Genet: Automatic Curriculum Generation for Learning Adaptation in Networking

Zhengxu Xia, Yajie Zhou, Francis Y. Yan , Junchen Jiang

摘要

随着深度强化学习 (RL) 在网络方面展示其优势，它的缺陷也引起了公众的注意。在广泛的网络环境上进行训练会导致性能次优，而在狭窄的环境分布上进行训练则会导致泛化效果较差。这项工作提出了一种新的训练框架 Genet，用于学习更好的基于 rl 的网络适应算法。Genet 建立在课程学习的基础上，它已被证明有效地解决了其他 RL 应用程序中的类似问题。在高层次上，课程学习逐渐为训练提供更多“困难”的环境，而不是千篇一律的随机选择。然而，在网络环境中应用课程学习并非易事，因为网络环境的“难度”是未知的。我们的见解是利用传统的基于规则 (非 RL) 的基线: 如果当前的 RL 模型在网络环境中的表现比基于规则的基线差得多，那么在此环境中进一步训练它往往会带来实质性的改进。Genet 自动搜索这样的环境，并迭代地将它们推广到训练中。本文通过三个案例研究——自适应视频流、拥塞控制和负载平衡——证明了 Genet 产生的 RL 策略优于常规训练的 RL 策略和传统基线。

1 引言

深度强化学习 (Deep reinforcement learning, RL) 训练深度神经网络 (DNN) 作为决策逻辑 (policy)，非常适合于网络中的许多顺序决策问题，例如深度强化学习被广泛的应用在自适应码率（ABR），拥塞控制 (CC)，和负载平衡中（LB）, 但是强化学习存在着两个局限性，第一个局限性是强化学习的泛化性能不强，第二个局限性是训练在一个广泛的网络会导致较差的渐进性能，因此，如何更进一步的改进强化学习算法，使它更好的适用于网络领域是当下主要解决的问题。

2 相关工作

2.1 传统的基于机器学习的方法

先前的工作试图应用和定制来自 ML 的技术,[1]通过生成相对平滑的带宽轨迹来应用对抗学习，从而最大化 RLw.r.t 最优结果,[2][3]表明，可以通过将给定的 RL 策略违反预定义安全条件的训练环境纳入其中来提高 RL 的泛化,[4][5]在基于 RL 的系统的评估中纳入随机化，而 Fugu[6]通过在现场学习传输时间预测器实现了类似的目标。其他建议寻求在新环境中安全地部署给定的 RL 策略[7][8]。

2.2 传统的序列化环境方法

传统的确定奖励值最大的环境的方法主要有三种，第一种是固有属性。通过某些固有属性来量化环境的难度级别。例如，在拥塞控制中，具有更高带宽方差的网络跟踪直观上更加困难。然而，这种方法只能区分在精心挑选的属性上不同的环境，在复杂的环境下可能不够用 (例如，在训练中添加方差相似的带宽轨迹可能会有不同的效果)。第二种方法是基于规则的基线的性能。或者使用传统算法的测试性能来指示环境的难度。较低的性能可能意味着更困难的环境[9]。虽然这种方法可以区分任何两种环境，但它并没有暗示在训练过程中如何改进当前的 RL 模型。第三种方法是与性能差距最优来

作为衡量标准。为了解决方法二存在的问题，可以使用当前 RL 策略和最佳 RL 策略之间的性能差距来代替[1]。如果当前模型在一个环境中的表现比最优值差很多 (例如，通过使用 ground-truth 带宽作为带宽预测获得)，那么在这个环境中训练它的性能可能会提高。这种方法存在的问题是，最佳性能的计算可能非常昂贵，甚至是不可行的。这种方法也可能无法提高 RL 在固有硬环境中的性能 (例如，ABR 和 CC 中带宽的高度波动)。

3 本文方法

3.1 本文方法概述

在 Genet 框架中，主要包含三个过程：

1. Genet 通过两个 API 与现有的 RL 训练代码库交互训练向 RL 发出信号，使用给定的环境配置分布继续训练，并在指定次数的训练迭代后返回模型;Test 计算给定算法 (RL 模型或基线) 在给定数量的环境中获得的平均奖励。

2. 我们集成了 Genet 与 Pensieve ABR、Aurora CC 和 Park LB，它们使用不同的 RL 算法 (如 A3C、PPO) 和网络模拟器 (如包级、块级)。我们使用现有代码库中提供的功能实现上述两个 api。

3. 我们的三个用例已经实现了至少一个基于规则的基线 (例如，ABR 中的 MPC, CC 中的 Cubic)，这些基线在它们的模拟器中运行。Genet 框架如图 1所示：

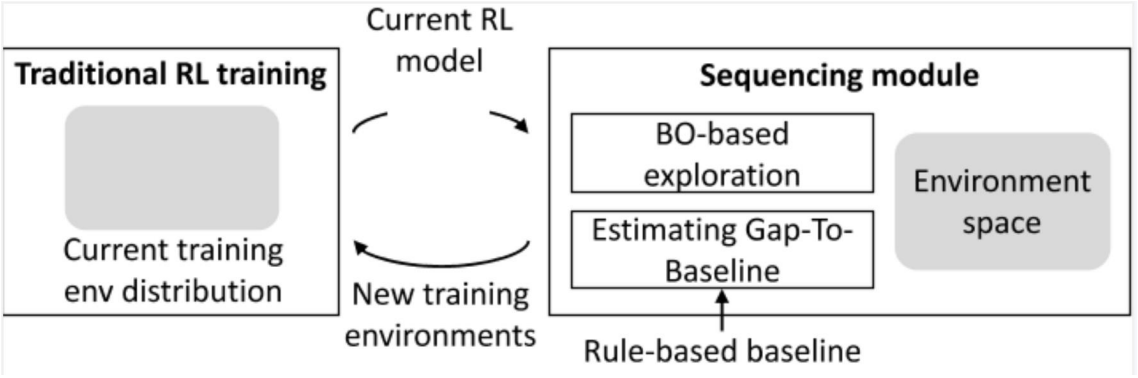


图 1: Genet 框架图

3.2 课程学习

传统强化学习算法训练每次迭代从固定的分布中对训练环境进行采样，课程学习改变训练环境分布，通过逐步增加训练环境的难度，从而在训练的过程中看到更多更容易改善的环境，我们称之为奖励环境。

3.3 奖励环境

传统的奖励环境是通过将强化学习与最优性能之间的差距作为奖励环境的值，但是此方法所需要花费的代价过大，除此之外，对于环境的提升也具有较大的难度，与以往强化学习采取的奖励环境设定不同，本文的奖励环境是通过与基线之间的性能差距作为标准，这种方式具有以下两个优点：

1. 当基于规则的基线在环境中的表现比 RL 策略好得多时，意味着 RL 模型在环境中训练时会学习“模仿”基线的已知规则，使其与基线持平。

2. 与基线的巨大差距表明当前 RL 模型有合理的改进空间。

4 复现细节

4.1 与已有开源代码对比

本次实验中引用了 https://github.com/GenetProject/Genet 链接里的源代码，包括其数据集以及逻辑实现，其 Genet 框架的伪代码逻辑如下：

Procedure 1 Genet training framework

Input : Ω :uniform configuration distribution (equal probability on each configuration),π^{rule}: rule-based policy.
Output: θ :final RL policy parameters
Function `Genet`(Ω, θ)**:**
  $\theta \leftarrow$ Random initial policy parameters
  $\Omega_{cur} \leftarrow \Omega$
  for *from 1 to* N_{iter} **do**
    BO.initialize(Ω) ▷ Initialize with full config spaceΩ
    for *from 1 to* $N_{boTrails}$ **do**
      $p \leftarrow$ BO.getNextChoice()
      $adv \leftarrow$ CalcBaselineGap$(p, \pi_\theta^{rl}, \pi^{rule})$
      BO.update(p, adv)
    end
    $p_{new} \leftarrow$ BO.getDecision()
    $\Omega_{cur} \leftarrow (1-w) * \Omega_{cur} + \omega * p_{new}$
    $\theta \leftarrow$UniformDomainRand$(\Omega_{cur}, \theta, N_{iters})$
  end
  return θ

4.2 实验环境搭建

实验需要在 Ubuntu 环境下运行，使用的是 python3.8 解释器，tensorflow 采用的 2.4.1 版本，并创建 anaconda 虚拟环境。

4.3 界面分析与使用说明

复现的论文无可视化界面

4.4 创新点

在复现论文的过程中，发现本篇论文有以下不足，首先是 Genet 在进行实验的时候假设存在合理的基线，若当基线不存在时，该如何完善，是否可以利用基于 ground truth 知识的最优解 (如未来带宽变化) 与当前 RL 模型之间的性能差距作为奖励网络环境选择的指导, 我认为这是一个可以尝试的方向。

5 实验结果分析

本部分对实验所得结果进行分析，详细对实验内容进行说明，实验结果进行描述并分析。

5.1 Genet 对于渐进性能的提升

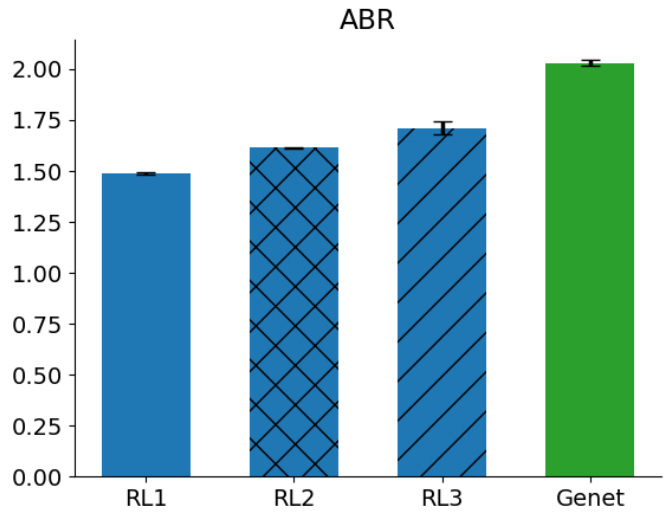


图 2: ABR

我们可以看到，与传统的 RL 训练方法相比，ABR 的 Genet 持续提高 6-25%。传统 rl 训练的三种策略之间没有明确的排名。因为 RL1 可以帮助训练更好地收敛，但只能看到目标分布的一小部分，而 RL3 可以看到整个分布，但不能训练出一个好的模型。相比之下，Genet 的表现优于它们，因为课程学习允许它从大型目标分布中更有效地学习.

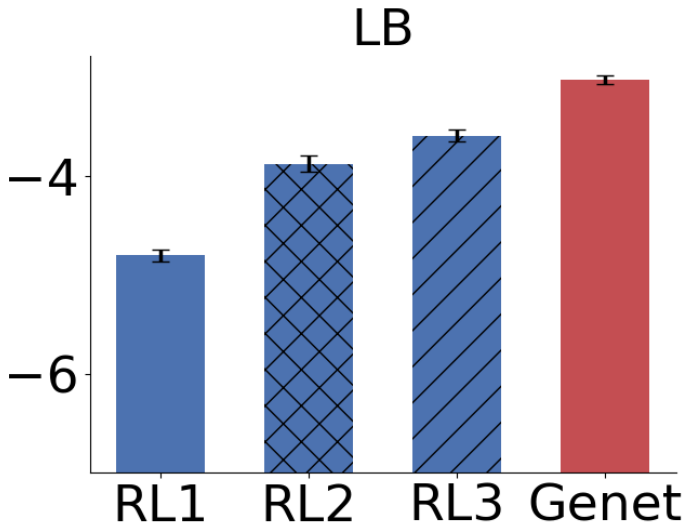


图 3: LB

我们可以看到，与传统的 RL 训练方法相比，LB 的 Genet 持续提高 15%。

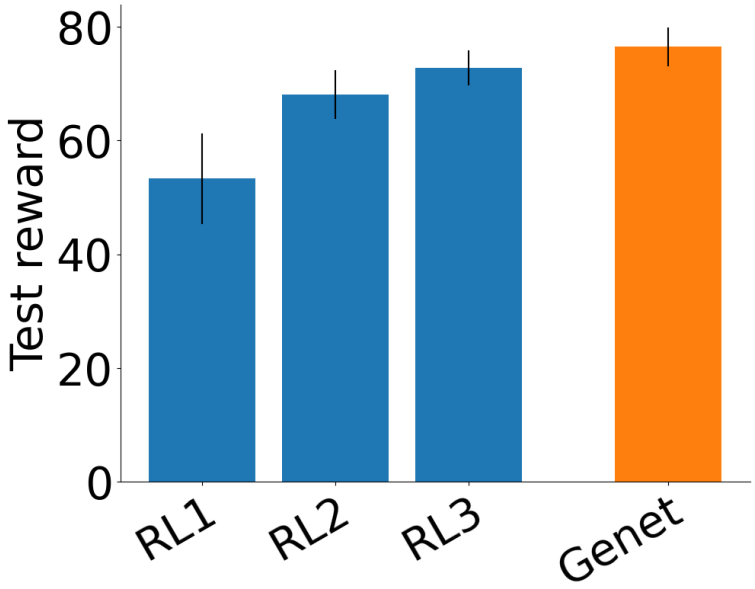
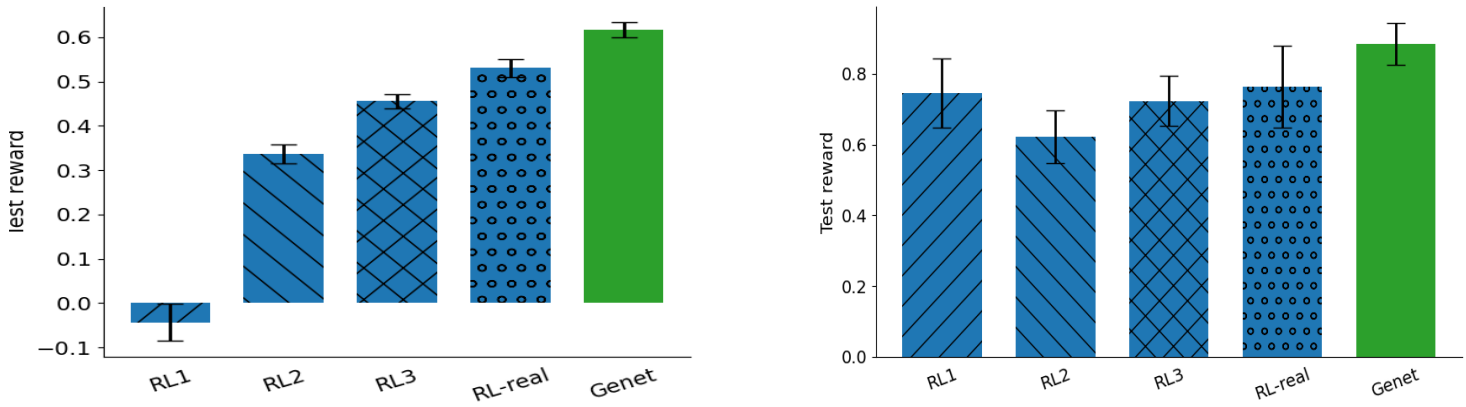


图 4: CC

我们可以看到，与传统的 RL 训练方法相比，CC 的 Genet 持续提高 4%−23%。

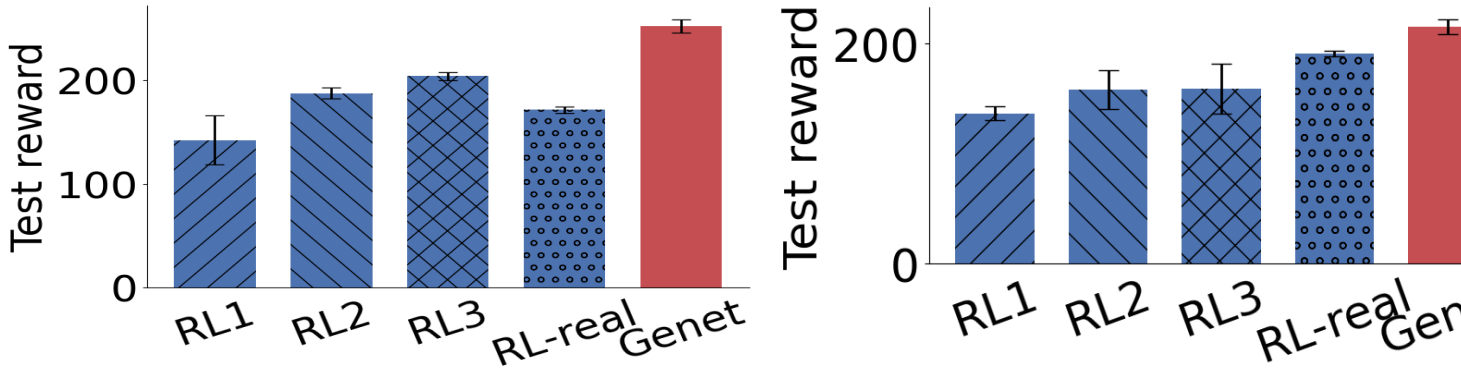
5.2 Genet 对于泛化性能的提升

我们采用 ABR 和 CC 的 RL 策略完全在合成环境进行训练，并且在追踪驱动的环境中测试它们的泛化性



(a) ABR(Norway)　　(b) ABR(FCC)

图 5: ABR 在合成环境训练在追踪环境中测试



(a) CC(Cellular)　　(b) CC(Ethernet)

图 6: CC 在合成环境训练在追踪环境中测试

从图中可以看出，我们以 ABR 和 CC 两个用例为例，Genet 对于泛化性能有着显著的提升。

6 总结与展望

Genet 假设存在一个合理的基于规则的基线，但是存在一些应用中存在没有基线的情况，除此之外，与基线差距小的有时候并不意味着 RL 模型在训练时有小的改进例如，Cubic 拥塞控制算法可能在偶尔出现随机丢包的高带宽链路上表现不佳，因为没有区分随机丢包和拥塞引起的丢包，导致它在可用带宽没有下降的情况下降低拥塞窗口大小。在这种情况下，即使 RL 模型与 Cubic 的基线差距很小，RL 模型仍然有提高性能的空间，但 Genet 可能不会选择优先考虑这样的环境，因此如何进一步改善强化学习，从而使之更适合于网络中的情况，是下一步要进行的主要工作。

参考文献

[1] GILAD T, JAY N, SHNAIDERMAN M, et al. Robustifying Network Protocols with Adversarial Examples[J]. hot topics in networks, 2019.

[2] ELIYAHU T, KAZAK Y, KATZ G, et al. Verifying learning-augmented systems[J]. acm international conference on applications, technologies, architectures, and protocols for computer communication, 2021.

[3] KAZAK Y, BARRETT C, KATZ G, et al. Verifying Deep-RL-Driven Systems[J]. acm international conference on applications, technologies, architectures, and protocols for computer communication, 2019.

[4] SCHAARSCHMIDT M. End-to-end deep reinforcement learning in computer systems[C]//. 2020.

[5] SCHAARSCHMIDT M, FRICKE K, YONEKI E. Wield: Systematic Reinforcement Learning With Progressive Randomization.[J]. arXiv: Learning, 2019.

[6] YAN F Y, AYERS H, ZHU C, et al. Learning in situ: a randomized experiment in video streaming[J]. networked systems design and implementation, 2019.

[7] ROTMAN N H, SCHAPIRA M, TAMAR A. Online Safety Assurance for Deep Reinforcement Learning [J]. arXiv: Learning, 2020.

[8] SHI J, SHA M, PENG X. Adapting Wireless Mesh Network Configuration from Simulation to Reality via Deep Learning based Domain Adaptation.[J]. networked systems design and implementation, 2021.

[9] WEINSHALL D, COHEN G, AMIR D. Curriculum Learning by Transfer Learning: Theory and Experiments with Deep Networks[C]//. 2022.