

# 昇腾算力 AI 训练指南

随着深度学习和强化学习等技术的发展，基于神经网络的模型已经在 Botzone 大多数棋牌类游戏的天梯榜上名列前茅。Botzone 本身作为多智能体游戏 AI 评测平台，服务器仅支持用户 Bot 的推理时计算，并不提供 GPU 等异构芯片算力资源以承载需要耗费大量算力的训练任务，用户需要依靠自己的 GPU 算力或云算力服务商提供的算力平台来进行 Bot 模型的训练。

本指南以国标麻将 AI 监督学习训练为例，给出基于华为云提供的昇腾 AI 芯片算力，支撑基于神经网络的 AI 训练任务的详细流程。

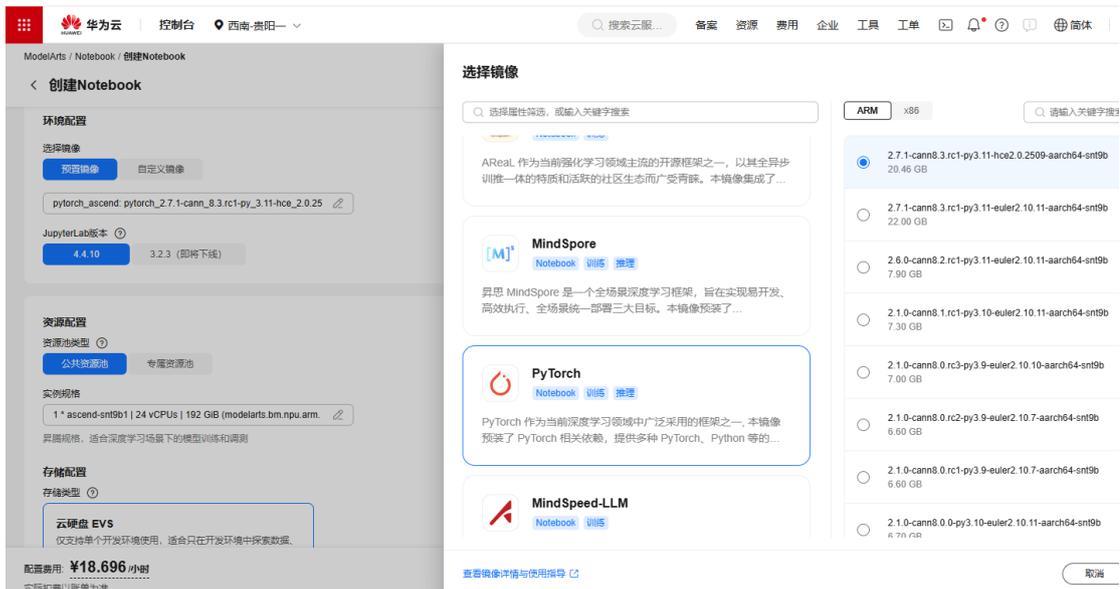
## 一、在华为云上创建 Notebook 容器

(1) 注册并登录华为云账号；

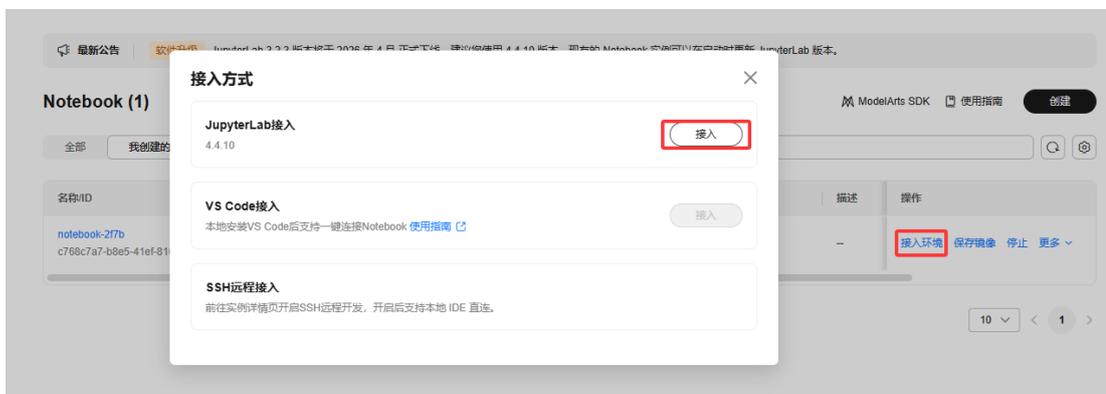
(2) 访问 <https://console.huaweicloud.com/modelarts>，将左上角的机房区域调整成“西南-贵阳一”（该区域有昇腾 910B 算力资源），在左边菜单栏选择 Notebook 并点击创建实例；



(3) 在 Notebook 的配置界面，选择合适的镜像、资源配置和存储配置。本指南中使用单卡 910B 的资源配置，镜像选择 ARM 架构下具有 PyTorch 的最新版本 cann 工具链和 python 版本的镜像，存储配置选择 10G，单价约 18.7 元每小时。校内课程使用时老师可以通过 <https://edu.hicomputing.huawei.com/industry-edu-base> 为学生申请算力券；



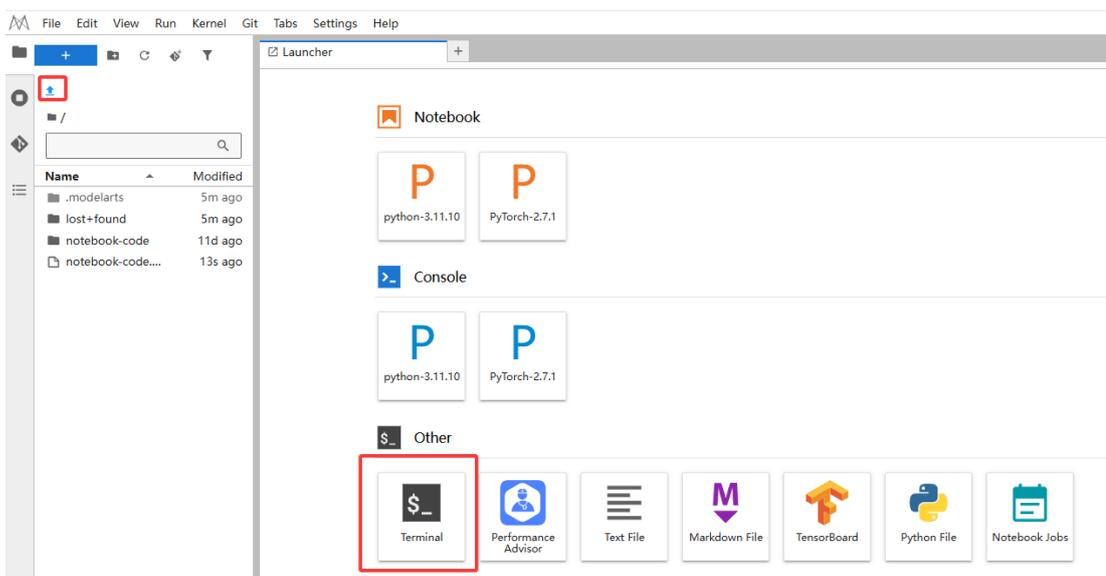
(4) 容器开启后，通过容器列表的菜单接入环境。建议设置容器自动停止的时间（设置之后也可以视情况延长），并在训练完成后手动停止容器，因为停止期间只有挂载的硬盘计费（费用很低），以节省算力成本。在容器内需要将有效数据均保存在挂载硬盘即 /home/ma-user/work 目录下，避免容器停止再重新启动时丢失数据。



## 二、在 Notebook 容器中训练国标麻将 AI 模型

(1) 从 <https://botzone.org.cn/static/notebook-code.zip> 下载 notebook-code.zip 文件包，该文件包包含了国标麻将监督学习的示例代码以及训练数据；

(2) 点击左上角上传按钮将压缩包传到容器内，然后点击 Terminal 启动一个命令行，命令行刚启动时可能需要等待一段时间进行环境初始化；



(3) 在命令行中依次运行以下指令：

|                         |   |
|-------------------------|---|
| unzip notebook-code.zip | 解压文件包   |
| cd notebook-code        | 进入代码目录  |
| pip install PyMahjongGB | 安装国标麻将算番库   |
| python preprocess.py    | 将原始对局数据转换为 numpy 数组，该步骤需要约 20 分钟，数据保存在 notebook-code/data 下 |
| python supervised.py    | 运行监督学习训练  |

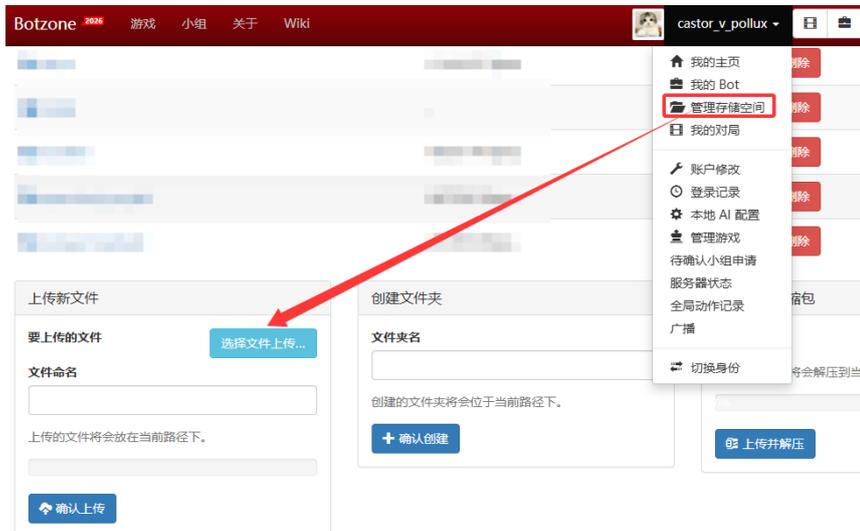
```
Processing match 96256 6163fb025ddc087351c9ba6f...
Processing match 96384 6163fbd85ddc087351c9bd60...
Processing match 96512 6163fcb35ddc087351c9c01f...
Processing match 96640 6163fd825ddc087351c9c300...
Processing match 96768 6163fe6c5ddc087351c9c5bc...
Processing match 96896 6163ff4f5ddc087351c9c87b...
Processing match 97024 616400345ddc087351c9cb3e...
Processing match 97152 616401125ddc087351c9cddc...
Processing match 97280 616401ea5ddc087351c9d097...
Processing match 97408 616402c55ddc087351c9d34d...
Processing match 97536 616403a95ddc087351c9d5fa...
Processing match 97664 616404915ddc087351c9d8e1...
Processing match 97792 616405795ddc087351c9dbd7...
Processing match 97920 616406575ddc087351c9de8f...
Processing match 98048 6164072b5ddc087351c9e160...
Processing match 98176 616407fb5ddc087351c9e41d...
(PyTorch-2.7.1) [ma-user notebook-code]$python train.py
python: can't open file '/home/ma-user/work/notebook-code/train.py': [Errno 2] No such file or directory
(PyTorch-2.7.1) [ma-user notebook-code]$python supervised.py
/home/ma-user/anaconda3/envs/PyTorch-2.7.1/lib/python3.11/site-packages/torch_npu/utils/collect_env.py:58: UserWarning: Warning:
end-toolkit/latest owner does not match the current owner.
  warnings.warn(f"Warning: The {path} owner does not match the current owner.")
/home/ma-user/anaconda3/envs/PyTorch-2.7.1/lib/python3.11/site-packages/torch_npu/utils/collect_env.py:58: UserWarning: Warning:
end-toolkit/8.3.RC1/aarch64-linux/ascend_toolkit_install.info owner does not match the current owner.
  warnings.warn(f"Warning: The {path} owner does not match the current owner.")
/home/ma-user/anaconda3/envs/PyTorch-2.7.1/lib/python3.11/site-packages/torch_npu/utils/collect_env.py:58: UserWarning: Warning:
end-toolkit/latest owner does not match the current owner.
  warnings.warn(f"Warning: The {path} owner does not match the current owner.")
/home/ma-user/anaconda3/envs/PyTorch-2.7.1/lib/python3.11/site-packages/torch_npu/utils/collect_env.py:58: UserWarning: Warning:
end-toolkit/8.3.RC1/aarch64-linux/ascend_toolkit_install.info owner does not match the current owner.
  warnings.warn(f"Warning: The {path} owner does not match the current owner.")
loading 0
loading 128
loading 256
loading 384
```

```
ma-user@notebook-c768c X +
loading 8832
loading 8960
loading 9088
loading 9216
loading 9344
loading 9472
loading 9600
loading 9728
Epoch 0
Iteration 0/5153 policy_loss 1.951117992401123
Iteration 128/5153 policy_loss 1.3288288116455078
Iteration 256/5153 policy_loss 1.1833256483078003
Iteration 384/5153 policy_loss 1.1212856769561768
Iteration 512/5153 policy_loss 1.095914363861084
Iteration 640/5153 policy_loss 1.057652235031128
Iteration 768/5153 policy_loss 0.9954516887664795
Iteration 896/5153 policy_loss 1.0302046537399292
Iteration 1024/5153 policy_loss 0.9709790349006653
Iteration 1152/5153 policy_loss 0.9667976498603821
Iteration 1280/5153 policy_loss 0.8968393802642822
Iteration 1408/5153 policy_loss 0.925978422164917
Iteration 1536/5153 policy_loss 0.9147598743438721
Iteration 1664/5153 policy_loss 0.9407048225402832
Iteration 1792/5153 policy_loss 0.9261711835861206
Iteration 1920/5153 policy_loss 0.9104251861572266
Iteration 2048/5153 policy_loss 0.931752622127533
Iteration 2176/5153 policy_loss 0.8438109159469604
Iteration 2304/5153 policy_loss 0.9161695241928101
Iteration 2432/5153 policy_loss 0.8778056502342224
Iteration 2560/5153 policy_loss 0.9179399609565735
Iteration 2688/5153 policy_loss 0.8855304718017578
Iteration 2816/5153 policy_loss 0.8932324051856995
Iteration 2944/5153 policy_loss 0.8932958245277405
Iteration 3072/5153 policy_loss 0.884306788444519
Iteration 3200/5153 policy_loss 0.8839064240455627
```

(4) 训练时每个 epoch 的模型会保存在 notebook-code/model/checkpoint 目录下，可以从左侧下载模型文件。

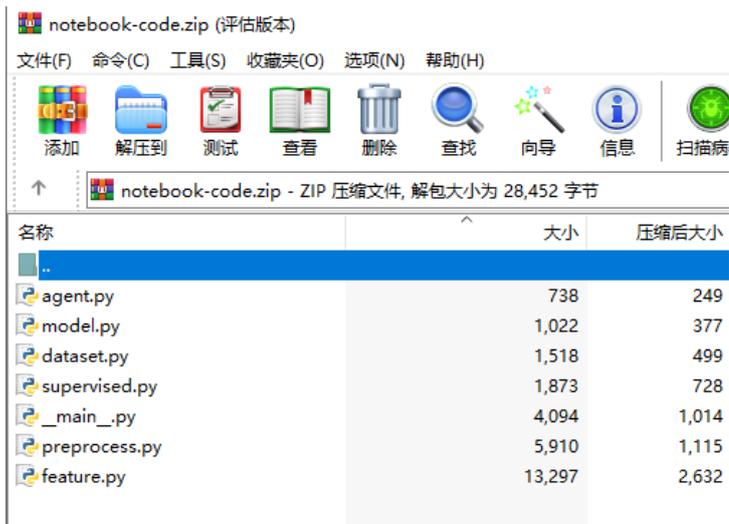
### 三、在 Botzone 上创建 Bot 并参与对局

- (1) 注册并登录 Botzone 账号；
- (2) 将前一步下载模型文件，上传到网站的用户存储空间，如图所示；



- (3) 将 notebook-code.zip 代码包在本地解压，将 `_main_.py` 中的 `data_dir` 目录修改为前一步上传的模型文件名，然后将文件包中的所有 python 代码打包成一个压缩文件，注意不要嵌套任何文件夹，所有 python 代码文件都在压缩文件的根目录中；

```
7 # Botzone interaction
8 import numpy as np
9 import torch
10
11 def obs2response(model, obs):
12     logits = model({'observation': torch.from_numpy(np.expand_dims(obs[
13         'observation'], 0)), 'action_mask': torch.from_numpy(np.expand_dims(obs[
14         'action_mask'], 0))})
15     action = logits.detach().numpy().flatten().argmax()
16     response = agent.action2response(action)
17     return response
18
19 import sys
20
21 if __name__ == '__main__':
22     model = CNNModel()
23     data_dir = '/data/10.pkl' # TODO: modify the name of checkpoint
24     model.load_state_dict(torch.load(data_dir, map_location = torch.device('cpu')))
25     model.train(False)
```



- (4) 在 Botzone 上创建 Bot，其中游戏选择 Chinese-Standard-Mahjong；源代码选择上传文件，将上一步得到的代码包上传；编译器选择 Python3.6.5；勾选长时运行和简单交互；不勾选开源代码；然后点提交；



**Bot 名字**   
Bot 名字应当在 3 到 18 个字符之间。

**游戏**   
选择你的 Bot 要参与的游戏。

**Bot 版本**

**Bot 简介**   
Bot 简介不要超过 100 个字符。

**源代码**  notebook-code.zip  
可用代码语言: C/C++, Java, C#, Javascript, python. 请保证用 UTF-8 编码。  
如果对输入输出格式不理解请查看维基的对应游戏和 Bot 词条。  
文件最大大小: 4.00 MB。  
如果你希望上传数据文件, 如模型文件等, 请使用 [用户存储空间](#)。

**运行时限** 该语言每回合运行时限 (对局第一回合翻倍):  
6 秒  
在“长时运行”模式下, 除第一回合外的运行时限:  
4 秒

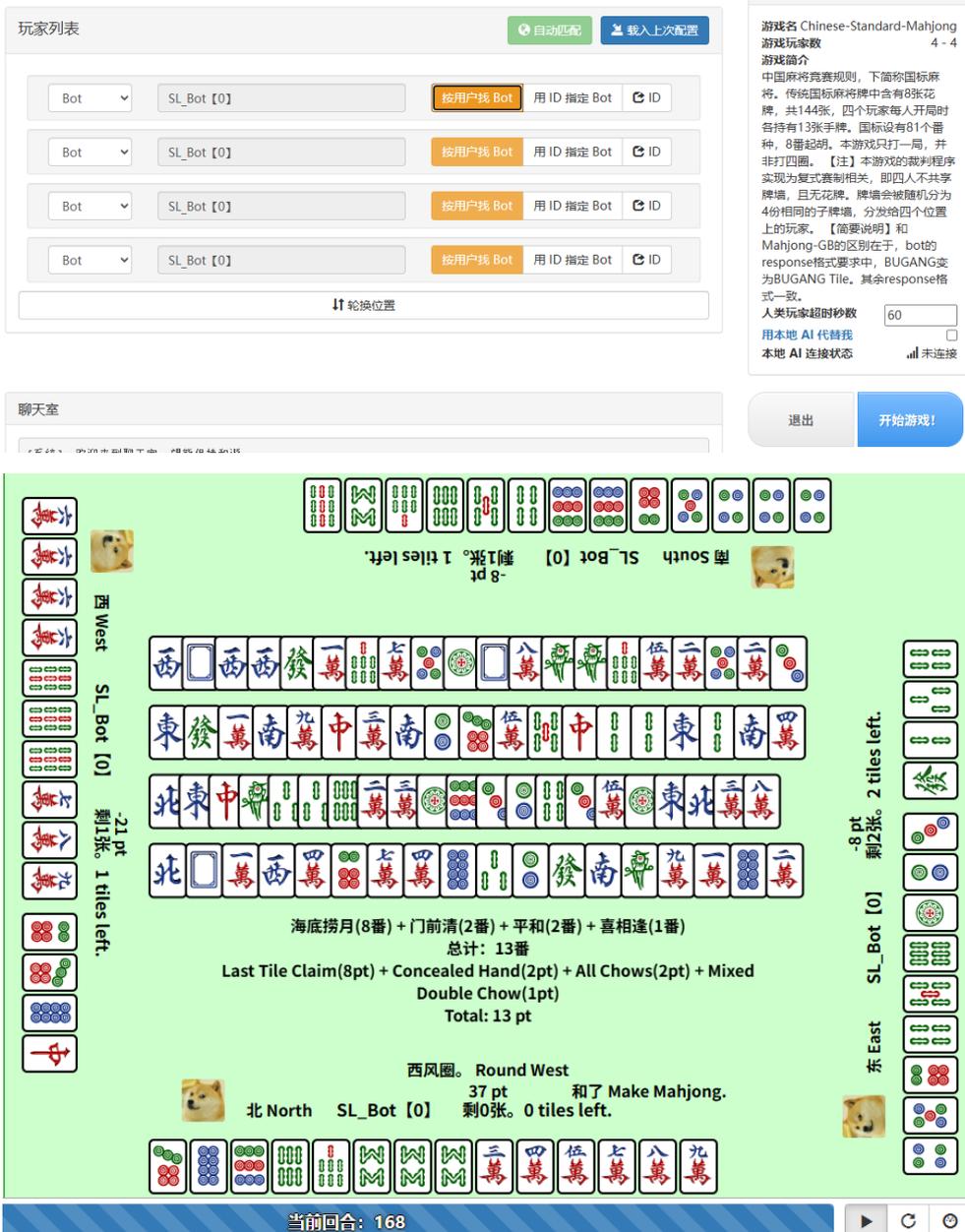
**允许长时运行 / 输出输入后不 EOF**  勾选此项, 则在给 Bot 输入后不关闭 stdin (标准输入), 从而允许“长时运行”模式。  
这种情况下请务必不要尝试读到输入结束再决策, 否则会陷入死等, 导致超时。  
请注意, 勾选后会立刻生效, 该 Bot 的所有版本都会受到影响。

**使用简单交互**  勾选后使用简单交互, 不勾选则使用 JSON 交互。  
某些游戏可能不支持简单交互。  
请注意, 勾选后会立刻生效, 该 Bot 的所有版本都会受到影响。

**开源**  选择此项, 所有人就能看到你 Bot 源码了。

(5) 使用新建的 Bot 创建新局, 可以正常对局。





本文中提供的代码仅为示例代码，用于提供一个最简的示例来说明华为云算力昇腾 AI 芯片的使用方式以及将使用神经网络模型的 Bot 上传到 Botzone 的过程。用户可以自行调整 AI 算法中的神经网络结构、麻将输入特征、训练超参数等，以得到更强的麻将 AI，或者对其他 Botzone 上有的棋牌类游戏编写自己的 AI 训练算法，如强化学习等。