arXiv:2305.01667v1 [cs.LG] 2 May 2023

使用 GP-NAS 堆叠集成模型预测 NAS 多任务

Ke Zhang

kezzhang@gmail.com

摘要

利用小样本训练准确预测架构性能是一项重要但并不容易的任务。如何分析和训练数据集以克服过拟合，是我们需要解决的核心问题。同时，如果存在多任务问题，我们还应考虑能否利用任务间的相关性，并尽可能快速地进行估计。在本赛道中，*Super Network* 基于 *ViT-Base* 构建了一个搜索空间，其中包含深度、头数、多层感知器比例和嵌入维度。我们首先根据对这一问题的理解对数据进行预处理，以降低问题的复杂度和过拟合的可能性。随后，我们尝试了不同类型的模型以及不同的组合方式。最终，我们选择使用带交叉验证的 *GP-NAS* 堆叠集成模型。我们的堆叠模型在 *CVPR 2022* 第二赛道挑战赛中排名第一。

1. 引言

由于数据源有限、高精度大规模数据集的采集成本高昂、人工标注数据困难、计算能力受限以及快速估计的需求，在计算机视觉、金融、电子商务、医学等诸多不同领域中，基于小规模数据进行训练并实现准确预测变得越来越重要。与此同时，随着我们能够从越来越多的不同来源收集数据，我们还应考虑如何在不牺牲精度的情况下，同时处理多项任务，或充分利用它们之间的协作关系。

神经架构搜索（NAS） [5] [13] [15] 已成功用于设计深度神经网络架构。在许多计算机视觉任务中，其表现优于人工设计模型。然而，使用神经架构搜索算法时仍有若干问题亟待解决。例如，同一网络独立训练时性能不一致；在小样本训练和预测中表现不佳。近期，研究者提出了 GP-NAS（基于高斯过程的神经架构搜索） [11], ，该方法通过核函数和均值函数对相关性进行建模。贝叶斯定理和信息论在该模型中得到了深入应用，从理论上确保了其性能良好，尤其是在样本量较小时。

然而，我们发现，将数据简单、直接地输入 GP-NAS 并不能取得相对理想的结果。我们认为原因有几个。首先，这类模型对输入较为敏感，需要对数据进行准确的预处理。其次，仅运行该模型无法充分挖掘数据中的大部分信息，尤其是在数据集中存在非线性或复杂特征时。

在本文中，我们深入分析了这一问题，并尝试了不同类型的模型，最终选择通过 GP-NAS 堆叠多个集成模型，在 CVPR 2022 Track2 Challenge 中取得了相当不错的性能。

2. 我们如何处理这个问题

2.1. 原始数据

在这项挑战中，Super Network 基于 ViT-Base 为 8 个任务构建了一个搜索空间。该搜索空间涉及深度数量、注意力头数量、MLP 比率和嵌入维度。对于每个任务，训练数据包含 500 个样本，其输入为模型结构，标签为结构的相对排名；测试数据则包含 99500 个样本。可以发现，训练样本量相比测试数据较小，因此需要具备基于小样本数据集进行准确拟合的能力。显然，这正是本赛道挑战的目标：在训练数据很少甚至没有训练数据的情况下，准确预测任意架构的性能。

2.2. 数据预处理

对于这个问题，最直接、最简单的方法是使用默认结构值训练所有网络。这种方法的问题在于存在奇异性（嵌入维度不包含信息），且未能利用背景信息。因此，我们按以下步骤对数据进行预处理：

- 将深度编码（j,k,l）转换为整数。这里我们采用序数编码而非独热编码，因为我们认为深度具有一定的单调相关性

和可预测性，即深度越深，信息量越多。实验也证实了我们的想法。 [12]
序数编码：

$$(j,k,l) \Rightarrow (1,2,3) \tag{1}$$

独热编码：

$$(j,k,l) \Rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

•去除嵌入维度特征参数，因为它不包含任何信息，这可以避免奇异性，并将问题维度减少 1/4。另一方面，当子网络的实际深度小于 12 时，其编码末尾会用 0 填充；这里我们将 0 改为 2，因为我们假设 2 代表中性信息。经过此转换后，深度编码特征与最后四个编码特征的平均相关性为 0.015，而不是 0.75。最后，我们将数据从 (1,2,3) 归一化到 (-1,0,1)。

• 使用反 sigmoid 函数作为激活函数[2]。众所周知，排序数据服从均匀分布。通过反 sigmoid 函数，我们将均匀分布转换为高斯分布，这对大多数模型来说是最佳选择。得到预测后，我们使用 Sigmoid函数将输出转换回均匀分布，作为排名输出，然后将其四舍五入到最接近的整数。

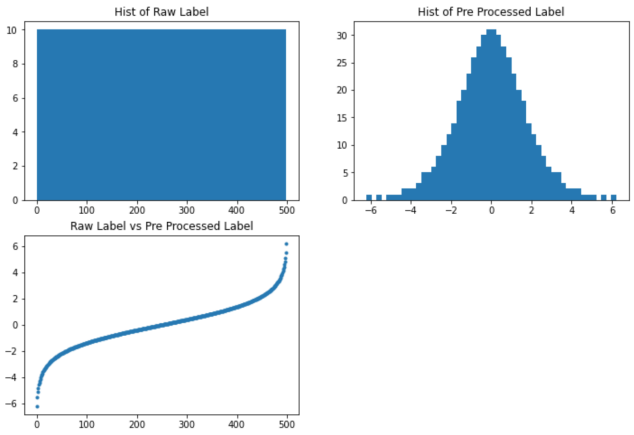


图 1. 使用反 Sigmoid 函数作为激活函数，将排名标签（均匀分布）转换为新标签（高斯分布）

上图展示了原始标签的直方图、预处理后标签的直方图，以及原始标签与预处理后标签之间的关系。

$$z = \frac{\log(y+1)}{n-y}, y \in [0, n-1] \tag{3}$$

公式（3）即我们上文提到的反 sigmoid 函数。

• 我们还尝试了许多其他方法，例如使用不同的激活函数、添加高斯噪声等，但这些方法都未能在不同任务上带来明显提升。

2.3. 模型构建

对原始数据集进行预处理后，我们开始构建训练与测试框架。我们尝试了许多模型及不同的组合方式（详情见下一节），最终选择对集成模型进行堆叠 [14] [8] [7]，并使用 GP-NAS 作为最终估计器 [4]。

要得到最终结果，需要经过几个步骤。首先，我们将数据集划分为训练数据集和测试数据集。使用完整的训练数据集训练子模型。然后，将训练数据集划分为 K 折，使用其中 K-1 折从每个子模型获取预测结果 [10]。其次，对于每个 K 折，将每个子模型的预测结果作为训练数据集，并将剩余的 1 折作为验证数据，用于训练 GP-NAS；重复 K 次。随后，我们可以基于 GP-NAS 进行预测，得到第二层预测。

最后，我们对数据集进行后处理：使用 sigmoid 函数作为激活函数，将第二层预测转换为均匀分布，并将其四舍五入到最近的整数，作为排名预测。

$$s = \frac{n-1}{1+\exp(-z)}, z \in [-\infty, +\infty] \tag{4}$$

公式（4）是我们使用的Sigmoid函数。

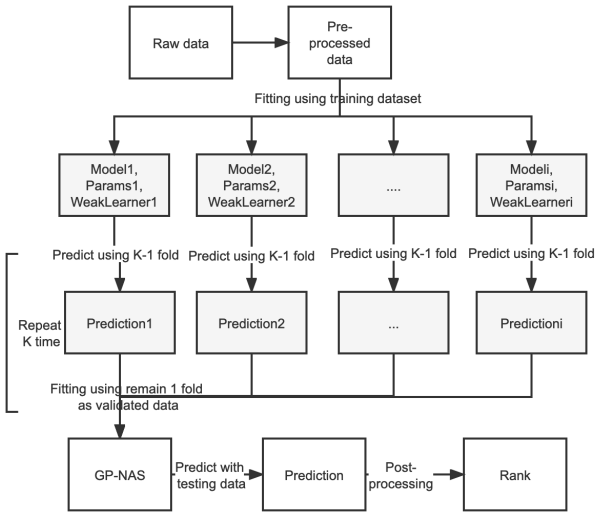


图 2. 端到端训练和预测过程

到目前为止，我们基本构建好了模型框架。然而，这并不意味着我们已经通过该框架取得了良好的结果。起初，对于不同的任务，我们使用相同的参数，例如学习率、损失函数和模型深度，但输出结果不如最终结果理想。这种现象完全合理，因为不同的数据集具有不同的信噪比和属性。例如，信噪比较高的任务需要更大的学习率；非线性程度更高、结构更复杂的任务则需要更大的最大深度。总之，我们应根据具体问题找到不同的方差与偏差权衡，以实现良好预测并避免过拟合。

2.4. 技术讨论与改进

上一节介绍了我们用于处理这项任务的框架。在此之前，我们尝试了不同的模型和组合方式。现在，让我们看看尝试过哪些方法，以及最终模型是如何得出的。

- 首先，我们获得了基线代码。仅使用 GP-NAS 模型 [11]，最终得分约为 0.67（我们计算的所有得分均为肯德尔秩相关系数）。

- 此外，我们还考虑了其他机器学习模型，例如线性回归、随机森林模型、核岭回归、树模型、轻量级梯度提升、 [9], XGBoost [1],CatBoost 梯度提升 [3] 等。我们发现，梯度提升或其他基于提升法的算法 [6]平均表现最佳。最初尝试时，得分约为 0.73；调整参数后，得分达到 0.78。这项实验使我们将子模型候选范围缩小到基于提升法的算法。

  基于提升法的算法是一种基于树的集成模型，由多个弱学习器组成，这些弱学习器通常是决策树。单独使用时，决策树通常较弱，往往只能处理是非类问题。对于基于提升法的算法，每棵树的目标并非在随机分配的数据子集上最大化自身的效用函数，而是拟合前一棵树的残差；这种方法在处理非线性和复杂问题时表现相对出色。

  大量实验表明，与其他方法相比，提升法算法具有以下优势：通常能提供难以超越的预测准确度；灵活性很高，可以针对不同的损失函数进行优化，并提供多种超参数调优选项。最重要的是，无需预处理数据，对数据分布和缺失值不敏感。

- 由于我们的问题属于多任务问题，我们能否同时训练不同的任务？我们考虑使用基于 MultiRMSE 的多变量梯度提升法 [3]，该方法来自软件包 catboost。

$$MultiRMSE = \sqrt{\frac{\sum_{i=1}^{N}\sum_{d=1}^{dim}(a_{i,d} - t_{i,d})^2 w_i}{\sum_{i=1}^{N} w_i}} \tag{5}$$

  然而，结果甚至比单变量梯度提升方法更差。原因可能如下：不同任务差异很大，应该采用不同的参数；不同任务之间没有明显的领先或滞后关系；我们无法在这种多变量梯度提升算法中有效降低噪声。

- 由于训练样本量很小，为避免过拟合，我们尝试对模型进行集成。我们选择梯度提升回归树 [6],直方图梯度提升 [9],CATGB [3],XGBoost[1], LightGBM [9] 作为子模型。最直接的方法是对所有结果取平均，这使结果提升至约 0.79。

- 在朴素平均之后，我们考虑使用最终估计器对梯度提升模型组进行堆叠。经过多次实验和比较，我们选择 GP-NAS 作为最终回归器。堆叠是一种通用方法，通过训练最终估计器来组合各个单独学习器。堆叠可以将每个单独估计器的输出作为最终估计器的输入，从而发挥各自的优势。为避免过拟合，我们仅在子模型训练中使用完整的训练数据集，而通过交叉验证训练最终模型。我们采用 K−1 折来预测子模型，并将剩余的 1 折作为验证数据；对每个 K 重复此过程。由于 scikit-learn 已提供这种回归器， [14],我们对 GP-NAS [11]进行了修改，使其符合 scikit-learn API。现在我们得到的结果约为 0.792。

- 到目前为止，我们的堆叠模型对不同任务仍使用相同的参数，例如损失函数、学习率和模型深度。因此，我们考虑为不同任务选择不同的损失函数、学习率和模型深度，因为我们假设不同的数据集和任务具有不同的信噪比和不同的特征。同时，我们增加了两个梯度提升回归树（ GBRT）和 CATGB 子模型，并为它们采用不同的损失函数（Huber 或均方误差），这也带来了改进。本轮结果约为 0.798。

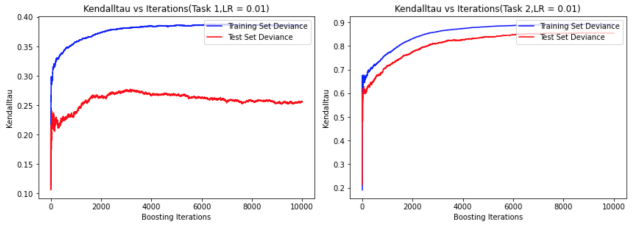


图3. 任务1和任务2在不同提升迭代次数下的肯德尔秩相关系数

图。3 显示，即使使用相同的模型，不同任务的信噪比和特征也完全不同。在此示例中，我们对任务1和任务2使用相同的参数和模型，但呈现出截然不同的现象。

- 我们还修改并调优了最终估计器神经架构搜索（GP-NAS），例如岭参数和估计器先验。我们没有使用单位矩阵，而是选择 inv(X.T*X) 作为先验协方差矩阵，如公式（9）所示。最终得分为：排行榜A 上 0.7991，排行榜 B 上 0.79849。

$$Mean(\beta|X,Y) = (X^TX)^{-1}X^TY \tag{6}$$

$$Var(\beta|X,Y) = \sigma^2(X^TX)^{-1} \tag{7}$$

公式 (6)公式 (7) 是估计器的均值先验和方差先验。

模型	肯德尔秩相关系数
GP-NAS 基线	0.67
梯度提升	0.78
多变量梯度提升	0.787
5个集成算法平均	0.79
使用相同任务参数的5个算法堆叠	0.792
使用不同任务参数的7个算法堆叠	0.798
将逆协方差用作先验的7个算法堆叠	0.7991

表 1. 结果比较

表。1 展示我们尝试的模型及其肯德尔τ得分。

3. 结论

本文介绍了原始数据的预处理方法、尝试过的模型，以及最终版本模型框架的详细信息。结果表明，将多个集成模型与 GP-NAS 堆叠使用，能够取得相当不错的效果。在拟合过程中，我们考虑了交叉验证，因此模型在测试数据上的预测结果也相当理想。受时间所限，还有许多想法尚未深入探讨，例如改进多变量梯度提升、堆叠多变量集成模型，以及在 GP-NAS 中采用不同的核函数和先验，这些都可以在未来进一步研究。

参考文献

[1] Tianqi Chen 和 Carlos Guestrin。Xgboost：一种可扩展的树提升系统。载于第22届ACM SIGKDD国际知识发现与数据挖掘会议论文集，第785–794页，2016年。3[2] Allen Chieng Hoon Choong 和 Nung Kion Lee。使用序数编码与独热编码方法评估卷积神经网络对DNA序列的建模。载于2017年计算机与无人机应用国际会议（IConDA），第60–65页。IEEE，2017年。2[3] Anna Veronika Dorogush、Vasily Ershov 和 Andrey Gulin。CatBoost：支持类别特征的梯度提升。arXiv预印本 arXiv:1810.11363，2018。3[4] Saso Dˇzeroski 和 Bernard ˇZenko。通过堆叠组合分类器是否优于选择最佳分类器？Machine Learning，54(3)：255–273，2004年。2[5] Thomas Elsken、Jan Hendrik Metzen 和 Frank Hutter。神经架构搜索：综述。The Journal of Machine Learning Research，20(1)：1997–2017，2019年。1[6] Jerome H Friedman。贪心函数逼近：梯度提升机器。Annals of Statistics，第1189–1232页，2001年。3[7] Benyamin Ghojogh 和 Mark Crowley。过拟合、交叉验证、正则化、装袋法和提升法背后的理论：教程。arXiv预印本 arXiv:1905 12787，2019年 2 。 。[8] Trevor Hastie、Robert Tibshirani、Jerome H Friedman 和 Jerome H Friedman。The Elements of Statistical Learning：数据挖掘、推断与预测，第2卷。Springer，2009年。2[9] Guolin Ke、Qi Meng、Thomas Finley、Taifeng Wang、Wei Chen、Weidong Ma、Qiwei Ye 和 Tie-Yan Liu。LightGBM：一种高效的梯度提升决策树。Advances in Neural Information Processing Systems，30，2017年。3[10] Ron Kohavi 等。关于用于准确率估计和模型选择的交叉验证与自助法的研究。载于IJCAI，第14卷，第1137–1145页。加拿大蒙特利尔，1995年。2[11] Zhihang Li、Teng Xi、Jiankang Deng、Gang Zhang、Shengzhao Wen 和 Ran He。GP-NAS：基于高斯过程的神经架构搜索。载于IEEE/CVF计算机视觉与模式识别会议论文集，第11933–11942页，2020年。1, 3[12] Kedar Potdar、Taher S Pardawala 和 Chinmay D Pai。神经网络分类器的类别变量编码技术比较研究。International Journal of Computer Applications，175(4)：7–9，2017年。2[13] Esteban Real、Alok Aggarwal、Yanping Huang 和 Quoc V Le。用于图像分类器架构搜索的正则化进化。载于AAAI人工智能会议论文集，第33卷，第4780–4789页，2019年。1[14] David H Wolpert。堆叠泛化。Neural Networks，5(2)：241–259，1992年。2, 3

[15] Barret Zoph、Vijay Vasudevan、Jonathon Shlens 和 Quoc V Le。学习用于可扩展图像识别的可迁移架构。载于 *IEEE/CVF*计算机视觉与模式识别会议，第8697 8710页，2018l– 。