1. 硬件加速神经网络前向推导的原理

本实验使用的基于 CNN 的 MNIST 手写数字识别工作流程为：首先将图片转化为灰度图，然后依次通过卷积层、池化层、卷积层、池化层、全连接层、全连接层，最后即可得到结果。

实验中使用 ZYNQ 异构平台上的 FPGA 对基于 CNN 的 MNIST 手写数字识别进行加速，将原来需要在 CPU 上进行的卷积和池化操作转移到 FPGA 上执行，通过 FPGA 上的专用电路来对卷积和池化操作进行加速，从而达到加速的目的。

2. 加速器设计流程

加速器设计流程为：

- 编写 HLS 代码并通过 TestBench；
- 通过 C synthesis 将 HLS 代码转换为 verilog 代码；
- 将 verilog 代码打包成 IP 核，以便后续在 vivado 工程中调用；
- 在 vivado 中创建工程，导入 IP 核，构建 Block Design，此时得到的工程即为专用的加速卡的硬件描述，然后生成比特流烧录进开发板中，此时 FPGA 即为专用加速卡；

3. 加速器设计方法

加速器设计主要集中在 HLS 加速器代码编写与 vivado Block Design 上。

首先是 HLS 加速器代码编写。需要用 HLS 语言将程序基本逻辑写出来，注意这里缩写的代码是要能够被综合成电路的，所以动态的内存调用、C 的函数库调用、递归指针等高级语法、面向对象的设计都不建议使用。编写完成 HLS 代码后还需要手动设置 IO，由于 HLS 代码会被综合成硬件，所以其调用也不是简单的函数传参，而是需要通过总线获取参数，通过管脚绑定的方式进行 IO 传输。完成编写后即可进行仿真与 IP 核生成。

然后是 vivado 工程中的 Block Design，需要调用刚刚写好的 IP 核，并进行布线连接，将 IP 核连接到已有的硬件电路上。

4. Python 调用 IP 核的方法

首先需要加载 IP 核，将比特流烧录到 FPGA 当中。

然后定义 python 函数，CPU 与 FPGA 的信息交互需要通过 write 与 read 函数实现，上层 python 代码通过这两个函数向 FPGA 收发信息，底层通过总线与内存映射进行参数传递，将数据传送到 FPGA 当中进行计算，并取回计算结果。

最后，根据之前设置与训练好的网络参数调用刚刚定义的 python 函数来实现调用 FPGA 进行计算。