

使用强化学习进行神经组合优化

Irwan Bello*, Hieu Pham*, Quoc V. Le, Mohammad Norouzi, Samy Bengio

2018 年 3 月 5 日

摘要

这篇 paper 提供一个框架, 使用神经网络和强化学习来解决 (tackle) 组合优化问题. 我们致力于 TSP 问题. 给定一个城市坐标 (city coordinates) 集合, 通过训练递归神经网络预测不同的城市序列 (city permutations) 的分布. 通过使用负的旅行长度作为奖赏信号 (reward signal), 我们使用梯度下降策略优化递归神经网络的参数. 我们对在一系列训练图上训练参数和单个测试图上训练参数进行比较. 尽管有计算费用, 但是没有太多的工程和启发式设计 (heuristic designing), 神经组合优化在高达 100 个节点的二维欧几里德图的结果达到接近最优. 应用到背包问题和另一个 NP-hard 问题, 相同的方法获得高达 200 物品的实例的最优解

1 引论

组合优化是计算机科学中的基础问题. 一个权威 (canonical) 例子是旅行商问题 (TSP), 问题描述如下: 给定一个图, 解决者需要搜索序列空间 (space of permutations) 找到拥有最短的边长度的点的最优序列 (tour length). TSP 和它的变种问题在计划 (plan), 制造业 (manufacturing), 遗传学 (genetics) 等拥有无数 (myriad) 应用

找到 TSP 问题的最优解是 NP-hard, 即使在一个两维的欧式空间 (点是 2D 的, 边的权重是欧式距离) 中. 在实践中, TSP 求解器依赖手工启发式 (handcrafted heuristics) 指导他们的搜索程序高效找到有竞争力的旅行路径. 即使这些启发式在 TSP 问题上工作很好, 一旦问题陈述稍微改变, 就需要修改 (be revised). 相反, 机器学习方法有在许多优化任务上可应用的潜力, 因为它基于训练数据发现每个任务的启发式. 所以比专门为一个任务优化的求解器需要更少的 hand-engineering.

尽管大多数成功的机器学习技术归于监督学习 (supervised learning) 家族 (学习从训练数据的输入到输出的映射), 监督学习对绝大多数组合优化问题并不是可应用的, 因为并不能获得最优的标签 (optimal labels). 但是你能使用一个验证者 (verifier) 比较一系列解决方案的质量, 提供一些奖赏回报 (reward feedbacks) 给一个学习算法. 所以, 我们遵循强化学习 (RL) 范式 (paradigm) 来解决组合优化问题. 我们经验性地证明 (empirically demonstrate), 即使使用标签数据来优化监督映射的优化方法, 也远逊色于一个探索不同的旅行路径和观察它们相应的奖赏的强化学习算法 (RL agent)

我们建议 (propose) 神经组合优化, 一个用强化学习算法和神经网络来解决组合优化问题的框架. 我们考虑两种基于梯度策略的方法. 第一种方法, 叫做 RL 预训练 (RL pre-training), 使用一个训练集合来优化一个递归神经网络 (RNN), 确定解决方案的随机策略的参数 (parameterizes a stochastic policy over solutions), 使用期望的奖赏作为目标 (using the

expected reward s objective). 在测试时, 策略是固定的 (fixed), 通过贪婪解码或采样进行推断 (one performs inference by greedy decoding or sampling). 第二种方法, 叫做积极搜索 (active search), 没有预训练。它从一个随机策略开始, 在一个测试实例上迭代地优化 RNN 参数。然后使用期望的奖赏目标, 同时跟踪在搜索期间采样得到的最好的解决方案。我们发现实践中联合 RL 预训练和积极搜索工作最好。

在有 100 个点的 2D 欧式图中, 神经组合优化显著胜过 (outperforms) 监督学习, 并且在允许更多的计算时间的情况下获得近似最优解的解。我们通过测试同样的方法在背包问题上的应用来阐述它的灵活性 (flexibility), 我们在高达 200 个物品的实例上获得最优解。这些结果给出了神经网络是如何作为一个通用工具解决组合优化问题的见解, 尤其是那些很难设计启发式的问题。

2 前导工作

旅行商问题是一个已经研究很好的组合优化问题, 许多确定的或近似的算法为解决欧式和非欧式的图被提出。Christofides(1976) 提出一个启发式的算法, 包括计算一个最小生成树 (minimum-spanning tree) 和一个最小权完美匹配 (minimum-weight perfect matching). 这个算法有多项式运行时间, 返回一个结果, 这个结果保证在 $1.5*$ 最优解之内。

TSP 最好的确定性动态规划算法的复杂度为 $\Theta(2^n n^2)$, 在规模较大的实例上这个算法是不可行的 (infeasible), 比如说 40 个点。然而, 感谢精心编写的描述如何以高效的方式遍历 (navigate) 可行解空间的启发式, TSP 求解器的最新进展 (state of the art TSP solvers), 可以解决上千个点的对称的 TSP 实例。Concorder(Applegate et al.,2006), 广泛地被接受为最好的确定性 TSP 求解器, 利用割平面算法 (Dantzig et al.,1954;Padberg & Rinaldi,1990;Applegate et al.,2003), 迭代解决 TSP 的线性规划松弛, 结合 (in conjunction with) 修剪 (prunes) 被证明 (provably) 不会包含最优解的搜索空间的分枝定界方法 (branch-and-bound approach)(英文原文:in conjunction with a branch-andbound approach that prunes parts of the search space that provably will not contain an optimal solution.). 类似地, Lin-Kernighan-Helsgaun 启发式 (Helsgaun,2000), 启发自 (inspired from)Lin-Kernighan heuristic(Lin & Kernighan,1973), 是对称 TSP 的近似搜索启发式的最新进展 (state of the art), 并且已被证明 (has been shown) 可以解决数百个节点的实例到最优性 (to optimality)。

更多的一般的求解器, 比如 Google's 车辆调度问题 (vehicle routing problem) 求解器 (Google,2016) 解决了 TSP 的超集 (superset of the TSP), 典型地依赖于 local search algorithms 和 metaheuristics 的组合。