

Lab 8 - 自动机器学习系统练习

实验报告

方书成

2021 年 5 月 27 日

1 实验环境

- CPU: AMD Ryzen 7 4800H 8 核 16 线程
- GPU: NVIDIA GeForce RTX 2060
- OS: Windows 10 Insider Preview Build 21387
- PyTorch 1.7.1 CUDA 10.2 NNI 2.2

2 实验过程

2.1 原始代码

超参：

超参名称	值	超参名称	值
initial_lr	0.1	weight_decay	5e-4
ending_lr	0	cutout	0
batch_size	128	epochs	300
optimizer	sgd	momentum	0.9
grad_clip	0	model	resnet18

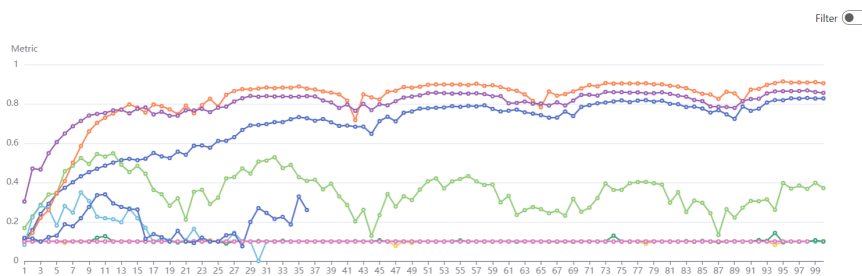
模型准确率：0.849100

2.2 NNI 自动调参

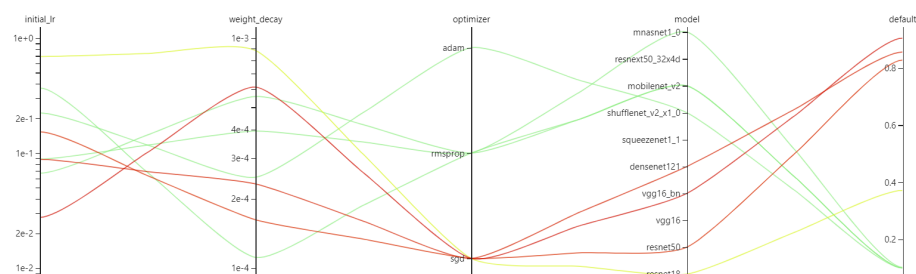
起初我希望使用 Annotation 来进行这一部分的实验，但是似乎新版的 NNI 对这一部分的功能还未支持，v2 configuration 的处理过程中直接将 config.yml 中的 useAnnotation 属性忽略了。所以我只能使用传统的方法。

在使用传统方法成功运行了之后，我发现网页未能正确显示中间结果，也不能接收到最终结果。在 GitHub 上检索一下，发现只要将 pytorch 的 DataLoader 的 num_workers 设为 0，即不使用并行模式，就可以正确的产生中间和最终结果。但是在这种情况下 CPU 的单核性能成为瓶颈，nvidia-smi 显示的显卡占用率只有 30%-40%，导致一整晚只能完成 3-4 个 trial。为了解决这个问题，我去翻看了 NNI 和 PyTorch 的源码，并进行了实验，然后发现在使用并行模式时在顶级代码中打开的文件会在 DataLoader 被 enumerate 时被清空。再次查询后得知，这是由于 Python 的 multiprocessing 库在 Windows 下使用 spawn 模式产生子线程，在这种模式下顶级代码中的文件句柄会被丢弃，在 Linux 下使用 fork 模式时则不会产生这个问题。在 NNI 中 Dispatcher 和训练程序使用文件进行通讯，而这些文件在 runtime/platform/local.py 中在顶级代码被打开，继而导致了每个 epoch 这些文件都会被清空。所幸有一个简单的解决办法，只需要把这些文件的打开模式从 w（写入）改为 a（追加），在 NNI 的运行模式下，这些文件不会被复用，每个 trial 都是独立的，故不会产生冲突，在一个 trial 新的线程中也不会清空这些文件了。这些改动我已经提交 Pull Request 至 GitHub 并被接收。

以下是运行结果。



最佳 trial：模型准确率：0.904700，比原始模型高了 6%。



超参名称	值	超参名称	值
initial_lr	0.02757	weight_decay	0.00061
ending_lr	0	cutout	0
batch_size	128	epochs	100
optimizer	sgd	momentum	0.9
grad_clip	0	model	vgg16_bn

3 网络架构搜索

在本部分中，我尝试使用 NNI 最新的 Retiarii NAS 进行实验。Retiarii 是一个支持神经网络架构搜索和超参数调优的新框架。它允许用户以高度的灵活性表达各种搜索空间，重用许多前沿搜索算法，并利用系统级优化来加速搜索过程。我参考了 `nni/test/retiarii_test/darts/` 来编写程序，但是一直无法运行，NNI 无法识别到 GPU。与助教老师和 NNI 开发人员沟通后，认为是 Windows 的问题，需要使用 Linux。虚拟机无法使用 CUDA，故选择了 WSL2 作为运行平台，为此加入了 Windows Insider Preview。配置完成后 `torch.cuda.is_available()` 显示为 True，CUDA 版本也符合，但是仍然无法运行程序。现在猜测是由于 WSL2 中没有真正的 NVIDIA 驱动，无法运行 `nvidia-smi`，导致 NNI 和 `pytorch-lightning` 无法获取 GPU 信息。安装双系统时碰到了一些困难（主要涉及显卡开闭源驱动切换），在截止的时候还没有完成调试，故这项任务遗憾未能完成。