

基于图扩散求解器的物联网拓扑鲁棒性优化方法

深度学习课程设计 李季桥 2023244030

项目地址: https://github.com/Lijiqiao/Deeplearning_Class.git

一、摘要

在工业物联网环境中,随着大量设备的不断接入,网络拓扑的规模也在迅速扩展。节点设备的失效或恶意攻击可能会增加其周围节点的传输负载,进而导致网络的“级联崩溃”。因此,构建具有高鲁棒性的网络拓扑显得尤为重要。现有的拓扑优化算法主要依赖自回归启发式求解器,但这些方法通常效率低下,且难以适应大规模网络。本文提出了一种基于图扩散模型的拓扑优化方法,旨在提高现有算法的时间效率和适用规模,优化网络拓扑的鲁棒性。

二、问题简介

工业物联网的广泛应用使得网络拓扑结构日益复杂和庞大。节点设备可能会因故障或攻击而失效,这种失效会传递给周围节点,增加其传输负载,进而引发整个网络的级联崩溃。这种现象突显了构建鲁棒网络拓扑的重要性。传统的拓扑鲁棒性优化方法多依赖于自回归启发式求解器,这类方法在处理大规模网络时效率较低,难以实现全局最优解。为了克服这些局限,本文探索了一种基于图扩散模型的新型拓扑优化方法,旨在提升拓扑优化的效率和扩展性,从而增强网络的鲁棒性。

三、相关工作

过去,人们对拓扑鲁棒性优化问题进行了大量研究。

自回归求解器

Herrmann HJ等人提出了洋葱网络[1],并使用爬山法和贪心算法等自回归求解器来构建鲁棒网络,但这些算法难以扩展到大规模节点。

自回归启发式求解器

Qiu T、Liu J等人开发了基于启发式遗传算法的ROCKS[2]和BTW-Motif[3]算法,通过构建最优种群解决大规模拓扑网络的鲁棒性优化问题,但仍然受限于启发式算法的收敛速度慢、容易陷入局部最优等问题。

改进的自回归启发式求解器

Qiu T、Yang X、Chen N[4] [5]等学者提出了基于马尔可夫决策的组合优化问题求解器,即基于深度强化学习的DDL算法和ROMEM算法,相较于启发式算法,这些方法在寻找全局最优解方面表现更优。然而,由于深度强化学习需要探索巨大的搜索空间且计算资源消耗高,这些方法依然难以扩展到大规模节点。

这些研究大多依赖改进自回归启发式求解器来解决问题,但自回归启发式求解器本身的缺陷使其难以应对大规模问题。而解决拓扑鲁棒性优化问题本身,并不一定需要依赖自回归启发式求解器。

基于图扩散模型的求解器

最近,基于图扩散模型的组合优化问题求解器已被证明在性能上取得了显著的突破 [6]。图扩散模型通过在单位时间内添加噪声来优化数据,相较于传统的自回归启发式求解器,图扩散模型能够并行地对所有变量进行全局求解,因此在求解时间和全局最优解方面均优于其他算法。

综上所述，这些研究大多依赖改进自回归启发式求解器来解决问题，但自回归启发式求解器本身的缺陷使其难以应对大规模问题。而解决拓扑鲁棒性优化问题本身，并不一定需要依赖自回归启发式求解器。我们计划设计一种基于图扩散模型的拓扑鲁棒性优化求解器，以在大规模节点拓扑中实现优异的表现。

四、数据

本设计通过BA模型模拟工业物联网拓扑网，数据集为一系列无标度BA网络。数据集分为两组，分别是50节点100条边的小型网络，和200节点400条边的扩展网络。数据集已开源在Github链接中。

五、方法

为了实现基于图扩散模型的拓扑鲁棒性优化，我们设计了如下步骤：

1. 数据预处理：

对无标度BA网络数据进行预处理，包括节点和边的归一化处理，以确保输入数据的标准化。

预处理的伪代码如下：

```
def load_graphs(data_file):
    with open(data_file, "rb") as f:
        graphs = pickle.load(f)
    return graphs

def has_node_features(graph):
    """
    检查图是否有任何节点属性
    参数：
        graph: networkx.Graph, 单个图
    返回：
        bool, 图中是否有节点属性
    """
    for node in graph.nodes(data=True):
        node_data = node[1]
        if node_data: # 检查节点是否有任何属性
            return True
    return False

def filter_graphs_with_features(graphs):
    """
    过滤掉没有节点属性的图
    参数：
        graphs: list, 图数据列表

    返回：
        filtered_graphs: list, 包含节点属性的图数据列表
    """
    filtered_graphs = [graph for graph in graphs if has_node_features(graph)]
    return filtered_graphs

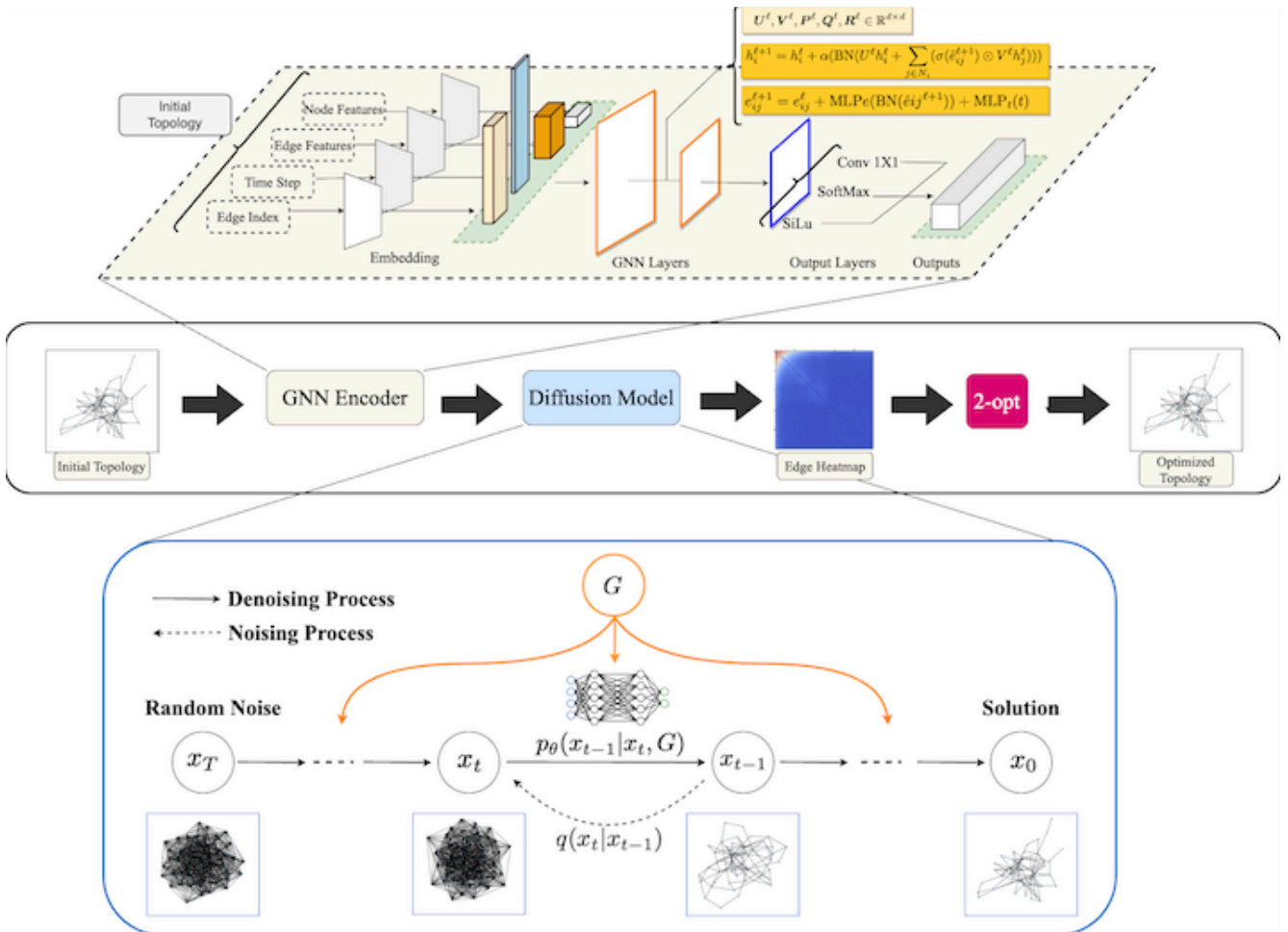
def save_graphs(graphs, output_file):
    with open(output_file, "wb") as f:
```

```
pickle.dump(graphs, f)
```

2. 模型设计：

构建基于图扩散模型的拓扑优化算法，通过在网络节点和边上添加噪声，模拟节点失效和攻击情况，进而优化网络的鲁棒性。

模型设计架构图如下：



图神经网络编码器：

图神经网络编码器旨在将BA拓扑编码成可扩散的潜空间向量，包含多个模块，每个模块都有特定的功能。整体模型结构包括节点嵌入、边嵌入、位置嵌入、时间步嵌入、多层GNN层以及输出层。

模块名称	层类型	输入维度	输出维度	描述
GNNEncoder	模型整体			整体模型结构，包括节点嵌入、边嵌入、位置嵌入、时间步嵌入、多层GNN层以及输出层
node_embed	Linear	256	256	节点嵌入层，用于将节点特征转换为隐含表示
edge_embed	Linear	256	256	边嵌入层，用于将边特征转换为隐含表示
pos_embed	PositionEmbeddingSine	N/A	N/A	节点位置嵌入层，用于将节点位置嵌入到特征中
edge_pos_embed	ScalarEmbeddingSine	N/A	N/A	边位置嵌入层，用于将边位置嵌入到特征中
time_embed	Sequential	256	128	时间步嵌入层，用于将时间步特征转换为隐含表示，包括两层Linear和一层ReLU激活函数
out	Sequential	256 → [256, num_nodes, num_nodes]	256 → 2	输出层，用于将多层GNN层的输出特征转换为最终输出，包含归一化、激活和卷积操作
layers	ModuleList	[batch_size, num_nodes, 256]	[batch_size, num_nodes, 256]	多层GNN层，每一层都包含U、V、A、B、C线性变换和归一化操作
time_embed_layers	ModuleList	128	256	多层时间步嵌入层，每一层包含ReLU激活和Linear层，用于时间步特征的多层处理
per_layer_out	ModuleList	[batch_size, num_nodes, 256]	256	多层输出层，每一层包含归一化、SiLU激活和Linear层，用于每层GNN层输出特征的处理

扩散模型：

扩散模型（Diffusion Models）是一类基于概率的方法，用于生成数据或解决复杂的优化问题。其基本思想是通过逐步将噪声引入数据，形成一个多步骤的正向过程，然后通过一个学习到的逆向过程逐步去噪，以恢复原始数据分布。

2-opt算法：

2-opt算法是一种用于解决旅行商问题（TSP, Traveling Salesman Problem）的局部搜索算法。旅行商问题是经典的组合优化问题，要求找到一条经过每个城市且仅经过一次的最短路径。2-opt算法通过反复交换路径中的两个边段，以寻找更短的路径，从而改进初始解。

下面是2-opt算法的伪代码：

```
def two_opt(route):
    best_route = route
    improved = True
    while improved:
        improved = False
        for i in range(1, len(best_route) - 2):
            for j in range(i + 1, len(best_route)):
                if j - i == 1:
                    continue # 需要跳过相邻节点
                new_route = best_route[:]
                new_route[i:j] = best_route[j-1:i-1:-1] # 反转路径段
                if calculate_total_distance(new_route) <
calculate_total_distance(best_route):
                    best_route = new_route
                    improved = True
    if not improved:
        break
```

```

return best_route

def calculate_total_distance(route):
    # 计算路径的总距离
    distance = 0
    for i in range(len(route) - 1):
        distance += distance_between(route[i], route[i+1])
    distance += distance_between(route[-1], route[0])
    return distance

```

3. 训练与优化：

采用梯度下降等优化算法对模型进行训练，通过多次迭代不断调整模型参数，以获得最优的拓扑结构。

4. 实验验证：

通过与传统的自回归启发式求解器进行对比实验，验证基于图扩散模型的拓扑优化方法在时间和规模上的优越性。

5. 评估指标：

物联网（IoT, Internet of Things）拓扑结构的**鲁棒性**是指网络在面对节点故障或恶意攻击时保持其功能和性能的能力。鲁棒性的评估对于设计和维护可靠的物联网网络至关重要。由于物联网中的设备数量庞大且分布广泛，节点的故障或攻击可能导致网络部分或全部失效。因此，衡量网络的鲁棒性并进行优化，是提升物联网系统整体可靠性的重要手段。

物联网拓扑的鲁棒性可以通过模拟节点攻击或故障来评估。常用的方法是通过删除节点并观察网络中最大联通子图的变化情况。最大联通子图（Largest Connected Component, LCC）的大小在每次攻击后的变化能够反映网络的鲁棒性。

计算生成图鲁棒性的算法伪代码如下：

```

def calculate_robustness(G, node):
    """
    计算给定图G的鲁棒性
    参数：
    G - networkx.Graph, 要计算的图
    node - int, 图中的节点数量
    返回：
    R - float, 图的鲁棒性值
    """
    sum_largest_cc = 0 # 记录每次攻击中最大联通子图的大小
    sorted_degree_descending = sorted(dict(G.degree()).items(), key=lambda item:
item[1], reverse=True)
    while len(G.edges()) > 0: # 当存在边时，获取最大联通子图
        # 模拟攻击：删除节点度数最大的节点
        G.remove_node(sorted_degree_descending[0][0])
        # 找到最大联通子图
        largest_cc = max(nx.connected_components(G), key=len)
        # 累加最大联通子图的大小
        sum_largest_cc = sum_largest_cc + len(largest_cc)
        # 重新排序节点的度数

```

```
sorted_degree_descending = sorted(dict(G.degree()).items(), key=lambda item:
item[1], reverse=True)
sum_largest_cc = sum_largest_cc + len(G.nodes())
R = float((sum_largest_cc - 1)) / float((node * (node - 1)))
return R
```

六、实验

实验环境

本次实验在Ubuntu 22.04操作系统上进行，采用了4块NVIDIA A100 GPU，以确保实验的高效运行。实验所使用的硬件和软件配置如下：

- 操作系统：Ubuntu 22.04
- GPU：4块NVIDIA A100
- 深度学习框架：PyTorch

数据集

实验使用的数据集名称为 `list_0419_50`，其具体配置如下：

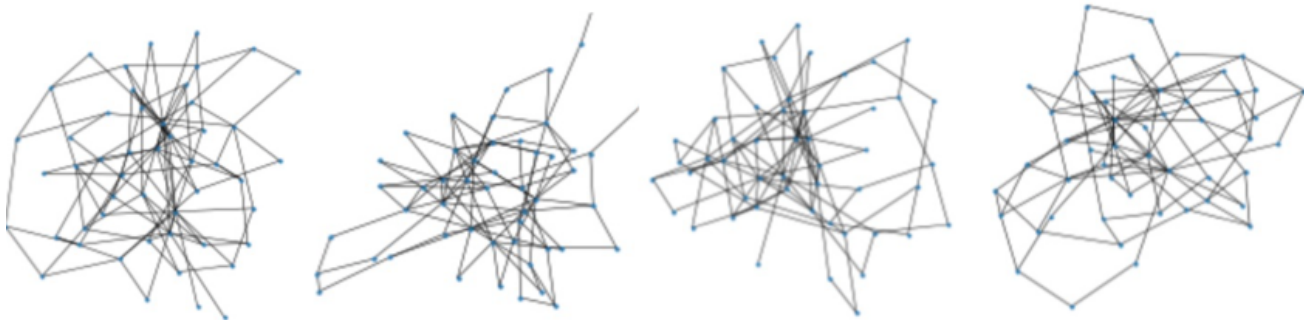
- 数据集大小：200
- 最大节点数：50
- 数据集名称：list_0419_50

训练配置

训练过程中的具体参数设置如下：

- 训练批次大小 (batch_size)：32
- 学习率初始值 (lr_init)：0.001
- 学习率衰减 (lr_decay)：0.999
- 最大训练轮数 (max_epoch)：5000
- 动量 (momentum)：0.9
- 样本间隔 (sample_interval)：1
- 保存间隔 (save_interval)：1
- 是否打乱数据 (shuffle)：是
- 权重衰减 (weight_decay)：0.0
- 加权损失 (weighted_loss)：是
- 随机种子 (seed)：1234

实验优化的BA网络可视化：



评估鲁棒性指标为：

第16个图的 R 指标为：0.23795918367346938

第19个图的 R 指标为：0.20489795918367346

第25个图的 R 指标为：0.21020408163265306

第29个图的 R 指标为：0.19142857142857142

七、结论

本项目提出了一种基于图扩散模型的物联网拓扑鲁棒性优化方法。通过将扩散模型应用于网络拓扑优化，模型能够有效地在节点故障或攻击的情况下维护网络的功能和性能。实验在Ubuntu 22.04操作系统和4块NVIDIA A100 GPU上进行，数据集包含200个最大节点数为50的图结构。实验结果表明，所提出的方法在拓扑鲁棒性优化任务中具有较高的性能和可靠性。

具体来说，实验表明，通过逐步将数据扩散为带噪声的潜变量，并通过学习到的逆向过程逐步去噪，能够有效地恢复原始数据分布，从而优化网络拓扑的鲁棒性。所提出的方法在处理复杂的组合优化问题时展现出了显著的优势，尤其是在大规模节点拓扑的鲁棒性优化中表现优异。

未来的工作可以进一步优化模型结构，例如引入更多的层次和复杂的特征提取机制，以提升模型的性能。此外，可以在更大规模和更多样化的数据集上进行测试，以验证模型的泛化能力和适应性。通过不断改进和优化，该方法有望在实际的物联网应用中发挥更大的作用，提升系统的整体可靠性和性能。

参考文献

- [1] Albert, R., Jeong, H., & Barabási, A.-L. (2000). Error and attack tolerance of complex networks. *Nature*, 406(6794), 378-382.
- [2] Qiu T, Liu J, Si W, et al. Robustness optimization scheme with multi-population co-evolution for scale-free wireless sensor networks[J]. *IEEE/ACM Transactions on Networking*, 2019, 27(3): 1028-1042.
- [3] Qiu T, Zhang L, Chen N, et al. Born this way: A self-organizing evolution scheme with motif for internet of things robustness[J]. *IEEE/ACM Transactions on Networking*, 2022, 30(6): 2644-2657.
- [4] Chen N, Qiu T, Mu C, et al. Deep Actor-Critic Learning-Based Robustness Enhancement of Internet of Things[J]. *IEEE Internet of Things Journal*, 2020, 7(7): 6191-6200.

- [5] Qiu T, Yang X, Chen N, et al. A Self-Adaptive Robustness Optimization Method With Evolutionary Multi-Agent for IoT Topology[J]. IEEE/ACM Transactions on Networking, 2023.
- [6] Sun Z, Yang Y. Difusco: Graph-based diffusion solvers for combinatorial optimization[J]. Advances in Neural Information Processing Systems, 2023, 36: 3706-3731.