

## 目 录

在英特尔独立显卡上训练 TensorFlow 模型.....	1
1.1 英特尔锐炫™独立显卡简介 .....	1
1.2 蝰蛇峡谷简介 .....	1
1.3 搭建训练 TensorFlow 模型的开发环境 .....	2
1.3.1 Windows 版本要求: .....	2
1.3.2 下载并安装最新的英特尔显卡驱动 .....	2
1.3.3 下载并安装 Anaconda.....	2
1.3.4 安装 TensorFlow2 .....	3
1.3.5 安装 tensorflow-directml-plugin.....	3
1.4 在英特尔独立显卡上训练 TensorFlow 模型 .....	3
1.4.1 下载并解压 flower 数据集.....	错误!未定义书签。
1.4.2 下载训练代码启动训练.....	错误!未定义书签。
1.5 总结: .....	5

# 在英特尔独立显卡上训练 ResNet PyTorch 模型

作者：武卓，张晶

《[在英特尔独立显卡上训练 TensorFlow 模型](#)》介绍了在英特尔独立显卡上训练 TensorFlow 模型的全流程；本文将基于[蝰蛇峡谷\(Serpent Canyon\)](#)详细介绍如何在英特尔独立显卡上训练 ResNet PyTorch 模型的全流程。

## 1.1 英特尔锐炫™ 独立显卡简介

英特尔锐炫™ 显卡基于 Xe-HPG 微架构，Xe HPG GPU 中的每个 Xe 内核都配置了一组 256 位矢量引擎，旨在加速传统图形和计算工作负载，以及新的 1024 位矩阵引擎或 Xe 矩阵扩展，旨在[加速人工智能工作负载](#)。

英特尔锐炫™ 显卡 A7系列		XeSS	Xe ULTIMATE	XM AI Acceleration	Xe Media Engine	PCI EXPRESS 4.0
产品名	英特尔锐炫™ A750显卡 限量版	英特尔锐炫™ A750显卡	英特尔锐炫™ A770显卡			
微架构	Xe HPG	Xe HPG	Xe HPG			
Xe 核心数	28	28	32			
光线追踪单元	28	28	32			
显卡时钟频率	2050 MHz	2050 MHz	2100 MHz			
显卡内存(GDDR6)	8GB	8GB	16GB			
显卡总线带宽	512 GB/s	512 GB/s	560 GB/s			
显卡功耗TDP	225W	225W	225W			
Xe 矢量引擎	448	448	512			
可变速率着色 (VRS)	是	是	是			

## 1.2 蝰蛇峡谷简介

[蝰蛇峡谷\(Serpent Canyon\)](#) 是一款性能强劲，并且体积小巧的高性能迷你主机，搭载全新一代混合架构的第 12 代智能英特尔® 酷睿™ 处理器，并且内置了[英特尔锐炫™ A770M 独立显卡](#)。



### 1.3 搭建训练 PyTorch 模型的开发环境

#### 1.3.1 Windows 版本要求:

在 Windows 上基于英特尔独立显卡训练 PyTorch 模型所依赖的软件包 torch-directml 要求:

- Windows 10 的版本 $\geq$ 1709
- Windows 11 的版本 $\geq$ 21H2

用“Windows logo 键+ R 键”启动“运行”窗口，然后输入命令“winver”可以查得 Windows 版本。



#### 1.3.2 下载并安装最新的英特尔显卡驱动

到 Intel 官网下载并安装最新的英特尔显卡驱动。驱动下载链接: <https://www.intel.cn/content/www/cn/zh/download/726609/intel-arc-iris-xe-graphics-whql-windows.html>。

#### 1.3.3 下载并安装 Anaconda

下载并安装 Python 虚拟环境和软件包管理工具 Anaconda: <https://www.anaconda.com/>

安装完毕后，用下面的命令**创建并激活**虚拟环境 `pt_a770`：

```
conda create --name pt_a770 python=3.9
conda activate pt_a770
```

### 1.3.4 安装 PyTorch

在虚拟环境 `pt_a770` 中安装 PyTorch 1.13。需要注意的是：`torch-directml` 软件包当前只支持 PyTorch 1.13。

```
pip install torch==1.13.0 torchvision==0.14.0 torchaudio==0.13.0
```

### 1.3.5 安装 torch-directml

在虚拟环境 `pt_a770` 中安装 `torch-directml`，这是一个在 Windows 平台上的机器学习训练加速软件包。

```
pip install torch-directml
```

### 1.3.6 安装其它支持软件包

在虚拟环境 `pt_a770` 中安装 `opencv-python`, `numpy`, `pandas`, `tensorboard`, `matplotlib`, `tqdm`, `pyyaml`, `wget` 和 `scipy`，这些是辅助训练模型的软件包。

```
pip install opencv-python numpy pandas tensorboard matplotlib tqdm pyyaml wget scipy
```

到此，在 Windows 平台上用英特尔独立显卡训练 PyTorch 模型的开发环境**配置完毕**。

## 1.4 在英特尔独立显卡上训练 PyTorch ResNet50 模型

请下载 [pt\\_training\\_on\\_A770.py](#) 并运行，该范例代码使用了 PyTorch 自带的 **StanfordCars** 数据集和 **resnet50** 预训练模型。

```
import torch
import torch_directml
import torch.nn as nn
import torch.optim as optim
import torch.nn.functional as F
from torch.utils.data import DataLoader
from torchvision import datasets, models, transforms
import time

# Set device & Hyperparameters
device = torch_directml.device()

num_classes = 196 # The Cars dataset contains 16,185 images of 196 classes of cars
learning_rate = 1e-3
batch_size = 32

# Step1: Load Flower102 dataset
# https://pytorch.org/vision/stable/generated/torchvision.datasets.StanfordCars.html
data_transforms = {
    'train':
        transforms.Compose([
```

```

        transforms.Resize((224,224)),

        transforms.RandomAffine(0, shear=10, scale=(0.8,1.2)),

        transforms.RandomHorizontalFlip(),

        transforms.ToTensor(),

        transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
    ]),

    'test':
    transforms.Compose([

        transforms.Resize((224,224)),

        transforms.ToTensor(),

        transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
    ]),
}

train_dataset = datasets.StanfordCars(root="dataset/", split="train",
transform=data_transforms["train"], download=True)

train_dataloader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)

test_dataset = datasets.StanfordCars(root="dataset/", split='test',
transform=data_transforms["test"], download=True)

test_dataloader = DataLoader(test_dataset, batch_size=batch_size, shuffle=False)

# Step2: Load Pretrained ResNet50 and add FC Layer
model = models.resnet50(weights='DEFAULT').to(device)

for param in model.parameters():
    param.requires_grad = False

model.fc = nn.Sequential(
    nn.Linear(2048, 256),
    nn.ReLU(inplace=True),
    nn.Linear(256, num_classes)).to(device)

model.train()

# Step4: define Loss and optimizer
loss_fn = nn.CrossEntropyLoss()

optimizer = optim.Adam(model.parameters(), lr=learning_rate)

# Step5: Train Network
for epoch in range(3):
    losses=[]

    for batch_idx, (data, targets) in enumerate(train_dataloader):
        start_time = time.time()

        data = data.to(device)

```

```

targets = targets.to(device)

# forward

preds = model(data)

loss = loss_fn(preds, targets)

losses.append(loss)

# backward

optimizer.zero_grad()

loss.backward()

# GSD

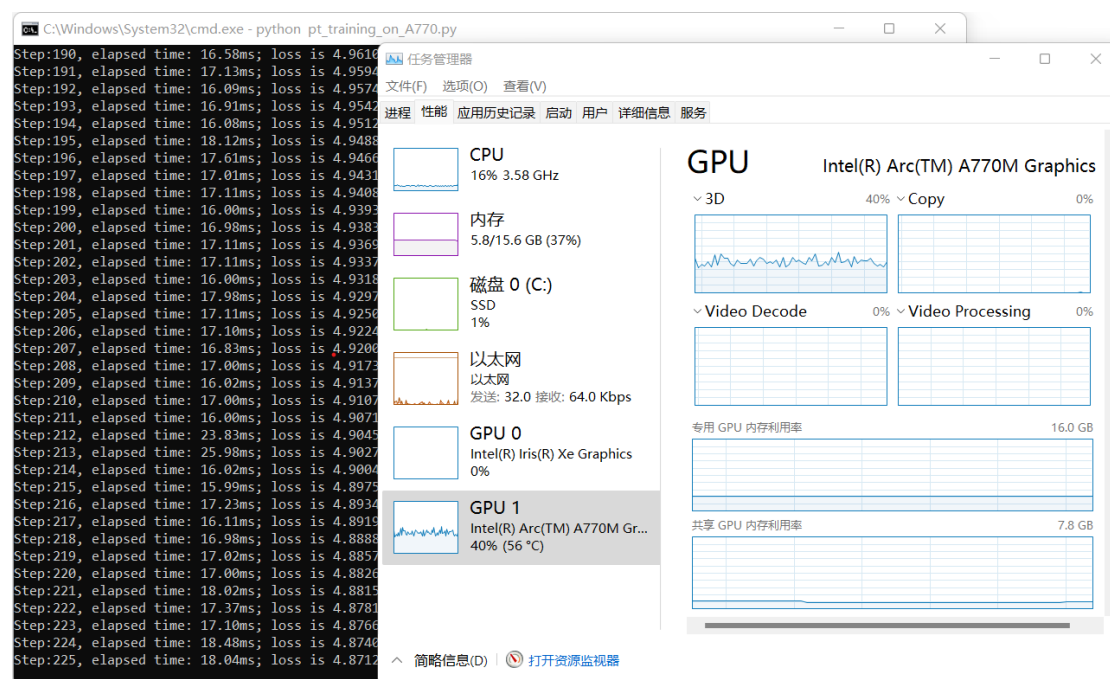
optimizer.step()

time_elapsed = time.time() - start_time

print(f"Step:{batch_idx}, elapsed time: {time_elapsed*1000:0.2f}ms; loss is {sum(losses)/len(losses)}.")

```

运行结果，如下图所示：



## 1.5 总结：

英特尔独立显卡不仅支持 TensorFlow 模型，也支持 PyTorch 模型训练。更多训练范例请参考：<https://github.com/microsoft/DirectML>