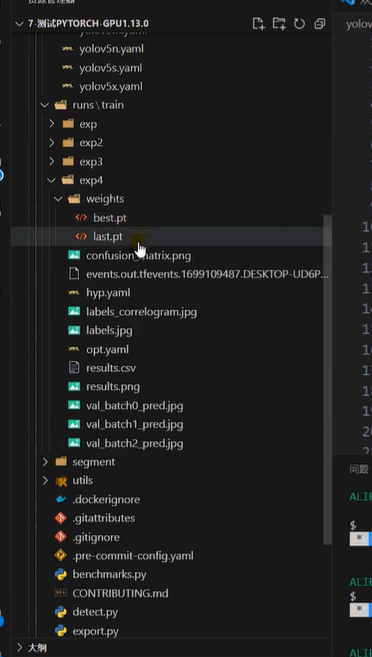
TensorFlow下训练的模型得到的tflite文件和pb文件，分别将这两个文件转化为RKNN文件，然后再调用Python或c/c++接口实现模型的推理。最终在开发板上将模型运行起来。

本讲讲解自训练的PyTorch文件，如何转化为RKNN模型文件，并运行：

/\*

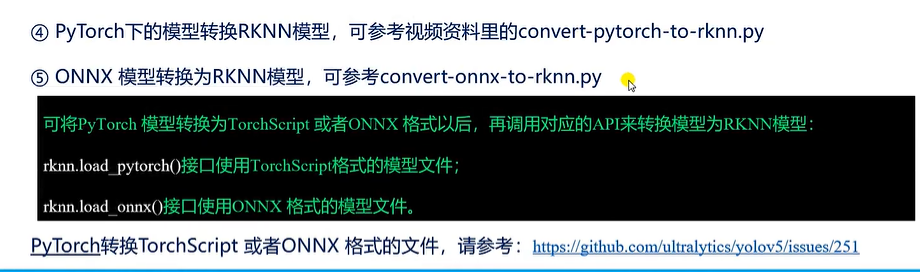
在前面的课程中，我们在Windows下安装了PyTorch-GPU并训练了YOLOv5 v7.0，得到了一个.pt模型文件。其权重保存在了best.pt和last.pt文件中：



本讲就要将这个pt文件转化为RKNN模型文件。

首先可以使用Netron打开这个文件。

转化步骤参考：转化有两条路



1. 将pt文件转化为TorchScript格式的文件，然后再调用rknn\_toolkit里的api,将转化得到的TorchScript格式文件转化为RKNN文件。
2. 将pt文件转化为ONNX格式的文件，然后再通过调用rknn\_toolkit里的api，将ONNX格式的文件转化为RKNN文件，具体的转化方式就是参考了YOLO v5的官网解决方法：

<https://github.com/ultralytics/yolov5/issues/251;>

注意，是在前面我们安装的PyTorch-GPU训练模型的环境下进行该页面的操作，不是在模型转换环境下操作的：

1、安装依赖库；

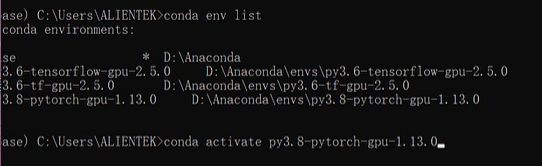
2、将PyTorch训练得到的.pt文件转换为.torchvision格式的文件和.onnx格式的文件。

我们Ubuntu下的环境是用来转换模型的，所以是模型转换环境，而上面链接里的（将pt模型文件转化为TorchScript或ONNX）操作是在模型训练的环境下进行的操作。

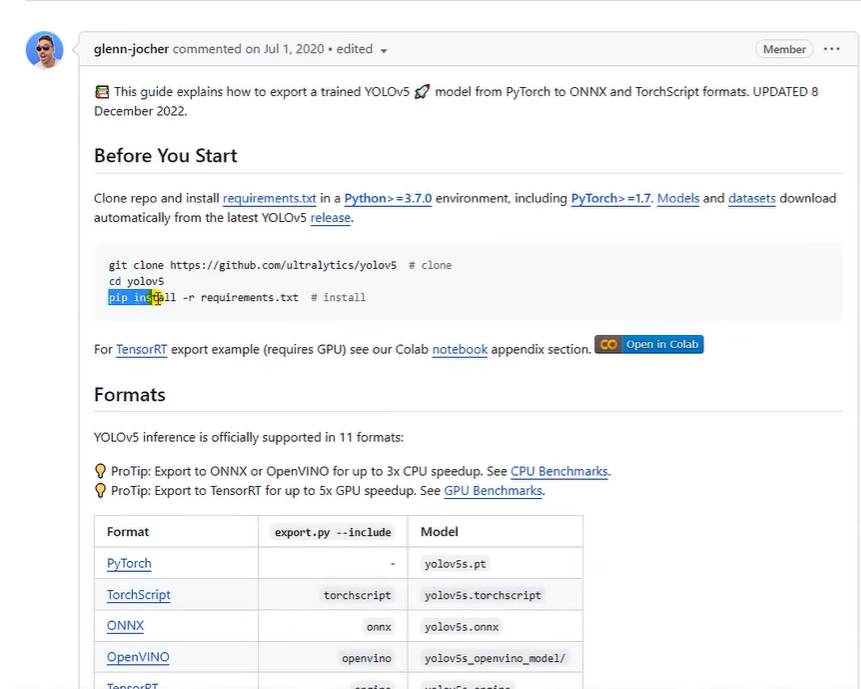
/\*

链接内容解读：

为了能够将PyTorch下的pt文件转化为TorchScript/ONNX格式的文件，我们可以在Windows下搭建一个新的虚拟环境，这个新的虚拟环境前面已经搭建好了。也就是训练模型的环境：

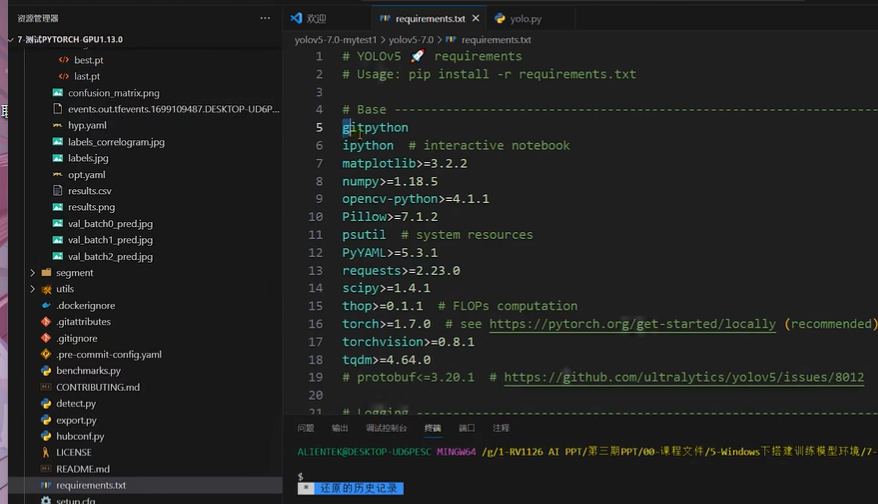


要求：

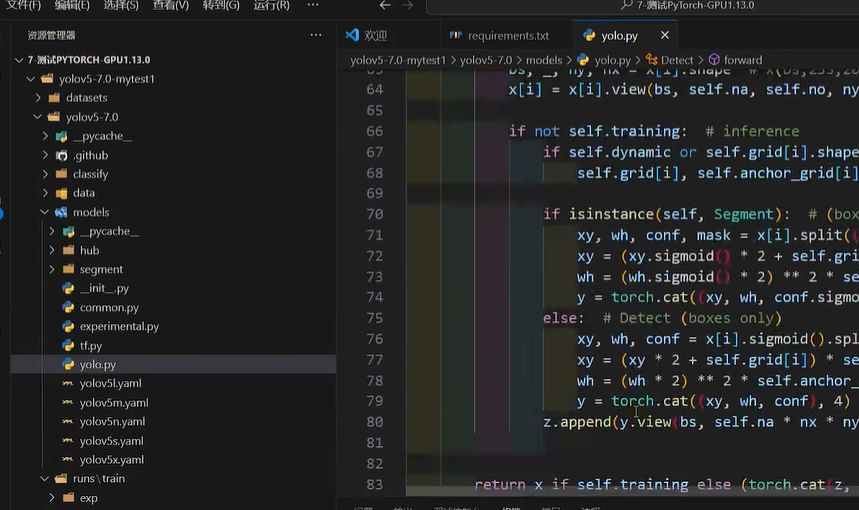


搭建的虚拟环境需要安装requirement.txt里的依赖库。

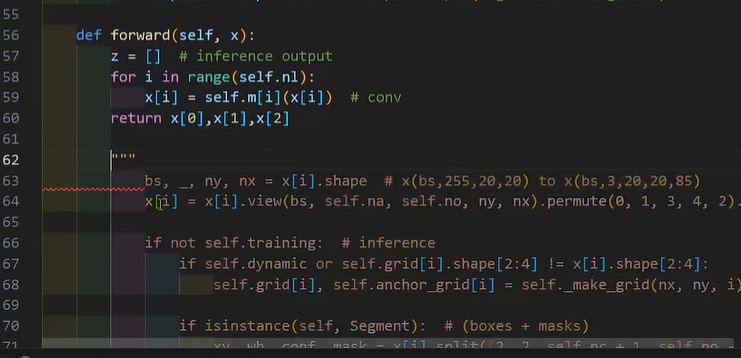
也就是安装YoLoV5的包：在前面我们也安装了。

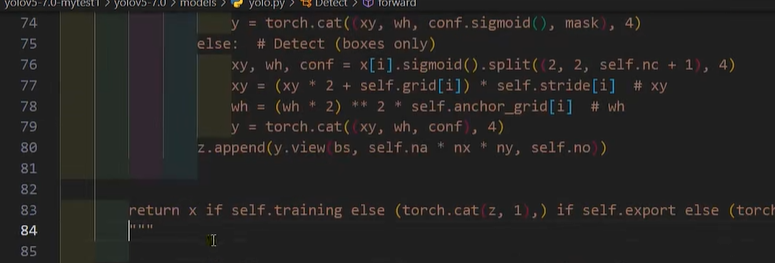


接下来，在转换模型之前，需要先修改一个文件：



我们需要修改里面的forward函数：也就是将return后面的中间那段代码注释掉。并将return的返回值改成3个返回值。

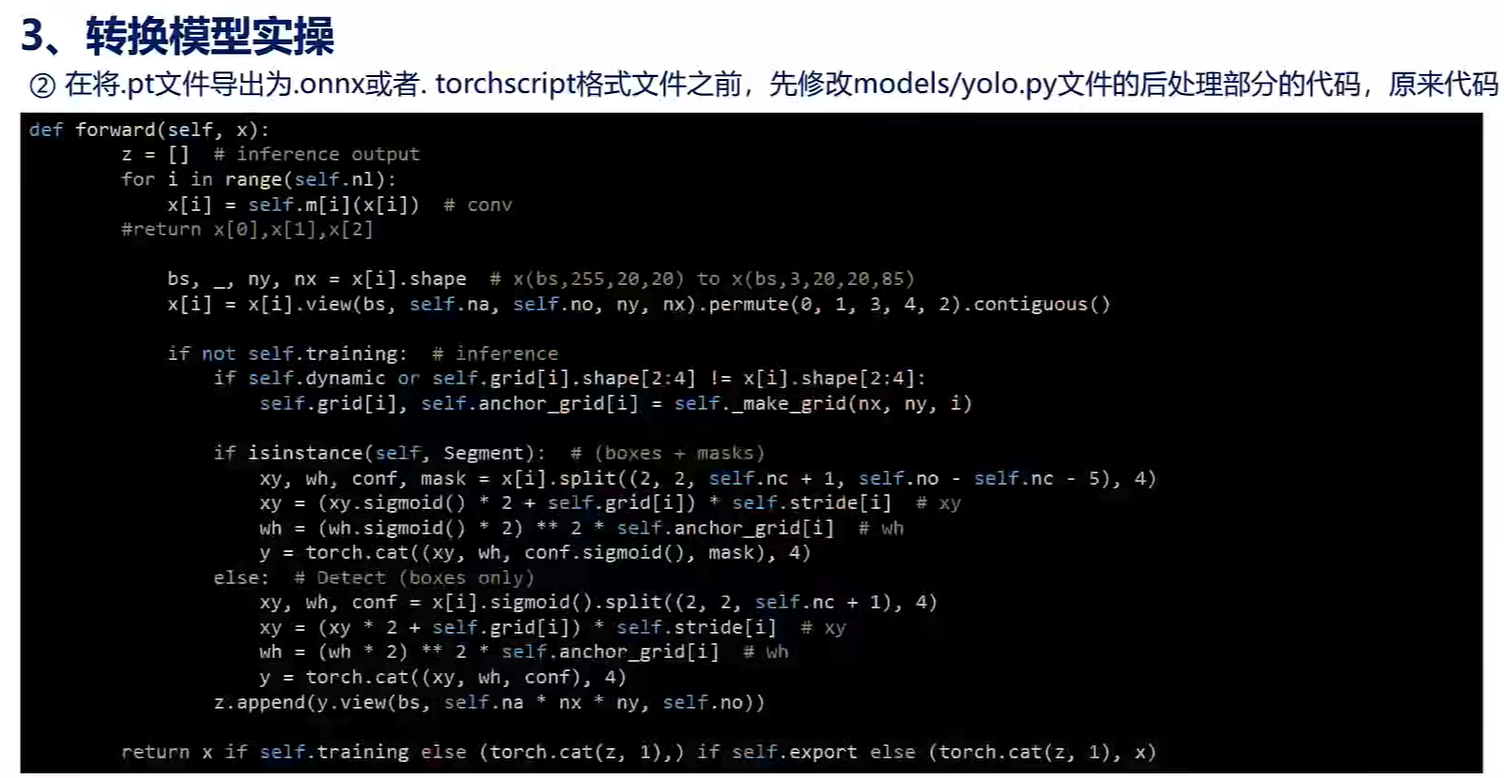


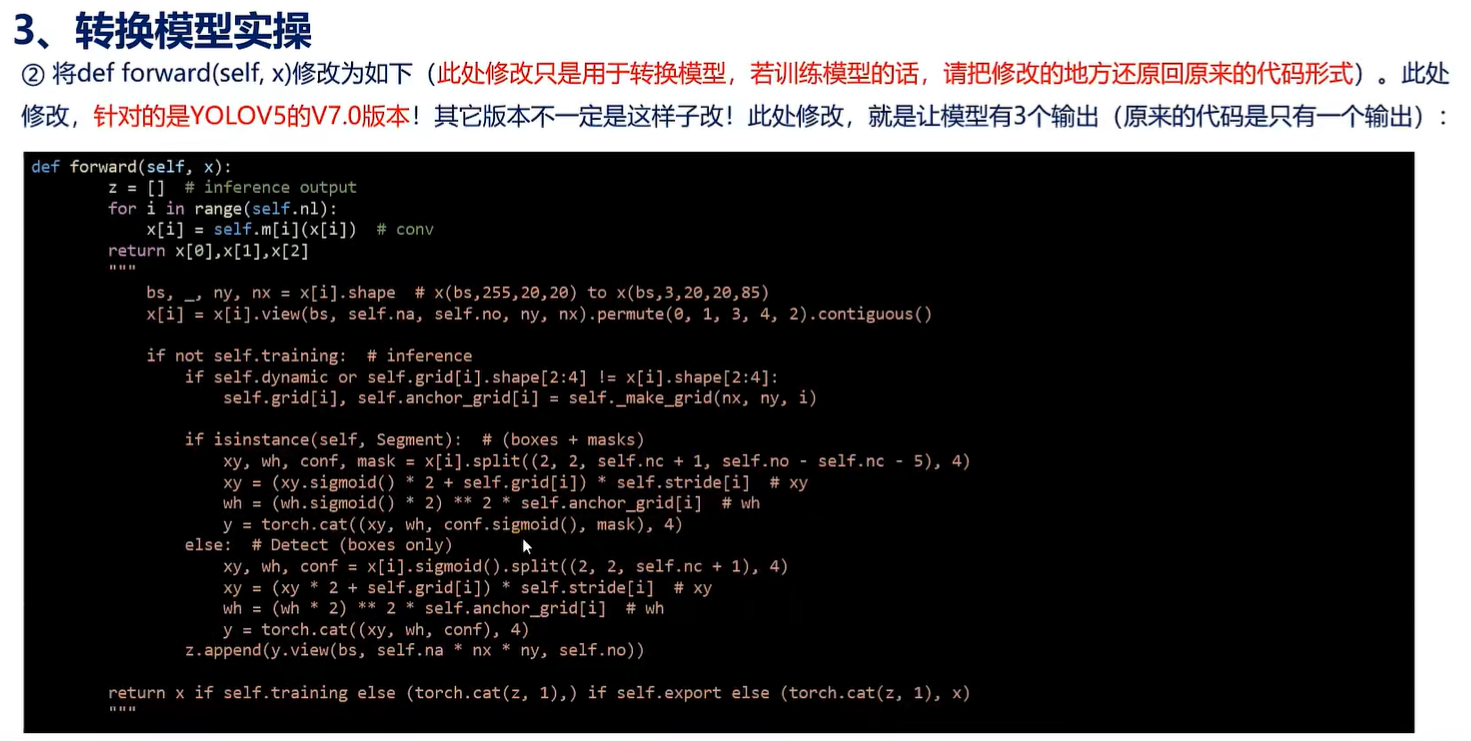


注意这里的修改仅针对这里的YoLoV5\_v7版本。这里就是根据YOLO官网的v5\_v7版本进行修改的。

自己可以训练YOLO模型，可以使用自己的数据集，使用自己的数据集来训练后续会讲。

/\*







//转换为TorchScript格式，得到best.torchscript文件

python export.py --weights ./runs/train/exp/weights/best.pt --img 640 --batch 1 --include torchscript

//转换为ONNX格式，得到best.onnx文件

python export.py --weights ./runs/train/exp/weights/best.pt --img 640 --batch 1 -include onnx

//若需要指定opset，可根据安装的onnx库的版本来调整，如安装的onnx库的版本是1.12.0，后面加上--opset12

python export.py --weights ./runs/train/exp/weights/best.pt --img 640 --batch 1 --include onnx --opset 12

\*/

\*/

\*/