文章引言

方法概括

Papers

深度强化学习解决组合优化问题的应用

1

Research Reviews of Combinatorial Optimization Methods Based on Deep Reinforcement Learning

2

该神经网络网络每次只能学习并解决单个小规模 TSP 问题实例, 对于新给定的一个 TSP 问题需要从头开始再次训练, 相对于传统算法并没有优势

2

在 1985 年提出 Hopfield 网络, 用于求解 TSP 问题以及其他组合优化问题

3

在 2015 年, Vinyals 等人将组合优化问题类比为机器翻译过程(即序列到序列的映射)，提出了可以求解组合优化问题的指针网络模型***(Pointer Network, Ptr-Net)***

3

NeurIPS

3

ICLR

3

Summary

都是"端到端 (end-to-end) 方法", 即给定问题实例作为输入, 利用训练好的深度神经网络直接输出问题的解。

上述方法在小规模问题上可以接近最优解, 但是在中大规模问题上与 LKH3、Google OR tools、Gurobi、Concorde 等专业组合优化求解器的优化能力还存在一定差距.

3

基本原理介绍

4

Pointer Network 的强化学习训练方法

4

由于监督式学习方法需要提供大量最优路径的标签数据集, 实际应用较为困难, 因此目前研究中通常以强化学习算法对模型的 W和v等参数进行训练.

4

首先需要将组合优化问题建模为马尔科夫过程, 其核心要素为状态、动作以及反馈，TSP 问题为例：

$$p_\theta(\pi \mid s) = \prod_{n}^{t=1} p_\theta\left(\pi_t \mid s, \pi_{1:t-1}\right) \qquad (3)$$

策略由神经网络参数 θ 进行参数化, 在马尔科夫过程中, 每一步动作的概率为 $p_\theta\left(\pi_t|s, \pi_{1:t-1}\right)$, 即根据已访问过的城市 $\pi_{1:t-1}$ 和城市坐标 s 计算选择下一步访问各个城市的概率, 根据链式法则累乘即可以得到城市坐标 s 到城市最终访问顺序 π 的映射 $p_\theta(\pi|s)$, 该随机策略可以建模为上节所述的指针网络模型, 其参数为 θ.

5

REINFORCE 强化学习算法

5

基于蒙特卡洛的策略梯度方法, 即不断执行动作直到结束, 在一个回合结束之后计算总反馈, 然后根据总反馈对策略的参数进行更新.
TSP为例：

$$\nabla \mathcal{L}(\theta \mid s) =$$
$$\mathbb{E}_{p_\theta(\pi|s)}\left[(L(\pi) - b(s)) \nabla \log p_\theta(\pi \mid s)\right]$$
$$\theta \leftarrow \theta + \nabla \mathcal{L}(\theta \mid s)$$

5

图神经网络 (Graph Neural Network, GNN)

5

以GNN为例

$$\mathbf{h}_v^{(t)} = \sum_{u \in N(v)} f\left(\mathbf{x}_v, \mathbf{x}_{(v,u)}^{\mathbf{e}}, \mathbf{x}_u, \mathbf{h}_u^{(t-1)}\right)$$

其中 $\mathbf{h}_v^{(t)}$ 代表节点 v 的表征向量, $N(v)$ 代表 v 的邻居节点的集合, $\mathbf{x}_v$ 是节点 v 的特征, $\mathbf{x}_{(v,u)}^{\mathbf{e}}$ 是与 v 相连的边的特征, $\mathbf{x}_u$ 是邻居节点 u 的特征, $\mathbf{h}_u^{(t-1)}$ 是邻居节点 u 在上一步更新的特征向量. 因此该公式根据节点 v 本身的特征、边的特征以及邻居节点的特征对节点 v 的表征向量进行更新, 从 $t=0$ 开始对不断对 $\mathbf{h}_v^{(t)}$ 进行更新直到收敛, 从而得到节点 v 的准确特征向量.

- **精确方法 (Exact approaches)**
  - 求解得到问题全局最优解的一类算法
  - **LINK:**https://www.baidu.com/link?url=j-plT4vC2EoZN8wBy9D8Ah8rb3aeUJtgzPzfvvgM8-a3gHyTpY6DfIdJIIKXk7Z9NqHaEDGzFTVBL_xKi9gYFdb7VhThfKjxjR1spD1dZCi&wd=&eqid=f228743e0035dbe400000005616ae700
  - 分支定界法 (Branch and Bound)
  - 动态规划法 (Dynamic Programming)
  - 割平面法(Cutting Plane)
  - Drawbacks
    - 精确方法可以求解得到组合优化问题的全局最优解, 但是当问题规模扩大时, 该类算法将消耗巨大的计算量, 很难拓展到大规模问题。
  - 解决思想
    - 分而治之：通过将原问题分解为子问题的方式进行求解通过不断迭代求解得到问题的全局最优解.
- **组合优化问题特点**：决策空间为有限点集, 直观上通过穷举法得到问题的最优解, 但是可行解数量随问题规模呈指数型增长, 无法在多项式时间内穷举得到问题的最优解。
- **近似方法 (Approximate approaches)**
  - 求解得到问题局部最优解的方法
  - 近似算法 (Approximate Algorithms)
    - 贪心算法
    - 局部搜索算法
    - 线性规划和松弛算法
    - 序列算法
  - 启发式算法 (Heuristic Algorithms)
    - 模拟退火算法
    - 禁忌搜索
    - 进化算法
    - 蚁群优化算法
    - 粒子群算法
    - 迭代局部搜索
    - 变邻域搜索
  - Drawbacks
    - 近似方法可以在可接受的计算时间内搜索得到一个较好的解.
    - 但仍然很难拓展到在线、实时优化问题.

理论与方法

Related Papers

表 1　现有算法模型、训练方法、求解问题、以及优化效果比较

Table 1　Comparison of model, training method, solving problems and performance with existing algorithms

方法类别	研究	模型以及训练方法	求解问题及优化效果
基于Pointer Network的端到端方法	2015年Vinyals等人[30]	Ptr-Net + 监督式训练	30 TSP问题: 接近最优解, 优于启发式算法. 40, 50-TSP: 与最优解存在一定差距. 凸包问题、三角剖分问题.
	2017年Bello等人[31]	Ptr-Net + REINFORCE & Critic baseline	50-TSP: 优于[30]. 100-TSP: 接近Concorde最优解. 200-Knapsack: 达到最优解.
	2018年Nazari等人[32]	Ptr-Net + REINFORCE & Critic baseline	100-TSP: 与[31]优化效果相近, 训练时间降低约60%. 100-CVRP/随机CVRP: 优于多个启发式算法.
	2018年Deudon等人[33]	Transformer Attention + REINFORCE & Critic baseline	20, 50-TSP: 优于[31]. 100-TSP: 与[31]优化效果相近.
	2019年Kool等人[34]	Transformer Attention + REINFORCE & Rollout baseline	100-TSP: 优于[30-33,37,40]. 100-CVRP、100-SDVRP、100-OP、100-PCTSP、SPCTSP: 接近Gurobi最优解, 优于多种启发式方法.
	2020年Ma等人[35]	Graph Pointer Network + HRL	20, 50-TSP: 优于[31,37], 劣于[34]. 250, 500, 1 000-TSP: 优于[31,34]. 20-TSPTW: 优于OR-Tools、蚁群算法.
	2020年Li等人[36]	Ptr-Net + REINFORCE & Critic baseline & 分解策略/参数迁移	40, 100, 150, 200, 500-两目标/三目标TSP : 优于MOEA/D、NSGA-II、MOGLS.
基于图神经网络的端到端方法	2017年Dai等人[37]	structure2vec + DQN	1 200-TSP: 接近[34]. 1 200-MVC(最小顶点覆盖): 接近最优解. 1 200-MAXCUT(最大割集): 接近最优解.
	2019年Mittal等人[38]	GCN + DQN	2k至20k-MCP(最大覆盖问题): 优于[37]. 10k, 20k, 50k-MVC: 优于[37].
	2018年Li等人[39]	GCN + 监督式训练 + 引导树搜索	实际数据集MVC、MIS(最大独立点集)、MC(极大团)、Satisfiability(适定性问题): 优于[37].
	2017年Nowak等人[40]	GNN + 监督式训练 + 波束搜索	20-TSP: 劣于[30].
	2019年Joshi等人[41]	GCN + 监督式训练 + 波束搜索	20, 50, 100-TSP: 略微优于[30,31,33,34], 优于[37].
	2019年Chen等人[47]	Ptr-Net + Actor-Critic	20-CVRP: 达到最优解. 50, 100-CVRP: 优于[32,34]、OR-Tools. 作业车间调度: 优于OR-Tools、DeepRM
深度强化学习改进的局部搜索方法	2019年Yolcu等人[48]	GNN + REINFORCE	实际数据集Satisfiability、MIS、MVC、MC、图着色问题: 更少搜索步数得到最优解、但单步运行时间长于传统算法.
	2020年Gao等人[49]	Graph Attention + PPO	100-CVPR: 优于[34]. 100-CVPRTW: 优于多个启发式方法. 400-CVRPTW: 劣于单个启发式方法, 优于其他.
	2020年Lu等人[50]	Transformer Attention + REINFORCE	20, 50, 100-CVRP: 优于[32,34,47], 以及优于OR Tools、LKH3. 且运行时间远低于LKH3.

Table 2　Comparison of end-to-end model on TSP.

方法类别	模型	TSP-20	TSP-50	TSP-100
最优	Concorde	3.84	5.70	7.76
针网络(Attention机制)	Vinyals[30]	3.88	7.66	—
	Bello[31]	3.89	5.95	8.30
	Nazari[32]	3.97	6.08	8.44
	Deudon[33]	3.86	5.81	8.85
	Deudon[33]+2OPT	3.85	5.85	8.17
	Kool[34](greedy)	**3.85(0s)**	**5.80(2s)**	**8.12(6s)**
	Kool[34](sampling)	**3.84(5m)**	**5.73(24m)**	**7.94(1h)**
基于图神经网络	Dai[37]	3.89	5.99	8.31
	Nowak[40]	3.93	—	—
	Joshi[41](greedy)	3.86(6s)	5.87(55s)	8.41(6m)
	Joshi[41](BS)	**3.84(12m)**	**5.70(18m)**	**7.87(40m)**

Table 3　Comparison of models on VRP.

模型	VRP-20	VRP-50	VRP-100
LKH3	**6.14(2h)**	**10.38(7h)**	**15.65(13h)**
Nazari[32]	6.40	11.15	16.96
Kool[34](greedy)	6.40(1s)	10.98(3s)	16.80(8s)
Kool[34](sampling)	6.25(6m)	10.62(28m)	16.23(2h)
Chen[47]	6.12	10.51	16.10
Lu[50]	**6.12(12m)**	**10.35(17m)**	**15.57(24m)**

Table 4　Summary and comparison of algorithms on different combinatorial optimization problems

组合优化问题	文献	模型细节
TSP问题	[30–36]	基于Ptr-Net架构 (Encoder-Decoder-Attention)
	[37]	GNN+DQN
	[40, 41]	GNN+监督式训练+波束搜索
	[32, 34]	基于Ptr-Net架构(Encoder-Decoder-Attention)
VRP问题	[47, 49, 50]	DRL训练局部搜索算子. [47]: Ptr-Net模型, [49]: Graph Attention模型, [50]: Transformer Attention模型.
最小顶点覆盖问题 (MVC)	[37, 38, 48]	GNN + RL
	[39]	GNN + 监督式训练 + 树搜索
	[37]	GNN + DQN
最大割集问题 (MaxCut)	[57]	Message Passing Neural Network (MPNN) + DQN
	[58] *	CNN&RNN + PPO
适定性问题 (Satisfiability)	[39, 48]	GNN + 监督式训练/RL
最小支配集问题 (MDS)	[48]	GNN + RL
	[59] *	Decision Diagram + RL
极大团问题(MC)	[39, 48]	GNN + 监督式训练/RL
最大独立集问题 (MIS)	[39]	GNN + 监督式训练 + 树搜索
	[60] *	GNN + RL + 蒙特卡洛树搜索
背包问题(Knapsack)	[31]	Ptr-Net + RL
车间作业调度问题	[47]	LSTM + RL训练局部搜索算子
装箱问题(BPP)	[61] *	LSTM + RL
	[62] *	NN + RL + 蒙特卡洛树搜索
图着色问题	[48]	GNN + RL
	[63] *	LSTM + RL + 蒙特卡洛树搜索

- 应用综述
- 网络与通信领域
  - 资源分配
    - Define
      资源分配问题是指将有限的 CPU、内存、带宽等资源分配给不同的用户或任务需求, 如虚拟网络功能部署问题、网络资源切片问题等
  - 拓扑与路由优化
    - Define
      在通信网络或者无线传感网络中, 通常需要对路由策略、传感器的连接拓扑进行优化, 以降低通信时延和成本.
  - 计算迁移
    - Define
      通过将部分计算任务从本地迁移到远程设备以解决移动终端资源受限问题的一个有效途径
      - Method
        基于深度强化学习的 DROO 计算迁移框架
- 其他领域
  - 交通领域
    - Method
      DQN 和 PPO 训练算法
  - 生产制造领域
    - 经典的车间工作流调度问题
      - Method
        采用深度强化学习方法对局部搜索的解搜索规则进行学习, 利用 Actor-Critic 深度强化学习算法对搜索规则进行优化学习
    - 置换车间工作流调度问题
      - Method
        采用指针网络模型
      - Method
        采用经典的指针网络模型, 并利用 CPLEX 求解器构造大量样本对该模型进行监督式训练
      - Method
        采用 Multi-head Attention 机制进行建模, 并利用 REINFORCE 强化学习算法对模型进行训练
  - 高性能计算领域
    - Method
      随着神经网络规模的不断扩大, 多 CPU 和多 GPU 混合训练是当前通用的一个方法, 将神经网络模型的不同计算功能部署在不同的计算设备上对训练速度有很大的影响
  - 微电网能量管理领域