

计算机科学与技术学院计算机系统基础课程实验报告

实验题目：冯·诺依曼计算机指令执行虚拟仿真	学号：202000130198
-----------------------	-----------------

班级：20 级计科 4 班	姓名：隋春雨
---------------	--------

Email: 2018640800@qq.com

实验目的：冯·诺依曼体系结构是当前计算机设计的重要基础，其包含三个基本原则：二进制逻辑、程序存储执行以及计算机由五个部分组成（即运算器、控制器、存储器、输入设备、输出设备）。了解冯·诺依曼体系结构的基本设计原则对理解计算机的结构组织、运行方式具有非常重要的意义。现代人工智能技术不断发展，深度学习应用到如人脸识别、推荐系统、增强现实等诸多领域，计算机为深度学习算法的部署与执行提供了支撑，为了更好地理解这一过程，我们选取了卷积计算这一最常见的深度学习运算模块，结合基于冯·诺依曼体系结构的模型机，在虚拟仿真环境下为学生介绍计算机基本组成和指令执行流程，并引导学生完成卷积计算完整流程，激发学生的实验兴趣，提升学生对计算机系统原理的理解及解决实际问题的能力。

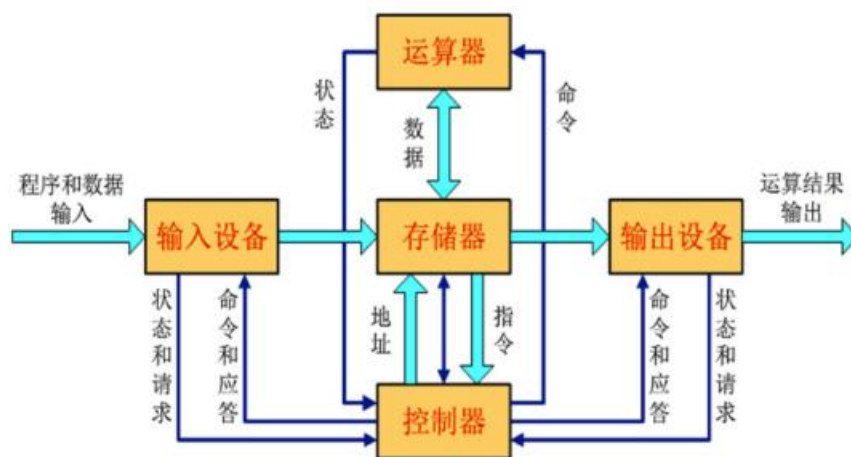
实验软件和硬件环境：

实验软件：火狐浏览器、<http://101.76.229.34:8001/>

硬件环境：windows 10

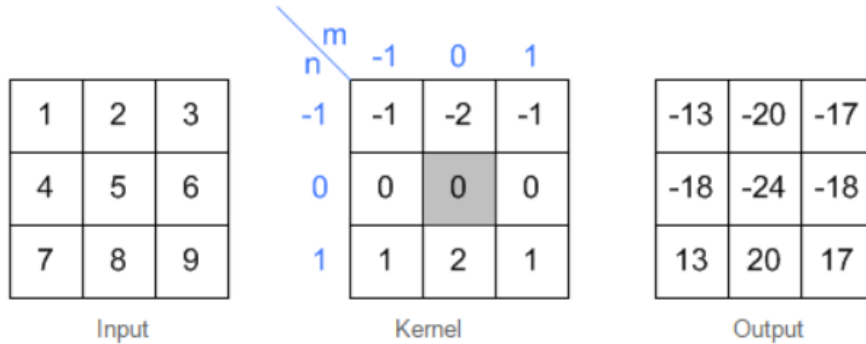
实验原理和方法：

1. 冯诺依曼计算机体系结构为：



2. 卷积计算：

二维卷积计算实际上就是两个 for 循环(也就是被卷积矩阵的一部分和卷积核做点积)，以下面这个为例



卷积在机器学习等领域有着重要的作用，如果对被卷积矩阵做一下该变，就能起到突出边缘的效果，如：

$$g = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

3. 指令体系

- ① Add R1 R2 R3: 将 R1 寄存器的值与 R2 寄存器的值相加放到 R3 寄存器
- ② Multi R1 R2 R3: 将 R1 寄存器的值与 R2 寄存器的值相乘放到 R3 寄存器
- ③ Load address R0: 将 address 地址的值加载到 R0 寄存器中
- ④ Store address R0: 将 R0 寄存器中的值写入到 address 中去
- ⑤ Halt: 终止指令

实验步骤：

(1) 卷积计算

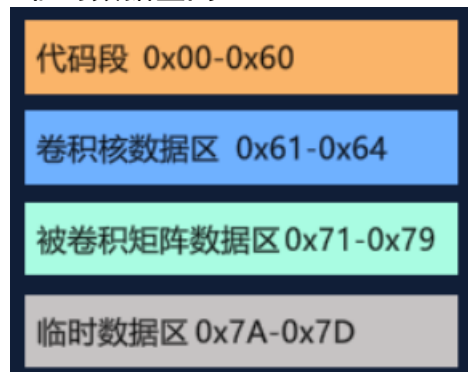
- ① 首先通过搜索引擎查一下卷积的概念和如何使用，了解到卷积主要用在机器学习等领域。对于题目中要求的



实际上就是 c 语言中的两个 for 循环，卷积核中的每个元素与被卷积矩阵中的每个元素对应相乘并累加求和，最后写在卷积计算结果中去。

准备工作：

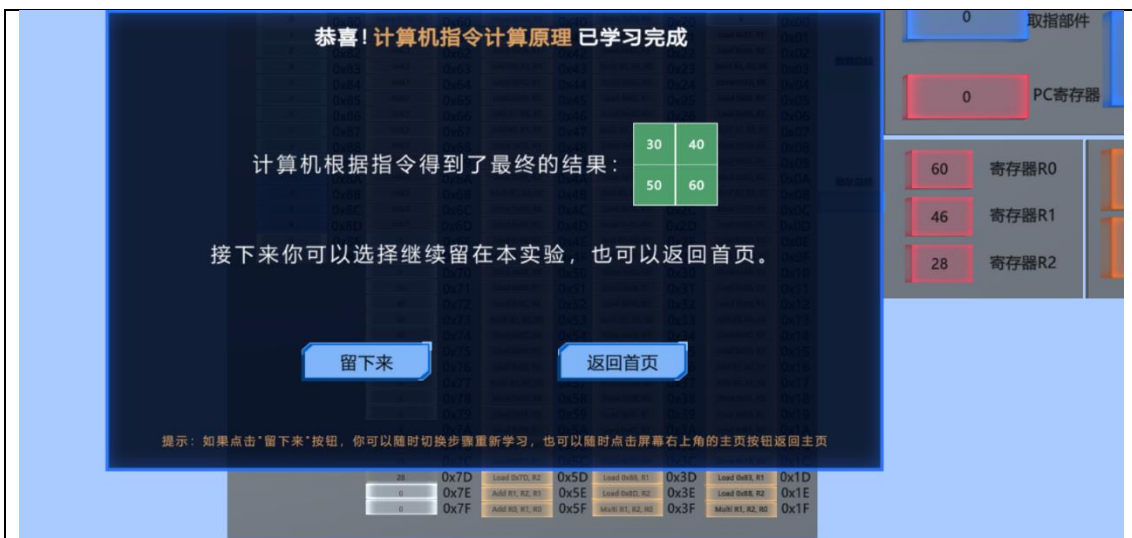
1. 设置代码段区间
2. 设置卷积核数据区区间
3. 设置卷积核的存放方式（行主存放）
4. 设置被卷积矩阵数据区间
5. 设置被卷积矩阵的存放方式（行主存放）
6. 设置临时数据区间



值得注意的是：尽量把数据区的存放位置跟它的值对应起来，这样我们在 Load 的时候也比较好找到对应的元素

执行：

- ② 步骤一：分别取出被卷积矩阵中的相应元素，放到寄存器中去（比如 load 0x71 R1）和（load 0x7A R2），我们可以近似的认为 R1 与 R2 类似于 c 语言中的变量，然后进行相乘操作（multi R1 R2 R0）将结果存入到了 R0 寄存器中，这个时候我们得到了一组对应的值（每轮循环需要产生 4 组这样的值），然后我们将其存入**临时变量区**（store 0x81 R0）。
- ③ 步骤二：上面的步骤进行 4 次，最终在临时变量区（0x81 0x82 0x83 0x84）得到了 4 个值，然后分别将其 load 到寄存器（load 0x81 R0），（load 0x82 R1），我们进行 add 操作（add R0 R1 R0）将 R0 与 R1 相加并将结果存放到 R0 寄存器中
- ④ 步骤三：再次进行 load 与 add 操作（load 0x83 R1）（load 0x84 R2），add 操作：（add R1 R2 R1），将结果保存到 R1 寄存器中去。
- ⑤ 步骤四：将 R0 与 R1 寄存器的值进行相加，保存到 R0 寄存器（add R0 R1 R0），最后将这轮循环的值回写。
- ⑥ 步骤五：以上步骤循环 4 次，即可得到对应的结果，将结果保存到相应区域，最后结尾用 HALT 指令结束，让 PC 寄存器的值为第一条指令的地址，开始执行，得到正确结果。结果如下：



(2) 考核:

☐ A. 多指令流单数据流 ☒ B. 按地址访问并顺序执行指令

☐ C. 堆栈操作 ☐ D. 存储器按内部选择地址

3. 下列不属于标准冯·诺依曼计算机体系结构部件的是_____。

☒ A. 寄存器 ☐ B. 控制器

☐ C. 运算器 ☐ D. 输入与输出设备

4. 计算机中的执行过程是_____。

①取指令 ②执行指令 ③指令译码

☐ A. ①②③ ☒ B. ①③②

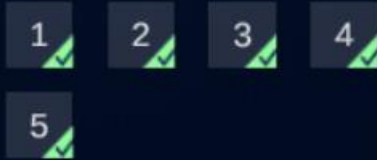
☐ C. ②③① ☐ D. ②①③

5. 从ENIAC到当前最先进的计算机, 冯·诺依曼体系结构始终占有重要地位。冯诺依曼体系结构的核心内容是_____。

☐ A. 采用开关电路 ☐ B. 采用半导体器件

☒ C. 采用左传程序和程序控 ☐ D. 采用键盘输入

答题结果：



冯诺依曼计算机的核心就是整体逻辑上是顺序执行，CPU 永远朝 PC 要下一条指令的位置。

结论分析与体会：

1. PC 寄存器的作用：指向下一条要执行的指令的地址. CPU 每次都是直接朝 PC 要指令。
2. 明白了冯诺依曼各个部件的功能，与各个部件之间的联系：如 CPU 永远朝 PC 要下一条指令的位置
3. 寄存器类似于 C 语言中的变量，我们每次要对被卷积矩阵和卷积核做运算时候，都应该将它们 load 到寄存器中做运算。同时由于寄存器数目的限制（本实验只有 3 个），我们应该将对应的计算结果保存临时数据区，每轮循环结束后，将临时数据区的值 load 到寄存器中进行运算，得到结果回写
4. 最后输入保存的结果地址的时候一定要区分大小写！否则会判错
5. 实验平台用的指令体系结果与 MIPS 的指令体系有细微的区别，需要注意
6. 面对不熟悉的知识，我们应该去搜索相关的资料。比如卷积是课本上没有的，但是却是机器学习中很重要的一个概念，比如用在输出图像边缘强化等领域。一维卷积的实质就是将一个函数反转，再滑动叠加。了解一些实际的应用能够更好的帮我们理解实验。输出图像边缘强化用到的就是二维的卷

$$g = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

积，需要对卷积核进行处理，比如：

就能突出边缘。