

NNI学生项目2020

Task 1.1

ID:14 Name:中科大飙车队

任务描述

- 1.安装NNI，尝试内置样例（eg.MN2IST）
- 2.为选择的样例添加NNI元素
- 3.使用NNI，自行跑腿程序
- 4.提交NNI集成的结果

体验

安装

Windows

在Windows电脑上由于配置过Anaconda，使用如下命令就可以很快的安装

```
1 | pip install nni
```

安装完成后，根据nni使用文档，这里有几个需要注意的地方

- config.yml文件中需要将command里的python3改为python

完成之后，因为没有自己的电脑上没有GPU资源，只需要用CPU即可，一些操作非常简单，配置完config文件之后用命令

```
1 | nnictl create --config config.yml
```

启动即可，并可通过web UI 可视化观察。

Linux

由于GPU使用的需要，需要在服务器上运行NNI。这里我们遇到几个问题，最终都顺利解决。

- 由于服务器上没有root权限，安装NNI必须在自己的目录下进行，使用pip命令安装后提示找不到nnictl命令，后来又顺利解决
- 完成后使用nnictl命令后使用命令提示缺少GPU_metric file
- 上述解决之后，使用nnictl 启动服务后任务一直处于WAITING的状态不动，只能手动结束服务，这在群中经过询问之后顺利解决，可能是服务器进程中残留进程一直在运行的缘故。
- 决定使用anaconda进行管理，下载好anaconad之后一键pip安装后成功解决。
- 由于服务器连接需要内网，没有校园网vpn只能通过在终端通过命令nnictl查看任务状态，也不能对服务器上的文件进行拷贝，后续只能通过截图给出

使用

由于对应的跑的的MNIST例子，这里发现一个疑惑的地方，官方给出代码中对同一个MNIST数据集下载了多次放在不同的trial中，个人认为这个操作没有必要，特别的浪费时间和空间。使用步骤如下

- 在代码中添加NNI元素，主要为一下三行代码

```
1 # report intermediate result
2 nni.report_intermediate_result(test_acc)
3 # report final result
4 nni.report_final_result(test_acc)
5 # get parameters form tuner
6 tuner_params = nni.get_next_parameter()
```

- 配置满足环境的需求的config.yml文件
- nnictl create --config config.yml创建nni
- 使用nnictl命令查看任务状态