。当然，我还要衷心感谢我的父母，在大学的四年，以及此前的二十年中，他们支持我的生活，鼓励我的学业，塑造我的性格。父母的殷切期盼始终是我努力奋斗的初心，亦是我不断前行的动力。最后，我还要感谢我的舍友，软件学院的同学与老师，感谢在大学里相识的每一个人，你们组成了我四年的会议，填满了我青春的底色。本科的终点，亦是人生的起点，希望我们都能坚守奋斗的初心，书写人生的华章，为社会繁荣与国家昌盛，砥砺前行。

装

订

线