AI 超算系统使用指南

目录

三大件的最低硬件要求

（1-16）张 GPU 物理卡：工作站可以 1-4 张，服务器至少 4 张以上 GPU 卡
物理内存=GPU 内存*（1-2）倍，物理内存不低于 GPU 显存总量
CPU 在 aida 的内存测试工具 read/writer 超过 100GB/s，主频不低于 3.0GB

普通用户升级 GPU

普通 PC 升级工作站可以加卡：M40，k40，QuadroK6000 都是 2000 元内可以入手计算卡，原有基础上额外插一张即可（用新卡做 AI 计算，不影响使用）。

超算系统作用于以下工具链

- Training Server+Training Client：网络化的超算服务体系，直接基于 VM 模型调度多 GPU 算力
  - CS 模式普适性是最好的，通过操作可以同时支持各种大规模计算任务
  - Training Server 适合公司内部有一台专用的建模服务器
  - Training Server 有能力发挥超算服务器的全部潜力，设备越好，潜力发挥会越充分
  - Training Server 最大可以支持到 128 张 GPU 卡的超算服务器
  - Training Server 可以部署于阿里云/腾讯云这类超算远程节点中，上传无网速限制，100GB 的样本只需几个小时就可完成上传
- Model Builder：建模时可以直接调度本地超算和远程 Training Server 超算，计算模式为 Normal Training
  - 使用方式于较早的版本一致，主要差异如下
  - Remote 工具模块不再内置，而使用 Training Client 替代了之前的内置 Remote 模块
  - 训练参数可使用”-GPU:0,1,2”来驱动多 GPU 算力
- Image Matrix Tool：建模时可以 Normal/Large-Scale 两种方式调度本地 VM 超算建模
  - 干掉了之前的”Build Training Package”模块
  - 本地 Normal Training VM 模块，直接驱动超算
  - 本地 Large-Scale Training VM 模块，直接驱动超算
- LargeScale Image Matrix：建模时可 Large-Scale 方式调度本地 VM 超算建模
  - 本地 Large-Scale Training VM 模块，直接驱动超算

建议超算用户使用授权 key，否则很可能遇到以下问题

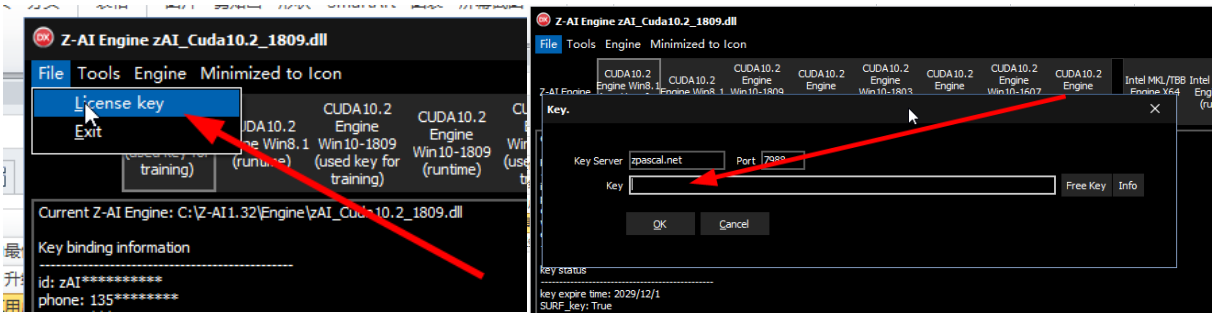
- 免费 key 会随 AI 引擎版本一起更新，免费时效一般在数月内，例如：5/1 日更新，免费 key（free user key）一般是 8/1 到期(大版本更新后一般 3 个月内免费，小版本一般 1 个月免费)，AI 引擎的 GPU 是免费且不需要 key，AI 引擎的 GPU 建模需要授权 Key 才能使用。
- 免费 key 无法安装 ToolChainSource/LicensedDemo，ToolChainSource 是整个 AI 引擎的工具链源码。LicensedDemo 是授权用户定制和应用级项目专属使用的 Demo。
- 免费 key 只提供了学习 AI 引擎的 FreeDemo 和 AI 库的源代码
- 不对免费用户提供硬件设备搭建咨询服务
- 不对免费用户提供技术咨询和支持服务
- 不对免费用户提供定制化 demo

查看授权 key 的到期时间

通过工具链主程序查看。

修改授权 Key

通过工具链主程序修改



部署超算服务器系统

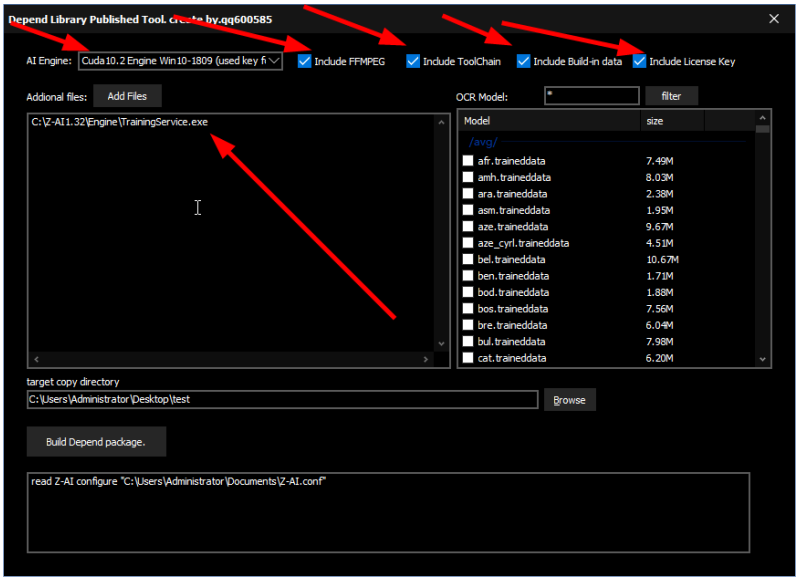
将 AI 算力放到超算服务器的部署方法

在超算服务器直接安装一个 AI 引擎（建议）

通过 https://zpascal.net 下载安装程序，填入 key，完成安装后，启动超算服务器
打开超算服务器后，注意箭头指向：
默认密码为 admin
addional parameter：这是训练虚拟机的附加启动参数，如果客户端没有给 addional param（为空），服务器将会使用该参数作为计算虚拟机的附加启动参数
temp directory：临时目录，该目录需要指向 nvme/ssd/pci-e 等接口的硬盘驱动器。注意：机械硬盘无法满足 multiGPU 的数据量，建议使用高速硬盘。

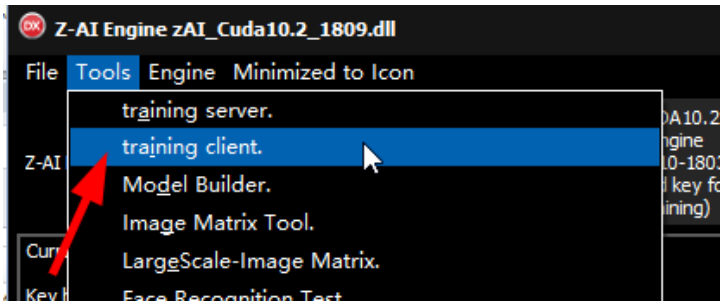
通过依赖库分发系统部署服务器

- AI Engine 部分要选择带 cuda 字样且可以训练的计算引擎，另外下图箭头指向的开关都要勾上
- addional files 里面添加 trainingserver.exe（超算服务器）
- 选择好目标目录后点 build depend package：它会给你部署一套大约 200M 的系统，这时候打包 copy 到服务器，点开 trainingserver.exe 即可

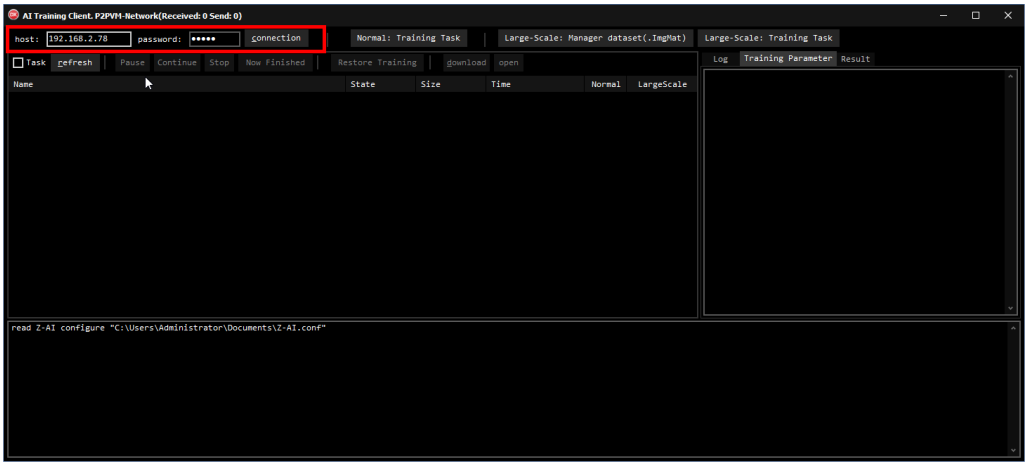


超算客户端使用指南

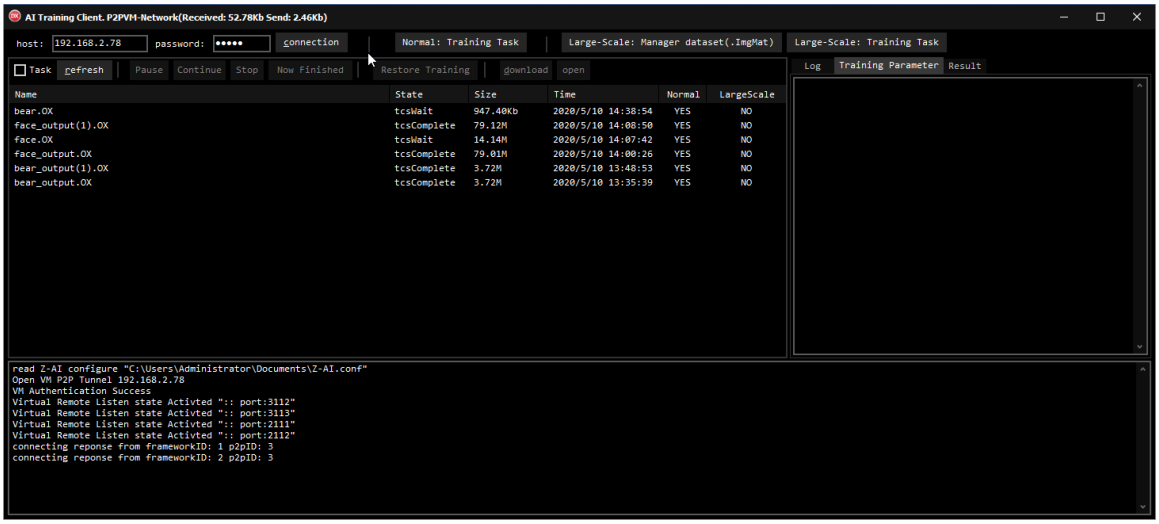
客户端可以通过工具链主程序打开，如下图，点开”training client.”即可



打开后填入超算服务器的地址和密码，connection 即可



超算服务器的网络架构以最尖端的 p2pVM 为主，为避免刷屏服务器/客户都安均关闭了网络 log 状态，任何时它都是安静的。同时超算服务器体系的容错性很高，非常稳定，属于中等规模的超算网络体系。

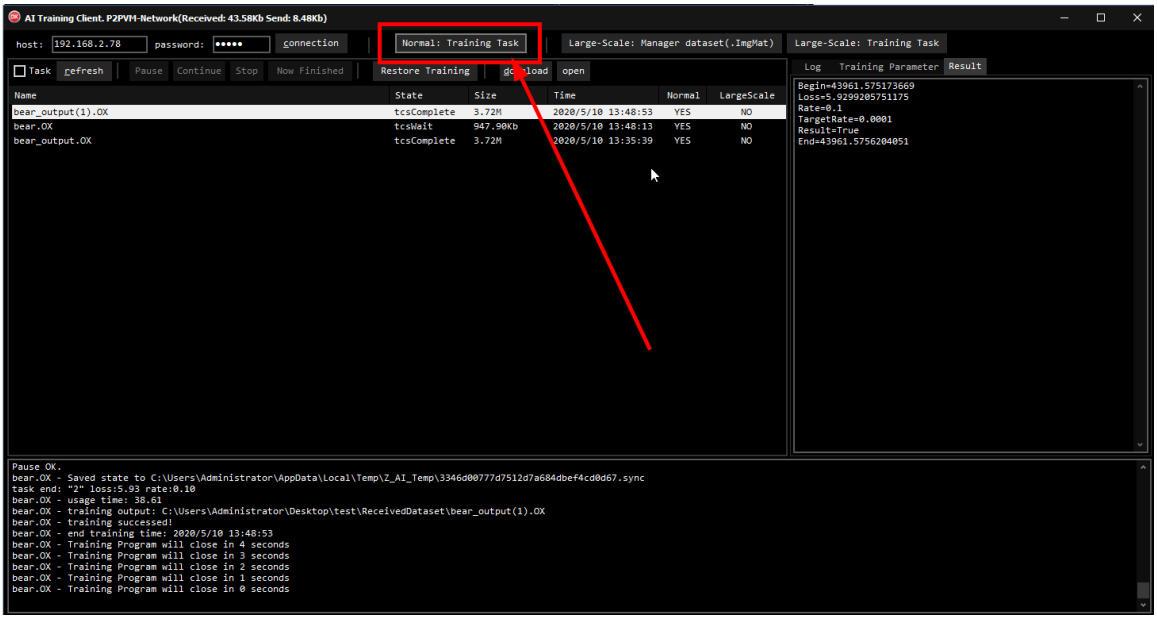


Normal Training 模式

Normal Training 是 input/output 都在同一个容器中的模式，假如我们的样本 100M，训练时会将样本+训练参数+所有的附加数据全部打包成一个后缀为.OX 的文件（该文件可以使用 File Package 工具打开编辑），当训练完成后，这个些文件会附加到输出的数据中。

用最简单的方式理解 Normal Training：输入/输出都是一个数据包文件，非常方便管理。缺点是无法负载大规模样本的输入/输出，Normal Training 适合小样本规模的模型训练。

如下图，我们点开 Nomral: Training Task



会看到一个 Normal Training task 的参数窗口

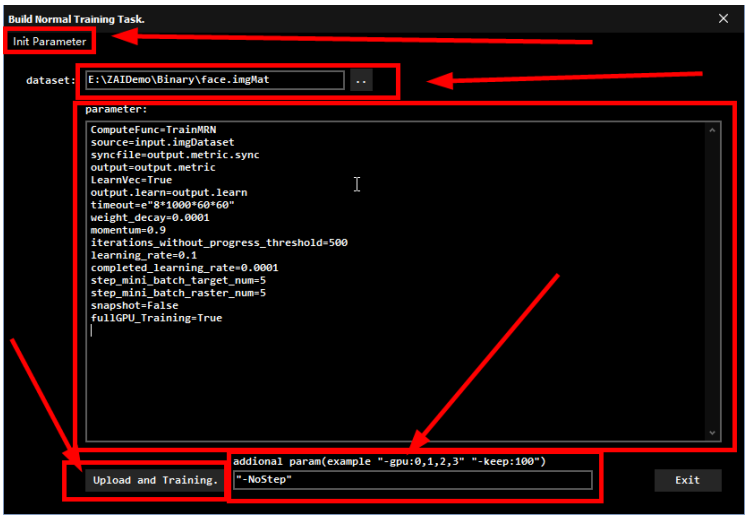
Init Parameter：选择 Parameter 预置的超参数菜单

dataset：选择样本数据，这些样本数据可以支持以下样本库

- .AI_Set（Model Builder 工具的样本格式）
- .Img_Mat（Image Matrix 工具的样本格式）
- .OX（来自已有训练任务的输出或则输入）

addional Param：超算虚拟机启动参数，如果使用多 GPU，这里需要给”-gpu:0,1,2”，按服务器支持 GPU ID 给定

Upload And Training：直接上传给服务器，完成后会自动开始训练



Normal Training 可以同时支持 CPU 架构训练+GPU 架构训练

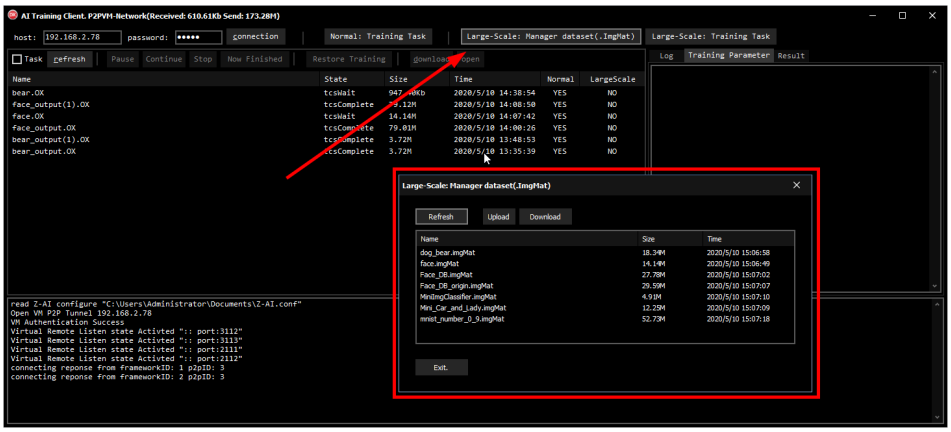
- Normal Training 支持的 CPU 架构模型
  - OD3L：可以暂停/继续，不支持 Stop/Now Finished 操作，不支持中续训练
  - OD6L：可以暂停/继续，不支持 Stop/Now Finished 操作，不支持中续训练
  - ODMarshal6L：可以暂停/继续，不支持 Stop/Now Finished 操作，不支持中续训练
  - SP：可以暂停/继续，不支持 Stop/Now Finished 操作，不支持中续训练
- Normal Training 支持的 GPU 架构模型
  - MMOD3L：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - MMOD6L：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - Metric：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - LMetric：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - RNIC：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - LRNIC：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - GDCNIC：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - GNIC：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - SS：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练

Large-scale Training 模式

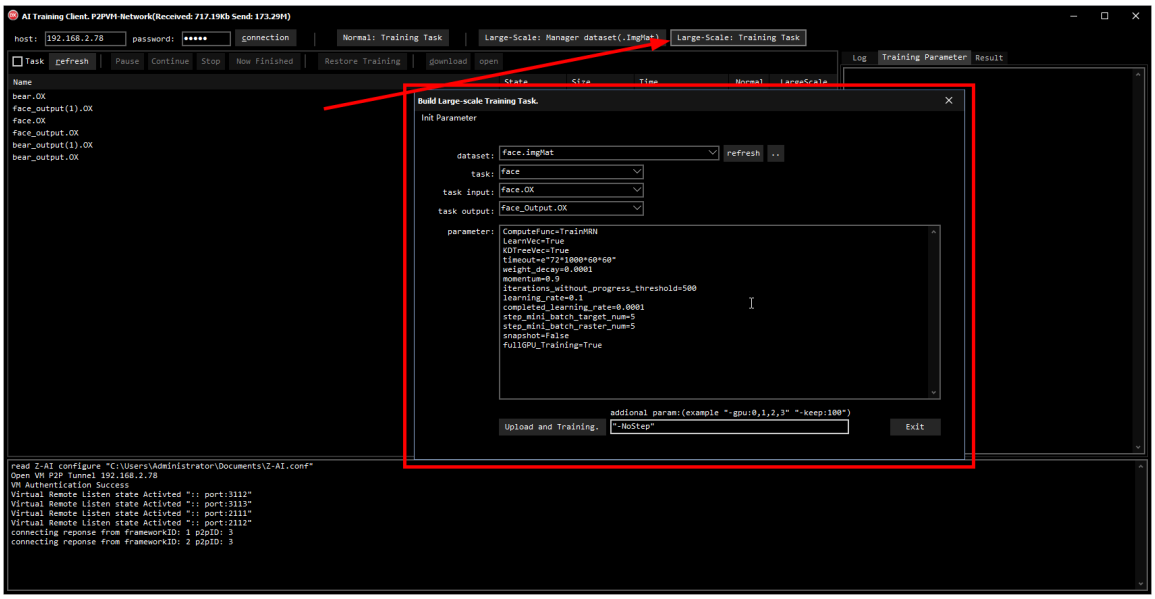
超算服务器支持首次是在 1.32 版本发行的，超算服务器重点支持的功能就是 Large-Scale Training 模式
Large-Scale 模式的工作在客户都安操作要分两步走

上传数据样本

下图是上传数据样本的基本操作，数据样本只能支持.ImgMat(Image Matrix Tool 格式)
数据样本的上传与下载均支持断点续传，并且全部测试 passed，大样本传输不会出错
超算服务器主循环的默认吞吐量为 512k，数据上传/下载的最大数据吞吐量为 50M/s，如果感觉吞吐量不够，可以自行重构超算服务器体系



构建 Large-Scale 训练参数



Large-Scale 训练参数只能支持 DNN 技术体系的模型

Large-Scale 不支持所有的基于 CPU 计算的模型架构

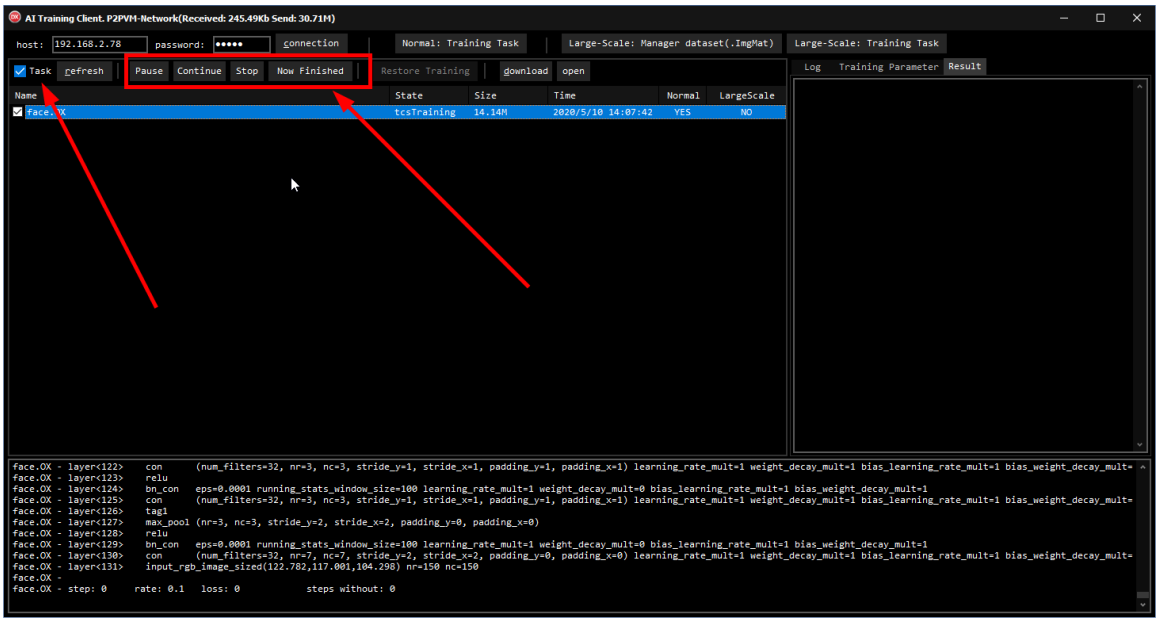
- 不支持 CPU 架构模型
  - OD3L：不支持，无法建模，无预置建模参数
  - OD6L：不支持，无法建模，无预置建模参数
  - ODMarshal6L：不支持，无法建模，无预置建模参数
  - SP：不支持，无法建模，无预置建模参数
- 支持的 GPU 架构模型
  - MMOD3L：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - MMOD6L：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - Metric：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - LMetric：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - RNIC：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - LRNIC：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - GDCNIC：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - GNIC：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练
  - SS：可以暂停/继续，支持 Stop/Now Finished 操作，支持中续训练

在训练中的可以对单个任务进行远程状态操作

训练中，如下图，task 开关会自动勾上，这个开关表示正在训练的任务，我们在列表中，会看到我们上传的 face.ox 正在训练

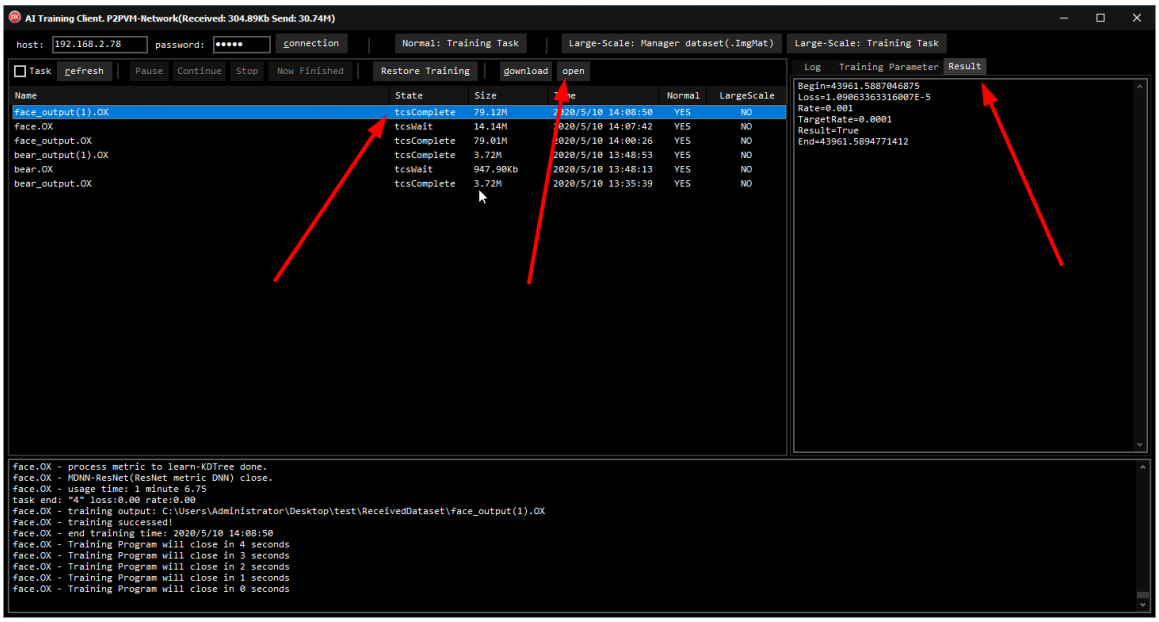
训练中，如下图，红框的状态操作功能可以生效

- pause：暂停训练，会立即暂停该任务在超算服务器的 GPU 计算，所有的 GPU 算力都会释放出来，成为 0%负载消耗，**暂停不会释放 GPU 使用的显存，只会释放出计算资源，如果有紧急训练任务需要插队，可以使用 pause**
- continue：继续训练，必须对暂停中任务操作才能生效
- stop：立即停止超算服务器对该任务 GPU 计算的 VM 虚拟机，并释放显存，返回该任务中止状态
- Now Finished：立即停止超算服务器对该任务 GPU 计算的 VM 虚拟机，并释放显存，返回该任务完成状态

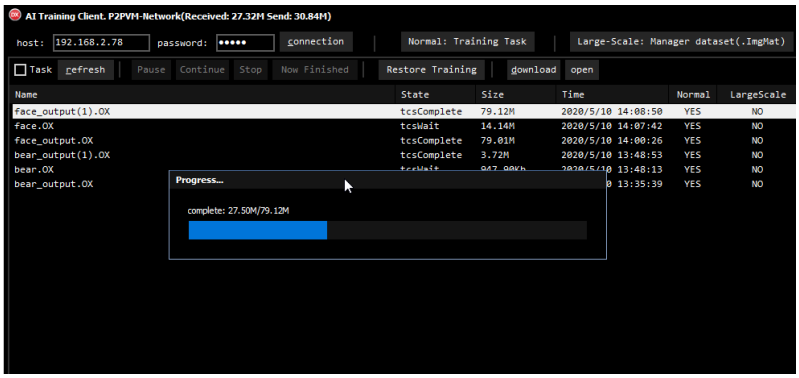


获取训练完成的模型

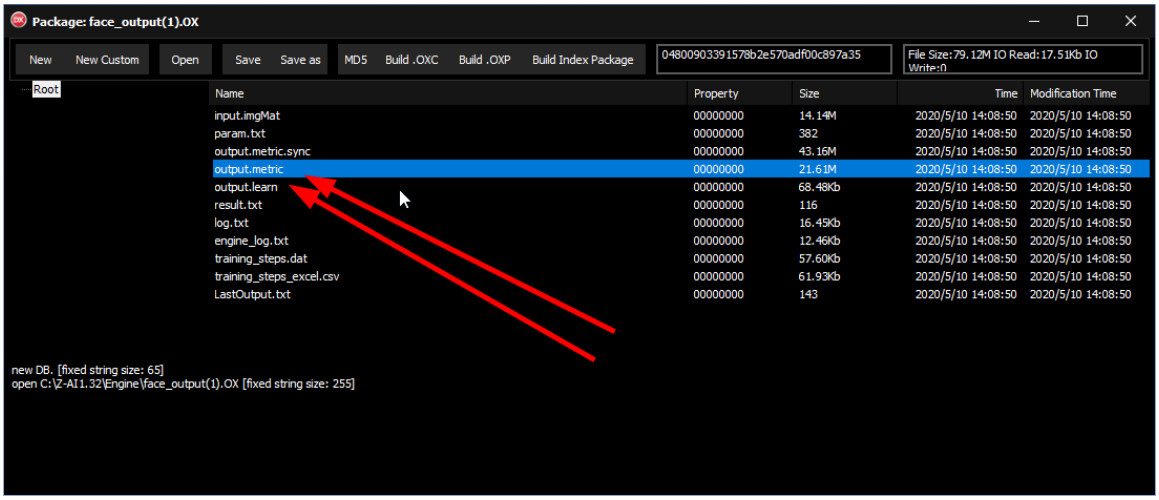
在主视口中，state 是 tcsComplete 都是已经完成的模型，旁边的 Result 则包含了该任务的 loss，rate 等等状态直接点 open 按钮，会自动化从服务器下载该任务，整个下载过程都是自动化支持断点续传的



下载过程中会有一个进度条，在下载过程中耐心等待即可



下载完成后，会打开 File Package 工具，这里面都是超算服务器训练好的模型，直接导出使用即可

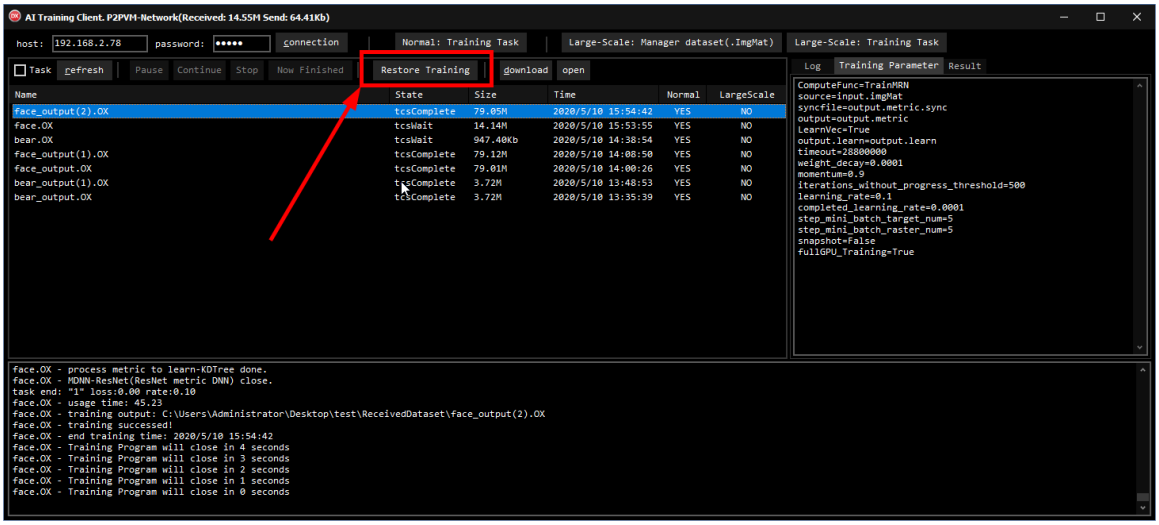


中续训练

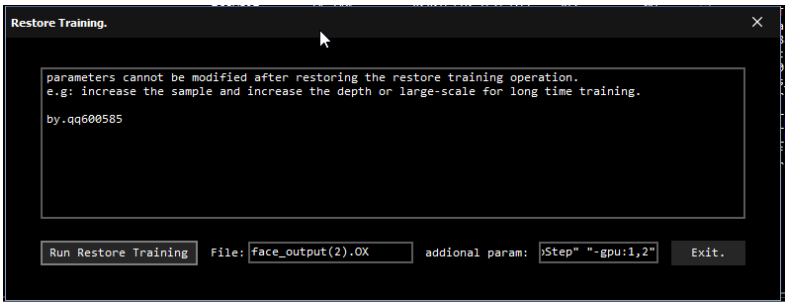
中续训练就是在长时间训练中，突然人为或则断电中断，这时候，通过状态还原系统，重新还原之前的训练状态，从过程中间继续上次未完成的训练任务。

中续训练内部是个非常复杂的支持体系，它需要区分 Normal/Large-Scale 两种训练模式，在超算服务器支持体系中，它是傻瓜化的一键中续。

凡是在列表中已有的任务，选中它，只要 Restore training 按钮亮了，都是一键完成中续训练。

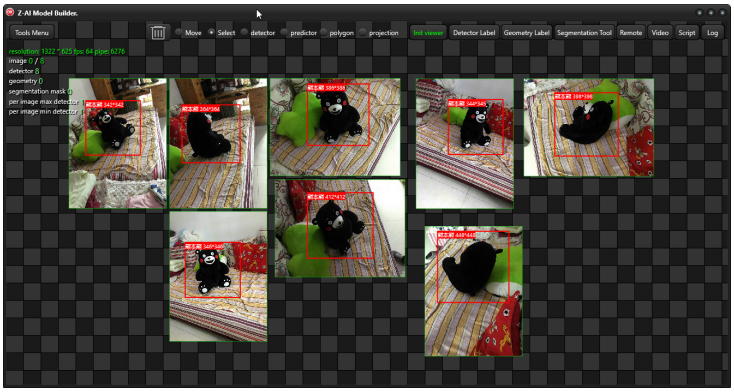


点击“Restore Training”后，会弹出对话框，让我们确认中续目标，这里需要注意：addional param 需要重新给定，如果之前是多 gpu 训练模型，中续时如果不给参数，超算服务器会变成单 GPU 来跑 VM。

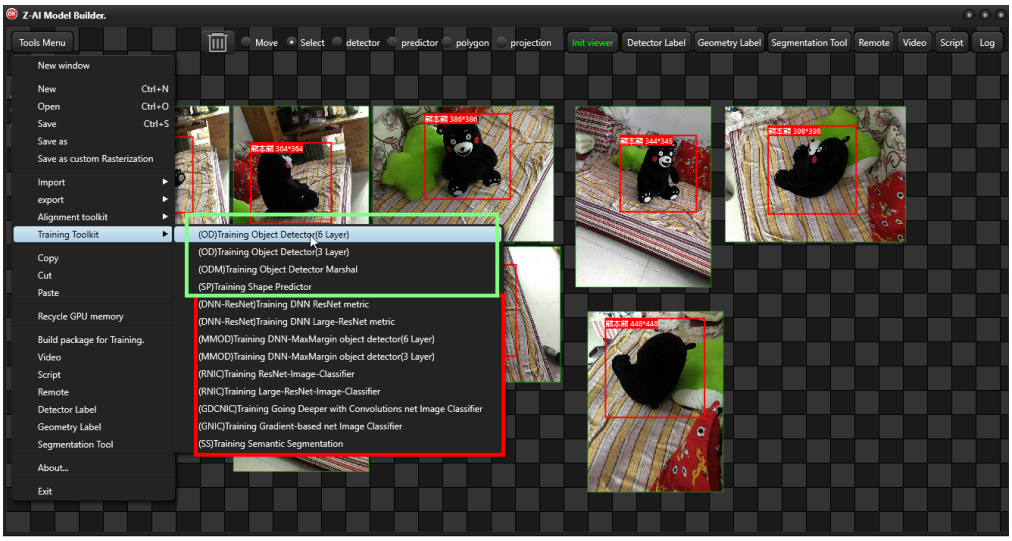


在 Model Builder 使用超算技术

先打开样本数据库

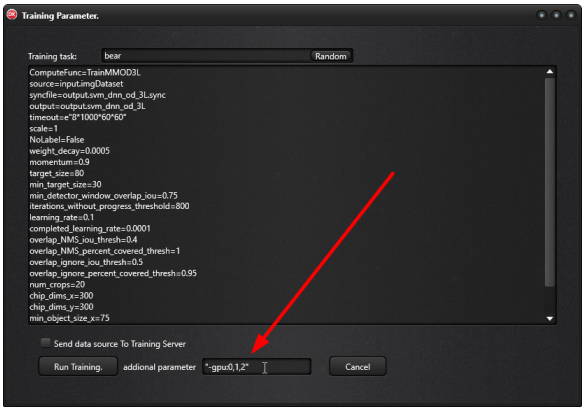


通过 Training Toolkit 打开训练器，**红框中的 AI 模型都可以直接使用 GPU 的超算技术，绿框中的 4 个模型都是使用 CPU 建模的技术不可以支持超算**。

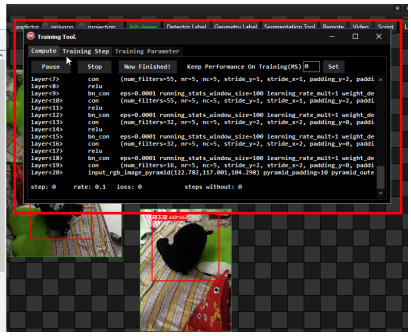


使用本地超算训练

这种方式适用于查了多张 GPU 卡的工作站，在 Addional parameter 框中，需要给出多个 GPU 的 ID
下图使用了 3 个 GPU，以参数，”-gpu:0,1,2”来启动 GPU 的 VM 训练器



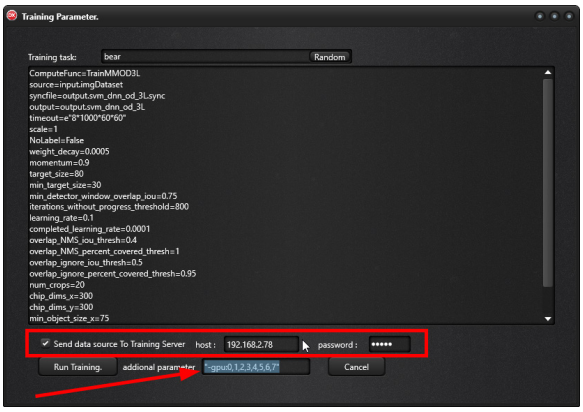
开始训练时，会弹出一个 Training Tool 窗口，这是 VM 技术的虚拟训练环境，容错性非常高。
在训练中，可以通过 nv 的控制面板，来查看多张 GPU 卡的利用率



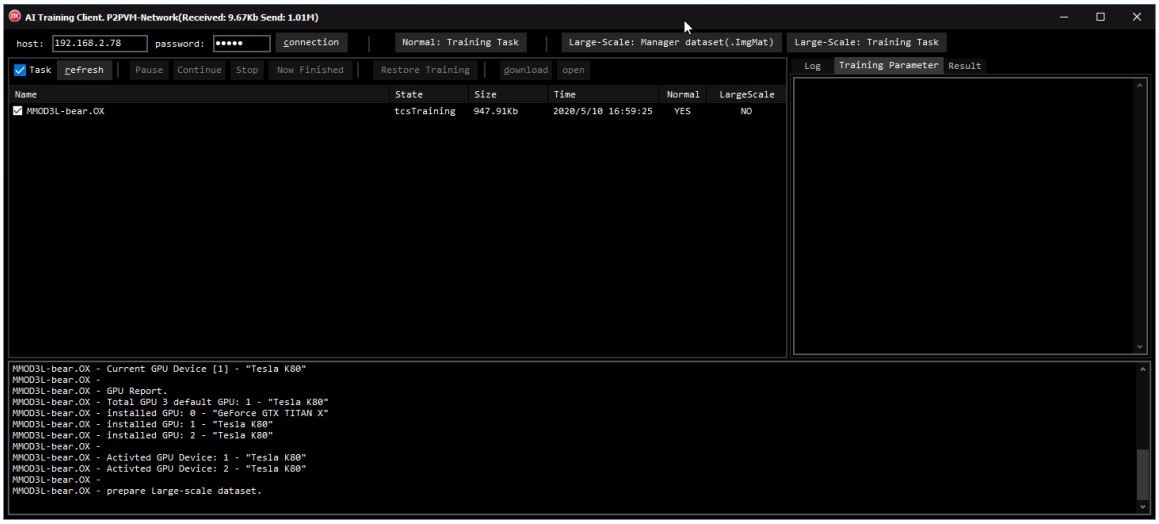
同样也可以通过命令行 **nvidia-smi.exe -l 2** 这类参数来监控 GPU 的使用率
不建议使用 gpu-z 来查看计算卡，gpu-z 可以查看民用图形卡，对于 tesla/quadro 这类计算卡，它的支持有很多不足。

在 Model Builder 使用远程超算

勾上远程超算，填入服务器地址+密码
在 addional parameter 栏，要按超算服务器端的 GPU ID 来给
如果服务器插了 8 张卡，addional parameter 可以给"-gpu:0,1,2,3,4,5,6,7"



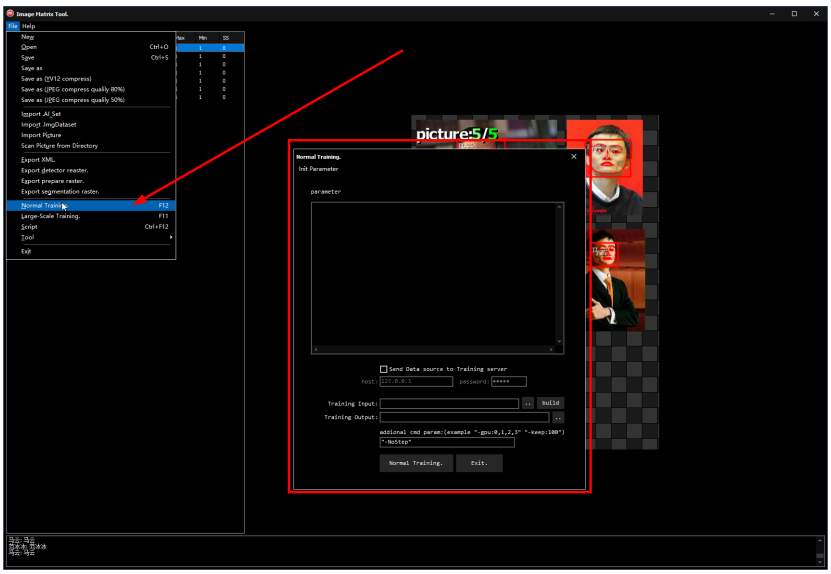
点”Run Training”后，会自动启动 Training Clinet 并且上传数据样本给服务器，然后自动化开始训练，整个过程会全自动化，无需人为干预。



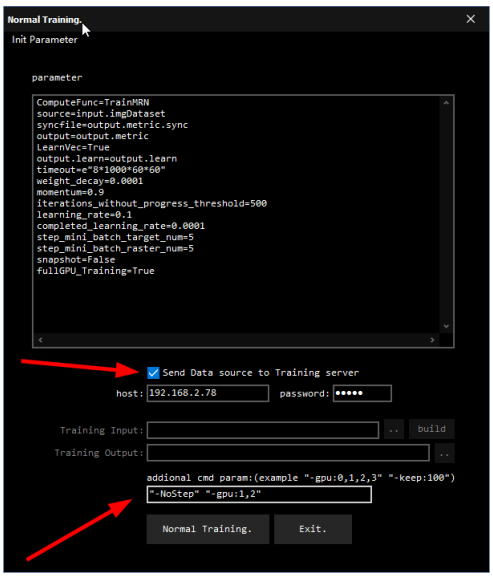
在 Image matrix Tool 使用超算技术

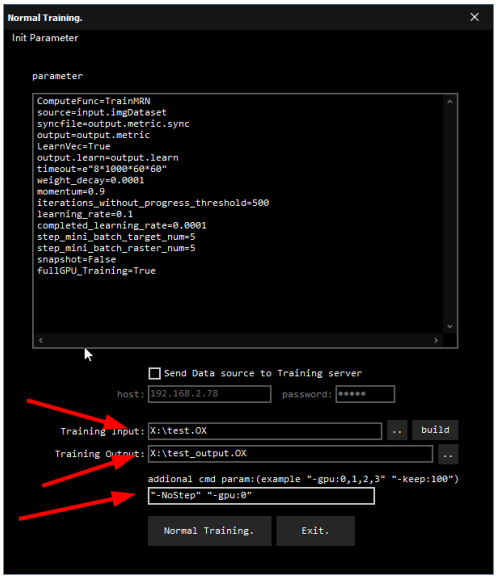
Image matrix Tool 支持两种模式接入超算技术

使用 Normal Training 模式训练模型



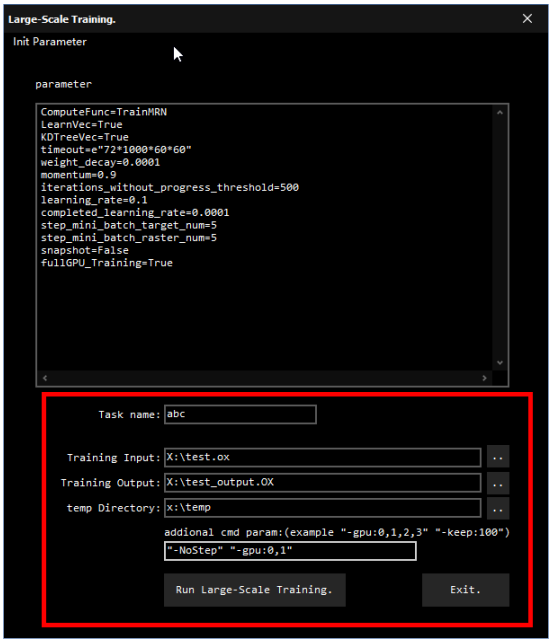
左图的 Normal Training 使用远程超算的 GPU 训练 右图使用本地 GPU 进行训练





使用 Large-Scale Training 模式训练模型

- 下图红框中每个参数都必须明确给出，不可以给空
- Large-Scale training 模式可以训练数百 GB 体量的样本数据，训练启动非常快
- Large-Scale Image Matrix 工具中的 Large-Scale Training 与 Image Matrix Tool 是一样的



addional parameter 详解

addional parameter 是指在训练时附加给 VM 训练程序的启动参数，例如”-GPU:0,1,2”，表示以 3 个 GPU 启动合算任务，该小节为 Addional Parameter 罗列了所有开放的可用参数

变量
-gpu:id，指定训练程序使用的 gpu 卡，以 gpu id 作为区分，例如，”-gpu:1,2”
-hint:text，在执行训练前显示一行信息，信息会停留 5 秒，之后继续执行训练
-CloseDelay:second，当 VM 训练器关闭时默认有 5 秒的延迟，该参数可以定义关闭延迟，时间是秒
-keep:ms，每一个训练步数完成后，需要等待的毫秒值，如果需要留出 gpu 的算力，这个值可以给大，比如 500，表示每秒最大只计算 2 个步数

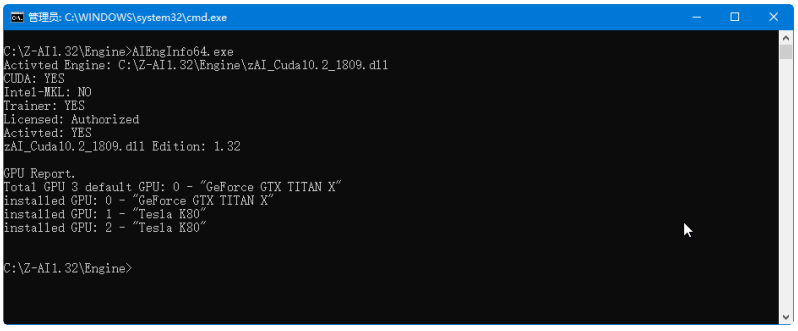
开关
-ShowStep，显示每一个步数，打开该参数容易造成刷屏问题，默认是关闭的，每隔 30 秒显示一次训练进度
-NoStep，不显示步数，避免刷屏

除此之外的其余开关都是系统内置，不能乱给，详情需要自行研究 TrainingTool 的源代码。

查看 GPU ID

方法 1，打开工具链主程序，选择一个 cuda 类计算引擎，会看到有效的 GPU ID

方法 2：运行 AIEngInfo64.exe，该文件为命令行程序，需要在命令行窗口执行
建议查看服务器的 GPU ID 信息时，在服务器端执行该命令



by.qq600585
2020-5