**FPGA深度学习加速器工具链说明文档**

1. **工具链职能**

为实现对深度学习模型的加速，FPGA需要将网络模型中的每一层操作拆分出来做计算。本工具链所做的就是为FPGA深度学习加速器提供每一层的数据信息、指令控制信息以及对比结果。

本文档仅介绍工具链如何使用，代码的注释中有更详细的细节。

1. **工具链总体架构**
2. **目录结构**

工具链由四个软件包和三个python文件组成：

main.py:工具链主函数，负责将网络模型按照网络结构进行拆分

transit.Py：分发器，根据每一层网络所做的操作来调用不同的算子

shared\_variable.Py:定义整个工具链的共享变量

conv\_operator软件包：存放conv操作算子

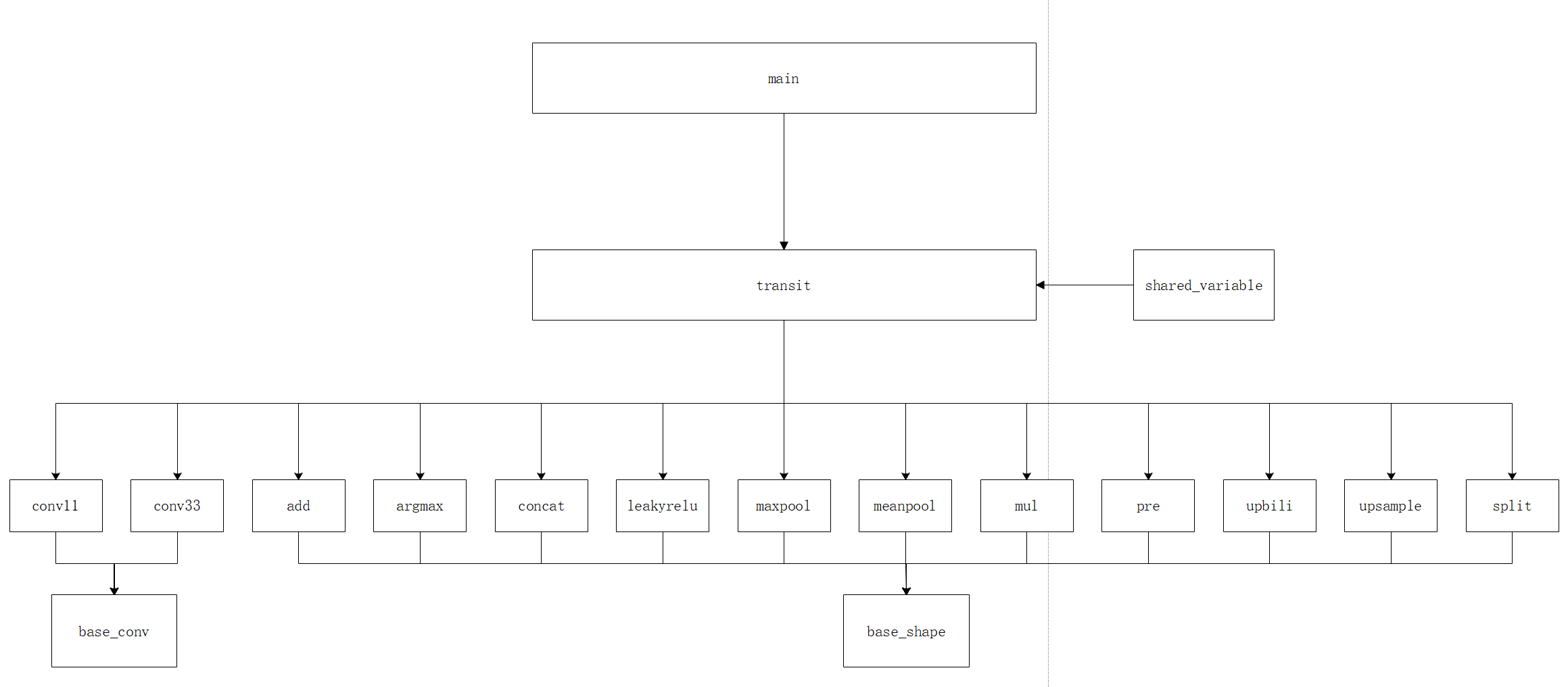
shape\_operator软件包：存放shape操作算子

lib软件包:存放算子操作过程中所需要的方法

utils软件包:存放数据信息、指令控制信息以及对比结果生成后所需要的一些便捷工具

1. **工具链总体设计**

工具链的设计思路如下图所示：



main函数根据网络层数会调用多个transit分发器，所有的transit都共用一个共享变量对象（shared\_variable）。同时transit会根据操作类型选择不同的算子，其中conv11算子和conv33算子继承了base\_conv父类，其余算子继承了base\_shape父类。

1. **主函数编写规范**

主函数编写由两部分组成。首先需要将网络模型的每一层拆分出来，之后将拆分出来的网络输入输出、提取出来的量化参数、本层操作传入transit分发器。

1. **拆分网络模型**
   1. **导入量化好的模型**



shared.model\_path是量化后模型的文件路径，使用torch导入模型

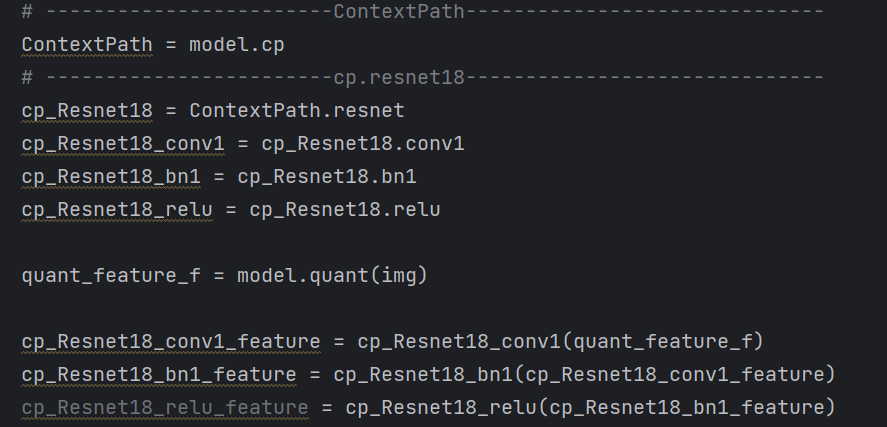
* 1. **导入需要处理的输入图片**



shared.img\_path是输入图片路径，使用picture\_load导入输入图片并做一次预处理。预处理一般是将图片转为灰度图并做一次归一化。

归一化的公式为[(图片数组/255)-数据集图片均值]/数据集图片方差。

* 1. **按照网络结构拆分网络**



以Bisenet网络为例，该网络第一层为3\*3卷积，并且做了特征融合，将conv、bn融合为一层。

先从模型中拆出第一层的三个操作（conv1、bn1、relu），再将输入图片经过量化，最后将量化图片经过conv、bn1、relu处理，得到输出特征图cp\_Resnet18\_relu\_feature。这就拆分出了网络结构的第一层，其他层也类似此操作。

1. **规范调用transit分发器**
   1. **conv操作调用transit分发器**

若为正常conv操作 :



para1='输入层的npy文件前缀名',

para2='本层的npy文件前缀名',

feature=[输入特征图, 输出特征图],

option=['卷积操作名',stride,padding,是否使用激活函数]

其中para参数中的npy文件是量化时提取出来的weight、scale、zp、bias参数，conv操作都有这些参数。

若为分块conv操作：



para1='输入层的npy文件前缀名',

para2='本层的npy文件前缀名',

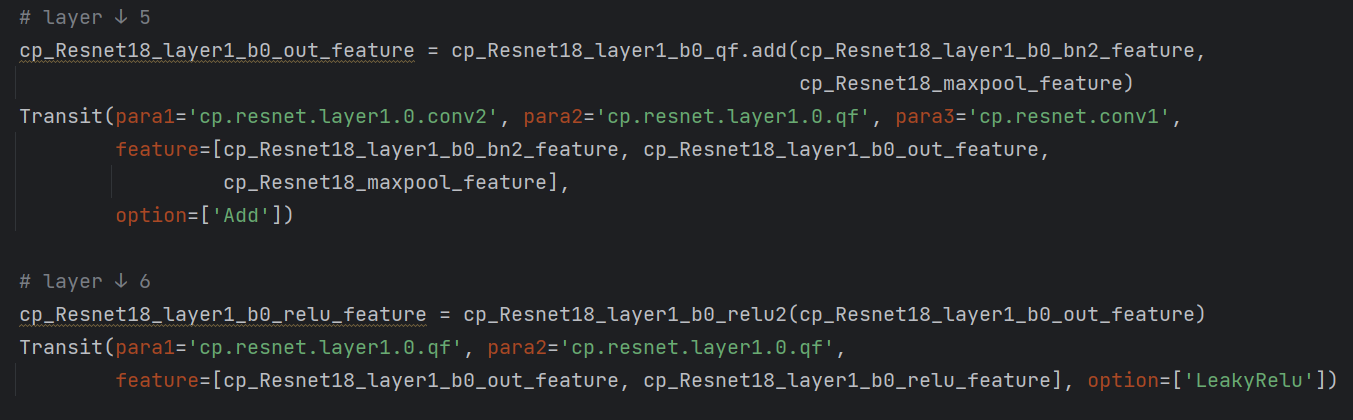
feature=[输入特征图, 输出特征图],

option=['卷积操作名',stride,padding,是否使用激活函数,分块数量]

分块conv操作只比正常conv操作多需要一个分块数量参数，并且根据分块数量生成 2\*分块数量-1 层指令、权重、对比结果(满二叉树)。

该函数本质上还是调用Transit分发器。

* 1. **Shape操作调用Transit分发器**



para1='左输入层的npy文件前缀名',

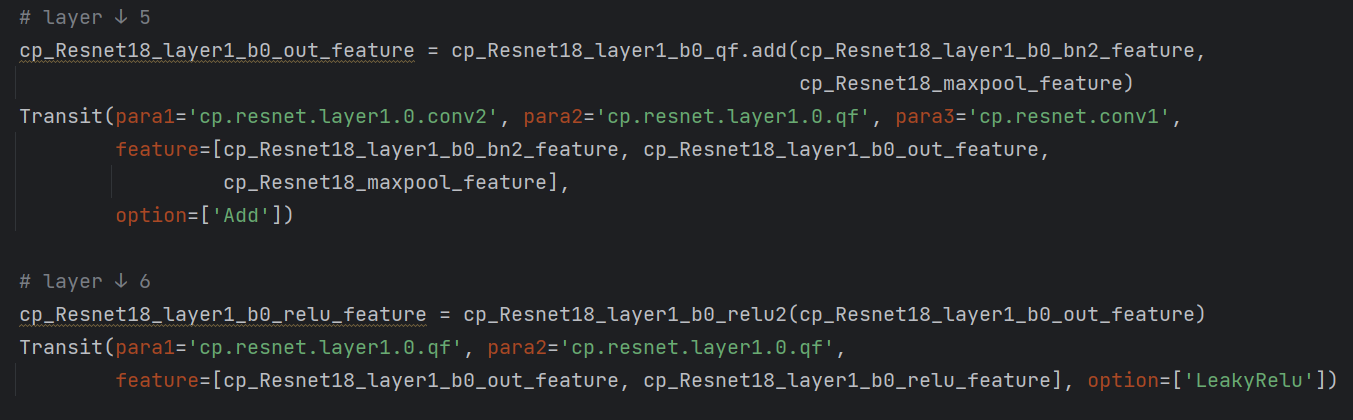
para2='本层的npy文件前缀名',

para3='右输入层的npy文件前缀名',

feature=[左输入特征图,输出特征图,右输入特征图],

option=['shape操作名']

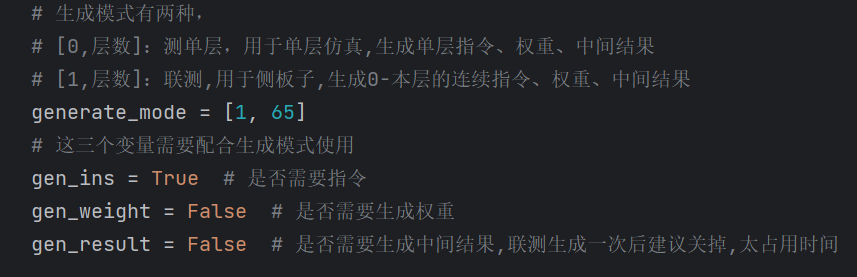
Shape操作调用Transit分发器的特殊形式：



由于leakyRelu操作不需要npy文件提供数据，所以本层的para2用para1代替。

由于leakyRelu操作只需要一个输入，因此para3可省略，feature[3]可省略。

1. **共享变量**
2. **生成模式相关变量**



以图中这种配置方式为例：

generate\_mode:表示生成到65层，并且每一层数据都囊括了之前的数据（1代表联测）。gen\_ins为true，gen\_weight为false，gen\_result为false表示本次只生成指令。

若generate\_mode=[0,64]表示生成到64层，并且每一层数据不包括之前的数据（0代表测单层）。

1. **配置相关变量**



picture\_address:记录了输入图片的起始地址

write\_address:记录了特征图(每一层的计算结果)的起始地址

weight\_address:记录了权重数据的起始地址

paraller:指示了FPGA的通道并行数，一般为8或16

shape\_control:用于维护shape操作的状态码，由于每个项目用到的shape操作各不相同，因此每个项目的shape状态码都不一样。将项目中使用到的n个shape操作从1到n排序，未使用到的shape操作排到n之后可以优化时序。

address\_table:用于记录工具链中的读地址，从而实现地址变换，此字典是自动更新的，不需要填任何信息。