通过潜在克罗内克高斯过程预测学习曲线的连续 halving 方法

Jihao Andreas Lin[1, 2] **Nicolas Mayoraz**[1] **Steffen Rendle**[1] **Dima Kuzmin**[1] **Emil Praun**[1] **Berivan Isik**[1]

[1]Google

[2]University of Cambridge

Abstract 连续削减是一种流行的超参数优化算法，它将指数级更多的资源分配给有希望的候选者。然而，该算法通常依赖于中间性能值来做出资源分配决策，这可能导致过早地剪除那些最终可能成为最佳候选者的起步较慢者。我们研究了基于潜在克罗内克高斯过程的学习曲线预测是否能引导连续削减以克服这一限制。在一个涉及不同神经网络架构和点击率预测数据集的大规模实证研究中，我们将这种预测方法与基于当前性能值的标准方法进行了比较。我们的实验表明，尽管预测方法达到了具有竞争力的表现，但在将更多资源投入到标准方法上时，并没有达到帕累托最优，因为它需要完全观察到的学习曲线作为训练数据。然而，这一缺点可以通过利用现有的学习曲线数据来缓解。

1 介绍

超参数优化（HPO）是机器学习工作流中至关重要但计算成本高昂的一部分。为了解决这个问题，连续缩小法（SH）(Jamieson and Talwalkar, 2016; Karnin et al., 2013) 通过一系列层级高效地分配计算资源给许多超参数候选者。在每一层中，都会淘汰一部分候选者，通常是基于他们当前的中间表现。幸存下来的候选者将被提升到下一层，并获得指数级增加的资源预算。

然而，SH 通常假设中间性能值可以指示未来的性能值，这是一个显著的弱点。该启发式方法对于*慢启动者*的看似前景不大的中间性能值不起作用，因为 SH 会错误地丢弃它们（见 Figure 1，中部）。这引发了我们是否可以通过基于预测性能值做出更明智的剪枝决策来改进 SH，而不是依赖当前性能值的问题。

在这项工作中，我们通过基于潜在克罗内克高斯过程（LKGPs）(Lin et al., 2024, 2025) 的学习曲线预测来研究 SH，这是一种可扩展的概率方法，用于根据一组完全和部分观察到的学习曲线作为训练数据来预测未来的性能值。我们的假设是，使用这些预测来指导 SH 可能会导致更稳健且可靠的最优超参数配置的识别（参见 Figure 1）。

我们通过使用 1TB 的 Criteo 点击预测数据集 (Criteo, 2023) 进行大规模实证研究来检验我们的假设。通过模拟数千次 SH 运行，我们将基于当前性能值的标准方法与基于 LKGPs 的预测进行了比较。我们的实验表明，尽管基于 LKGPs 预测的 SH 实现了具有竞争力的表现，但这种方法在将更多资源投入到标准 SH 算法时并不是帕累托最优的，因为它依赖于额外的

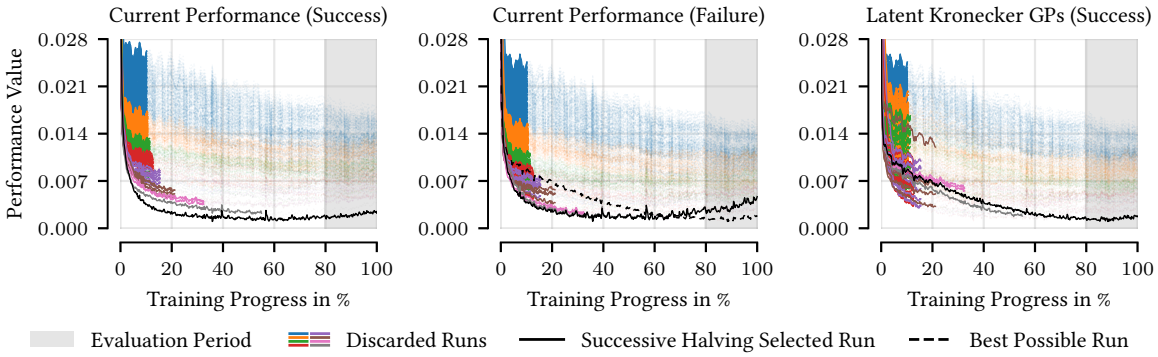

图 1: 基于当前性能值（左、中）和 LKGPs(Lin et al., 2024, 2025)（右）的预测进行连续缩小的可视化。如果中间的性能值能够预示未来的性能值，那么基于当前性能值的连续缩小将识别出最佳可能运行（左）。然而，有时最佳可能运行会被舍弃，因为其在相同阶段与其他候选相比中间的性能表现不佳（中）。使用 LKGPs 预测未来性能值可以识别出最佳可能运行，即使其中间的性能值并不乐观（右）。

完全观察到的学习曲线作为训练数据。尽管如此，我们相信可以通过利用现有的学习曲线数据来克服这一缺点，这为未来的研究开辟了新的途径。

2 背景

设共有 N 名候选人，通过其索引 $i = 1, \ldots, N$ 和相应的超参数 $x_i \in \mathcal{X}$ 来识别。每个候选人都是一种机器学习模型，它们是通过对所有候选人共享的固定数据序列进行迭代优化训练得到的。特别是，迭代训练程序为每个候选者 i 和时间步骤 $t = 1, \ldots, T$ 生成性能指标 $y_{i,t}$。

我们定义最佳候选人及其表现为 $i_* = \arg\min_i \text{perf}(i)$ 和 $y_* = \min_i \text{perf}(i)$，其中 $\text{perf}(i) = \frac{1}{\Delta} \sum_{t=T-\Delta}^{T} y_{i,t}$。换句话说，最佳候选人在最终窗口大小为 Δ 的平均性能指标中最小化。我们的目标是在减少观察到的绩效指标数量的同时识别最佳候选人 i_*，因为后者对应于消耗的资源。为了简化问题，我们假设所有时间步骤在所有候选人之间共享且均匀分布，并且生成任何性能指标 $y_{i,t}$ 所需的资源量相同。

连续减半. 连续丢弃（SH）最初被提出用于识别随机多臂老虎机设置中的最佳选项 (Karnin et al., 2013)。该算法后来应用于在非随机环境中执行超参数优化 (Jamieson and Talwalkar, 2016)。SH 的核心原则是向前景不佳的候选者分配较少资源，而向有希望的候选者分配指数级更多资源。特别是，分配计划在离散阶段运行台阶。在每一层级中，$1/\eta$ 的剩余候选者会被提升到下一层，并被提供更多的资源（参见 Algorithm 1）。通常，关于候选者晋升的决策是基于当前性能值做出的。然而，当前的性能值并不总是能够预示在提供更多资源后未来的性能值。例如，一个起步慢但稳步提高的候选人最终可能会超越那些快速收敛的其他候选人。因此，准确预测未来表现的能力可能潜在地提升 SH 的表现。

潜 **Kronecker** 高斯过程. 最近，Lin et al. (2024, 2025) 提出了潜在克罗内克高斯过程（LKGPs），这是一种可扩展的概率回归方法，可以通过考虑函数 $f : \mathcal{X} \times \mathcal{T} \to \mathcal{Y}$，将超参数 $\mathcal{X}$ 和时间步骤 $\mathcal{T}$ 的笛卡尔积空间映射到性能指标，从而用于预测学习曲线。特别是，LKGPs 模型使用乘

Algorithm 1 连续加倍减少法

Require: 候选人初始数量 N，最终候选人数量 F，减少率 η

$S \leftarrow \lceil \log_\eta(N/F) \rceil$ ▷ Calculate total number of rungs

for $s = 1, \ldots, S$ **do**

$R \leftarrow (\eta^s - 1)/(\eta^S - 1) \times 100\%$ ▷ Calculate relative resource budget for the current rung

对剩余候选人进行训练，直到每个候选人的最大资源预算的 $R\%$

根据当前或预测的表现值对剩余候选人进行排名

将表现最出色的前 $N/(\eta^{s-1})$ 候选人提升到下一个级别

end for

return 表现最出色的前 F 候选人

积核 f 来建模 $k((x,t),(x',t')) = k_X(x,x')k_T(t,t')$，并基于与观察到的训练数据的相关性进行非参数预测，而不是依赖特定的参数函数 (Domhan et al., 2015) 或合适的合成数据 (Adriaensen et al., 2023)。

3 贡献

本文中，我们将基于当前绩效值的 SH 下游表现与通过 LKGPs 预测学习曲线的晋升决策进行比较。为此，我们首先通过对神经网络的训练来创建学习曲线数据，然后使用这些数据模拟 SH。

创建学习曲线数据. 我们在固定的数据序列上训练神经网络，该数据序列包含在线广告的时间顺序点击反馈数据。特别是，我们使用了 Criteo 1TB 数据集 (Criteo, 2023)，该数据集中包含了 Criteo 在 24 天内的部分流量。每个数据点表示某个展示广告是否被点击。

我们考虑因子分解机（FMs）(Rendle, 2010)、深度与交叉网络（CNs）(Wang et al., 2017, 2021) 以及专家混合模型（MoEs）(Shazeer et al., 2017)。对于每种架构，我们通过优化参数如学习率或权重衰减的随机搜索来探索配置，并通过网格搜索探索架构参数，例如隐藏层数。

由于在线流量中的时间序列数据包含强烈的时间依赖趋势，我们考虑使用参考模型的差异而不是原始性能值。通过训练特定模型对训练数据进行两次遍历，并将第二次遍历用作参考模型来构建参考模型。该参考模型在所有架构和配置之间共享。

模拟连续减半. 为了获得具有统计显著性的结果，我们使用来自每个架构的 512 条学习曲线中的随机子集（256 条）来模拟 SH。我们将减小因子 η 设置为 2，并将 SH 延迟到训练运行完成 10% 后，以减少不稳定的决策。此外，我们将最终候选者数量设置为 $F \in \{1, 2, 4, 8, 16, 32, 64\}$，这会导致不同的性能与计算权衡。

我们预测候选排名要么基于当前的表现值，要么使用 LKGPs。对于前者，我们取完整运行最后 20%的平均值以提高鲁棒性。对于 LKGPs，我们使用每个维度的长度尺度、共享核幅度和同方差噪声参数的平方指数核。这些参数通过使用学习率为 0.1 且迭代次数为 100 次的 Adam 最大化边缘似然进行优化。我们将输入归一化至 $[0, 1]$，并通过减去最终时间步长上的均值并除以所有时间步长上的标准差来标准化输出。为了对候选者进行排名，我们使用 64

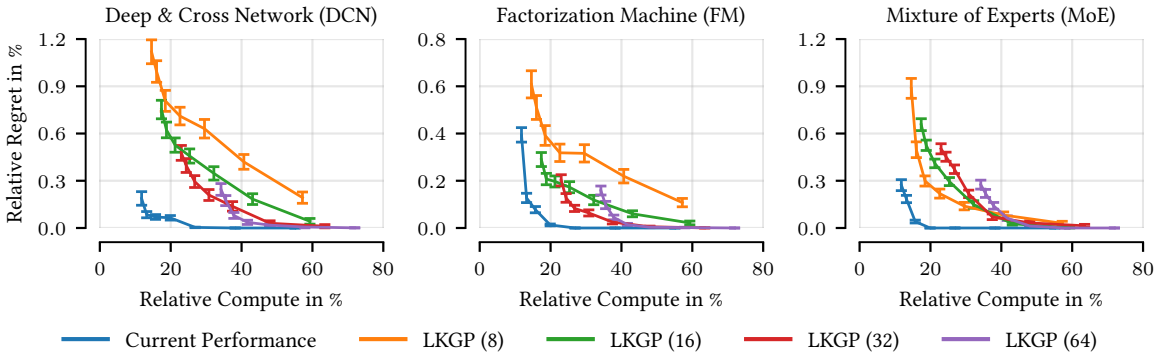

图 2: 基于当前性能值预测的连续减半（蓝色）与使用不同数量的学习曲线作为训练数据的 LKGPs（橙色、绿色、红色、紫色）在三种神经网络架构（DCN、FM、MoE）上的比较。如预期的那样，增加使用的计算量通常会提高性能。尽管 LKGPs 达到了具有竞争力的表现，但由于需要训练数据，它们并非帕累托最优。

个后验样本预测每个候选者的 $\text{perf}(i)$ 的均值 μ_i 和方差 σ_i^2。通过计算并在成对比较中排序每个候选者的预期胜场数来获得排名，

$$\mathbb{E}\,\text{wins}(i) = \frac{1}{n-1}\sum_{j\neq i}\mathbb{P}[\text{perf}(i) > \text{perf}(j)] = \frac{1}{n-1}\sum_{j\neq i}\Phi\Big(\frac{\mu_i - \mu_j}{\sqrt{\sigma_i^2 + \sigma_j^2}}\Big), \tag{1}$$

其中 Φ 是标准正态分布的累积分布函数。由于 LKGPs 使用完全观测的学习曲线作为训练数据，我们提供 $C \in \{8, 16, 32, 64\}$ 条曲线，这些曲线是从 256 条曲线子集中均匀随机选择的，并对剩余的曲线执行 SH。

性能评估. 为了比较基于当前性能值的预测与使用 LKGPs 进行预测的表现，我们考虑相对于参考模型的相对遗憾和所使用的计算量。相对遗憾定义为绝对遗憾除以参考模型的性能，其中绝对遗憾通过从最佳可能性能中减去 SH 提出的最优候选者的性能来计算。

计算相对量由将观察到的性能值数量除以所有性能值的数量 $N \times T$ 计算得出，其中每个性能值代表一个计算单位。特别是，观察到的性能值数量包括所有候选人的初始 10% 宽限期以及 SH 期间的后续观察结果。对于 LKGPs，它们还包括作为训练数据使用的完整观察学习曲线。

结果. Figure 2 显示了相对计算量使用的函数在 100 次试验中相对遗憾的平均值和标准误差。后者的变化是由于不同的 SH 计划，这些计划由最终候选人的数量 F 参数化。对于 LKGPs，使用的计算量还受用作训练数据的完全观察曲线的数量 C 的影响。

如预期的那样，通过增加 F 来使用更多计算通常会导致更低的遗憾。有趣的是，增加 F 似乎对 SH 当前性能值特别有效。在这种情况下，SH 在所有 100 次试验中都持续实现了零遗憾，对于 $F >= 32$ 也是如此。基于 LKGPs 进行预测时，SH 的表现也会随着 F 或 C 的增加而改善。然而，需要完全观察到的曲线作为训练数据显著增加了计算量的使用，因此带有 LKGPs 的 SH 不是帕累托最优。在所有实验中，相同的计算量可以通过将计算投资于增加 F 来实现更低的遗憾，对于当前性能下的 SH 也是如此。

4 结论

我们的研究表明，基于 LKGPs 预测的 SH 与投入更多计算资源以当前表现为基础进行预测的 SH 相比，并非帕累托最优。然而，这种情况主要是因为 LKGPs 需要完整的训练曲线作为训练数据，这显著增加了使用的计算量。这一缺点可以通过利用现有的学习曲线来抵消，这对于相同的神经网络架构而言是直接可行的，并且激励未来的研究工作探索在不同架构间进行转移的方法。此外，我们对 LKGPs 的训练数据进行了均匀随机的选择，这很可能不是最优的方式。进一步地，通过考虑专业内核和更复杂的特征工程（例如输入变形或嵌入）来提高 LKGPs 预测的质量是可能的。

References

Adriaensen, S., H. Rakotoarison, S. Müller, and F. Hutter (2023). "Efficient Bayesian Learning Curve Extrapolation using Prior-Data Fitted Networks". In: *Advances in Neural Information Processing Systems*.

Criteo (2023). *Criteo pCTR data - 1TB*. `https://ailab.criteo.com/download-criteo-1tb-click-logs-dataset/`.

Domhan, T., J. T. Springenberg, and F. Hutter (2015). "Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves". In: *International Joint Conference on Artificial Intelligence*.

Jamieson, K. and A. Talwalkar (2016). "Non-stochastic Best Arm Identification and Hyperparameter Optimization". In: *International Conference on Artificial Intelligence and Statistics*.

Karnin, Z., T. Koren, and O. Somekh (2013). "Almost Optimal Exploration in Multi-Armed Bandits". In: *International Conference on Machine Learning*.

Lin, J. A., S. Ament, M. Balandat, and E. Bakshy (2024). "Scaling Gaussian Processes for Learning Curve Prediction via Latent Kronecker Structure". In: *NeurIPS Bayesian Decision-making and Uncertainty Workshop*.

Lin, J. A., S. Ament, M. Balandat, D. Eriksson, J. M. Hernández-Lobato, and E. Bakshy (2025). "Scalable Gaussian Processes with Latent Kronecker Structure". In: *International Conference on Machine Learning*.

Rendle, S. (2010). "Factorization machines". In: *International Conference on Data Mining*.

Shazeer, N., A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean (2017). "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer". In: *International Conference on Learning Representations*.

Wang, R., B. Fu, G. Fu, and M. Wang (2017). "Deep & Cross Network for Ad Click Predictions". In: *AdKDD and TargetAd Workshop*.

Wang, R., R. Shivanna, D. Cheng, S. Jain, D. Lin, L. Hong, and E. Chi (2021). "DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems". In: *The Web Conference*.