MobileNetV2 numpy项目说明

1. 项目背景  
   本工程的目的是在FPGA平台上实现MobileNetV2神经网络的加速器，使其能够对ImageNet数据集的处理进行加速。但在实现过程中遇到了硬件设计的结果与pytorch模型的输出结果不一致的问题（[【MobileNetV2硬件加速器工程】MobileNet V2量化方法的研究及使用Pytorch quantization包遇到的问题\_fpga的mobilenet-CSDN博客](https://blog.csdn.net/qq_26371477/article/details/119878899)）。本工程使用numpy，根据论文Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference对pytorch模型进行复现，并与pytorch模型的输出进行比对。
2. 项目文件说明

Jpg、JEPG文件为神经网络的输入

Npy、pkl文件为参数M0、M1的数据

PT文件用于保存模型

Imagenet\_classes用于输出判断结果

Python文件中，**conv\_norm\_dp\_pw\_M\_V1\_simplified**和**conv\_norm\_dp\_pw\_M\_V1**定义了基本的卷积运算函数（普通卷积、深度卷积、点卷积）和shortcut函数，两个文件基本上是一样的，只是simplified删去了一些没有用到的内容。

**layer1\_to\_layer53\_M\_V1**和**MobileNetV2\_numpy\_simplified**是代码的核心部分，复现量化MobileNetV2神经网络的推理过程。其中前者比较冗杂，将每一层的参数配置和计算都写出来了，因此可以轻易的针对某一层进行修改。而后者是在前者的基础上简化的版本，使用循环来简化中间的bottleneck部分，代码量少了很多。

**Compare\_each\_layer.py**将pytorch模型中每一层的输入给到numpy模型中得到每一层的输出结果并进行比对，从而得到numpy模型的每一层输出与pytorch模型的差异。

硬件设计的结果与pytorch模型的输出结果不一致的问题已经被解决，具体内容可以查看文档：numpy模型和pytorch模型差别原因.docx

以上是搭建神经网络的部分，除此之外，还有一些用于处理硬件输出结果的程序。

**coe\_generate.py**:位于coe文件夹中，用于将每一层的M1、weight的.p文件转化为对应bram的txt文件用于初始化。Txt文件可以在硬件设计的部分中使用change.bat转换成coe。

**bram\_reshape.py**:用于将vivado运行之后写出的bram.txt文件中的结果重组为每一层的输出结果，并作为npy文件保存到outputs文件夹中。只需要说明每一层的形状和地址范围即可。

**compare\_hardware\_software\_output.py**:位于outputs文件夹中。当运行了MobileNetV2\_numpy\_simplified和bram\_reshape.py这两个文件后，outputs文件夹中有存有神经网络的每一层输出的npy文件。另外，在outputs文件夹中会有每一层输出的xlxs文件，用于验证。此时再运行该代码即可比较每一层的输出，比较结果会存储到comparison\_results文件夹中。

**MobileNetV2\_numpy\_accelerate.py & conv\_accelerate.py:**使用向量化运算对神经网络进行了加速，加速后只需要几秒就可以推理完一张图，而原来的版本需要几分钟。并且使用了round替代floor。

**MobileNetV2\_numpy\_imagenet.py & MobileNetV2\_numpy.py：**在imagenet验证集上测试该模型的图像分类能力，后者是包装为函数的模型。

1. 运行

在3006实验室接上网线后，通过ssh连接到实验室的服务器，配置好相应的环境并上传项目后即可运行。

如果希望在自己的电脑上运行，遇到的问题可以参考文件：pc运行过程.docx

1. 主要负责人

本项目由赵忠宇、陈家宝于2021年实现了软件设计部分和硬件设计的主要模块。谢易达于2025年完善了硬件设计部分，并增加了用于验证结果的软件部分。