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 7 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 tochastic 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 etal.,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。local search algorighms 将指定的本地移动操作符集合应用于候选解决方案 (apply a specified set of local move operators on candidate solutions), 基于一个手工设计 (hand-engineered) 的启发式比如 2-opt(Johnson,1990), 在搜索空间中从一个解决方案导航 (navigate) 到另一个解决方案. 应用一个元启发式 (metaheuristic) 来提出上坡动作并且逃离局部最优 (local optima). 一个 TSP 元启发式的流行的选择以及它的变种是指导式的本次搜索 (guiede local search)(Voudouris & Tsang,1999), 通过惩罚特别的不应该出现在一个好的方案中的方案特征来跳出局部最小 (local minimum).

应用存在的搜索启发式到新遇到的 (newly encountered) 问题或者甚至一个相似问题的

新实例的困难,是一个源自 (stem from) 没有免费午餐定理的著名挑战。因为所有的搜索算法当被平均到所有问题 (averaged over all problem) 时有相同的性能 (same performance), 你必须要适当地 (appropriately) 依赖于一个现有的 (prior over) 问题, 当选择一个搜索算法保证 (guarantee) 性能时. 这个挑战促进了 (foster) 提高优化系统运行的一般性水平的兴趣 (This challeng has fostered interest in raising the level of generality at which optimization systems operate), 并且是 hyper-heuristics 的潜在动力 (underlying motivation), 被定义为" 搜索方法或者学习机制 (learning mechanism), 来选择或生成启发式, 解决计算机搜索问题",hyper-heuristics 旨在 (aim to) 通过部分地抽象出选择启发式的知识密集型过程来比专门方法更容易地使用 (Hyper-heuristics aim to be easier to use than problem specific methods by partially abstracting away the knowledge intensive process of selecting heuristics given a combinatorial problem), 并且被证明成功地以优越 (superior) 的方式联合了许多任务中人工定义的启发式. 但是, hyper-heuristics 操作在启发式的搜索空间, 而不是解决方案的搜索空间, 所以仍然依赖于人工创造的启发式.

神经网络应用于组合优化有一段著名 (distinguished) 的历史, 研究的主体 (majority of research) 集中在 (focus on) 旅行商问题. 最早的提议 (proposal) 之一就是 Hopfield networks 在 TSP 上的使用。作者修改了网络的能量函数, 让它等价于 (equivalent to)TSP 目标, 并且使用拉格朗日乘子 (Lagrange multipliers) 来惩罚 (penalize) 问题约束的违反 (the violations of the problem's constraints). 该方法的一个局限 (limitation) 是对超参数和参数初始化敏感, 正如 (Wilson & Pawley,1988) 分析的那样。克服 (overcoming) 这一限制是该领域后续 (subsequent) 工作的核心, 尤其是 (Aiyer etal.1990;Gee,1993) 所做的工作。与 Hopfield 网络并行发展的是关于使用可变形的模板模型来解决 TSP 问题 (Parallel to the development of Hopfield networks is the work on using deformable template models to solve TSP.). 也许最突出的是发明弹性网络作为解决 TSP 的手段 (perhaps most prominent is the invention of Elastic Nets as a means to solve TSP)(Durbin,1987), 和自组织应用到 TSP 的映射 (and the application of Self Organizing Map to TSP)(Fort,1988;Angeniol et al.,1988;Kohonen,1990). 解决可变形模板模型的局限性对于这个领域接下来的工作至关重要 (Burke,1994;Favata & Walker,1991;Vakhutinsky & Golden,1995). 尽管这些神经网络具有许多吸引人 (appealing) 的特性, 他们仍然是有限的研究工作 (they are still limited as research work). 当他们被仔细检测 (carefully benchmakrked) 时, 他们与算法方法相比, 并没有取得令人满意的结果 (have not yielded satisfying results). 可能因为负面的结果, 自从世纪之交以来 (since the turn of the century), 这个研究方向很大程度上被忽视 (overlooked) 了

被 sequence-to-sequence learning 最近的进展所驱动 (Motivated by the recent advancements in sequence-to-sequence learning), 神经网络再一次成为不同领域优化的研究对象, 包括离散的 (discrete ones). 特别地, TSP 在介绍 Pointer Networks 时被重新提起 (revisited), 其中一个 recurrent network with non-parametric softmaxes 以受监督的方式训练以预测访问城市的顺序. 尽管在结构上有所改进, 但他们的模型使用由近似解算器 (appromixate solver) 给出的监督信号 (supervised signal) 进行训练

3 TSP 的神经网络框架

在这篇 paper 中我们集中于 2D Euclidean TSP. 给定一个输入图, 输入形式为在二维空间中的 n 个城市的集合 $s = \{X_i\}_{i=1}^n$, 其中每个 $x_i \in \mathbb{R}^2$, 我们关心找到点的序列 π, 称为游览 (termed a tour), 以最短的总长度游览每一个城市。我们定义一个序列 π 的长度是

$$L(\pi|s) = ||X_{\pi(n)} - X_{\pi(1)}||_2 + \sum_{i=1}^{n-1} ||X_{\pi(i)} - X_{\pi(i+1)}||_2 \tag{1}$$

其中 $||\cdot||_2$ 表示 (denote)l_2 范式

我们旨在学习一个随机 (stochastic) 策略 $p(\pi|s)$ 的参数, 即给定一个点集 s 的输入, 分配 (assign) 高的概率给短的游览, 低的概率给长的游览。我们的神经网络框架使用链式法则 (chain rule) 来因式分解 (factorize) 一个游览的概率如下:

$$p(\pi|s) = \prod_{i=1}^{n} p(\pi(i)|\pi(< i), s), \tag{2}$$

然后使用独立的 (individual)softmax 模块来代表公式(2)中的 RHS 方向的每一项

我们被之前的工作所启发 (inspired)(Sutskever et al.,2014), 利用基于链式法则的相同的因式分解, 其曾被用来解决 (address) 如机器翻译的序列到序列的问题. 我们可以用一个普通的 (vanilla) 序列到序列的模型来解决 TSP, 其中 TSP 的输出内容 (output vocabulary) 是 $\{1, 2, ..., n\}$