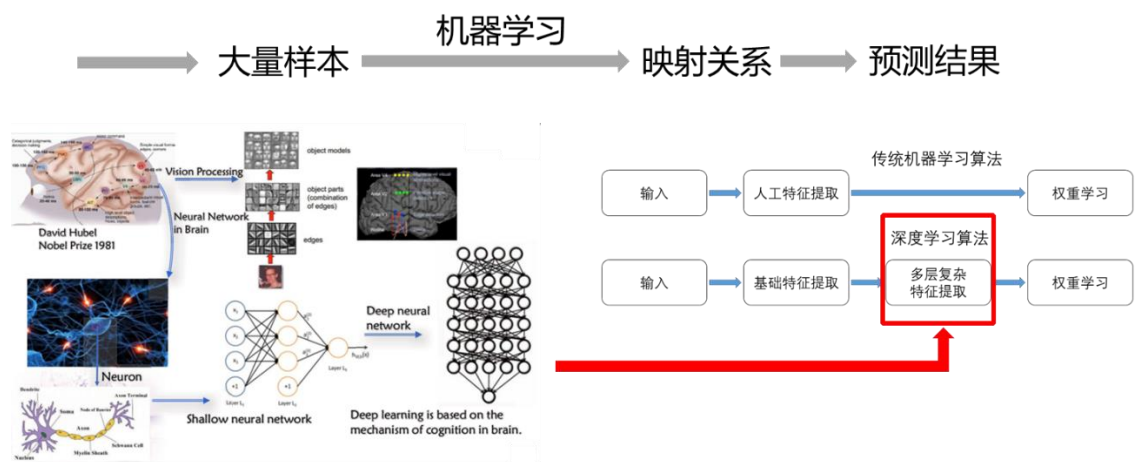


Project 说明

一、相关背景

人工智能的再次兴起得益于机器学习算法的实现。传统的机器学习算法常常需要使用人工的方法从输入数据中筛选出对输出结果有影响的因素，每一个因素被称为一个特征（feature），特征提取并不是一件简单的事情，需要深度学习算法得以实现。深度学习属于机器学习的一个分支，它解决的核心问题之一就是自动提取简单特征，并将简单特征组合成更加复杂的特征，再利用这些组合特征来解决问题。图 1 表示了机器学习的基本概念以及传统机器学习算法和深度学习算法的区别。



机器学习这门学科所关注的问题是计算机程序如何随着经验积累自动提高性能（更准确的映射关系）。
-Tom Mitchell 《Machine Learning》

图 1 机器学习，传统机器学习算法和深度学习算法的区别

近年来，卷积神经网络（CNN）成为了深度神经网络的主流研究对象。CNN 通过对一张图进行不断的卷积，从边缘的一些局部特征逐渐提炼出这张图的全局特征。例如，对于一辆车的图片，浅层网络可能提取出车身轮廓这类信息，中层网络可能就提取出轮胎这类的局部特征，深层网络就能提取出整个车的全貌。下图 2 表示了一个较简单的手写体识别的 CNN 模型，LeNet5，它包含 2 层卷积（Convolutions），2 层池化（Subsampling）以及 3 层全连接（Full connection）。

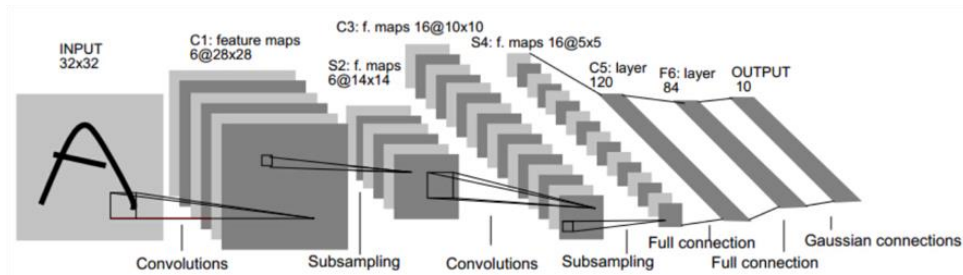


图 2 LeNet5 网络结构

卷积运算实际上是大量的乘加运算，图 3 所示即为卷积运算，紫色的小长方体为一个卷积核，其空间的每个位置存储着一个权重数据，白色长方体为输入图像数据，其空间位置上存储着灰度值。需要注意的是，输入图像是 3D 的，举个例子来说，一幅彩图由 R, G, B 三种颜色构成，组成这幅图的数据就是 3 组矩阵，也就是 3D 的。卷积核与输入图像进行卷积运算时，对应空间位置上的数值相乘，再将所有数据加起来，便得到了一次卷积结果。

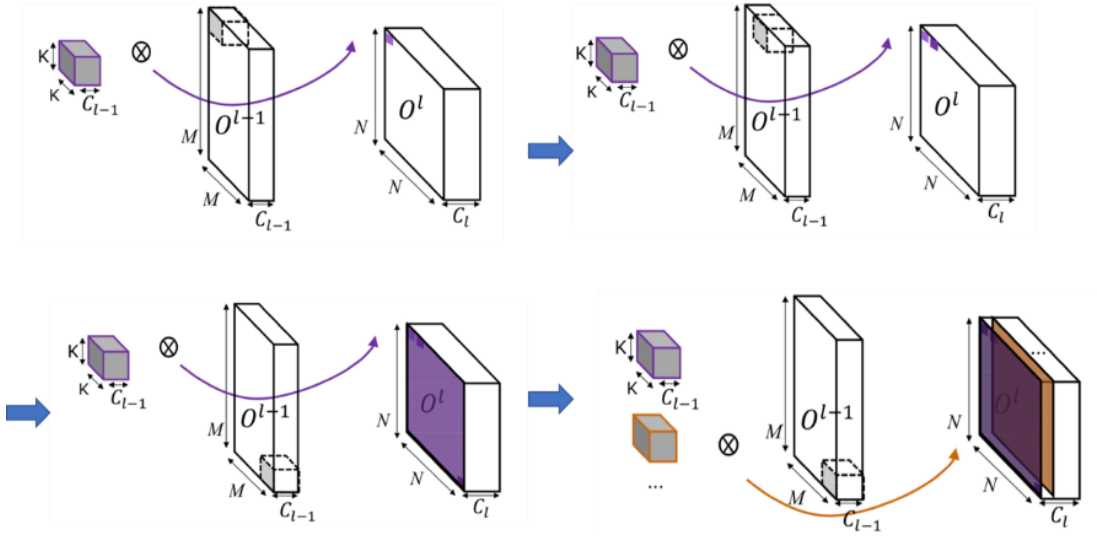


图 3 卷积运算

池化运算是一个简单的降采样过程，通常采用最大值或均值进行降采样。如图 4 所示。全连接层的计算为向量的乘法计算。由于卷积运算的空间并行度比较高，输入数据存在重复的情况，采用 CPU, GPU 进行运算速度、能效都不高，因此需要设计 CNN 的硬件加速电路，这也是目前的一个研究热点。

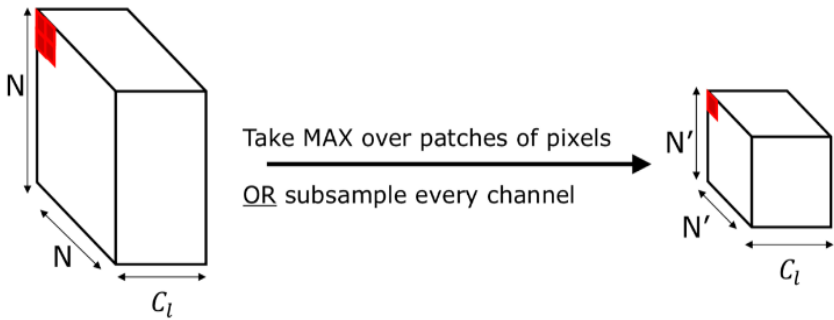


图 4 池化运算

二、题目要求

要求针对卷积运算，设计一个卷积运算阵列，能够实现对 LeNet5 模型的第一层卷积运算。比如，LeNet5 模型的第一层输入图片大小为 28x28，所需要进行的卷积运算为 5x5，卷积核移动步长为 1。每一次卷积即为 25 个乘法结果相加，移动步长为 1 时，完成整张图片的卷积总共需要进行 24x24 (24=28-5+1) 次卷积。如下图 5 表示了整个卷积过程。

为了简化，本 project 采用的卷积阵列是 3*3，输入阵列是 7*7。完成第一层卷积运算。

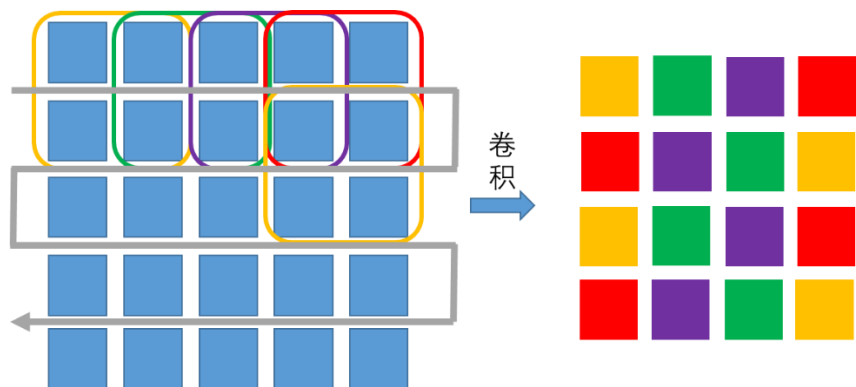


图 5 卷积过程示意图

● 计算阵列电路图

图 6 为一个 5x5 卷积运算阵列，阵列每一次运算即可完成 25 次乘加运算，即完成一次卷积运算。注：本 project 的卷积运算阵列大小是 3*3.

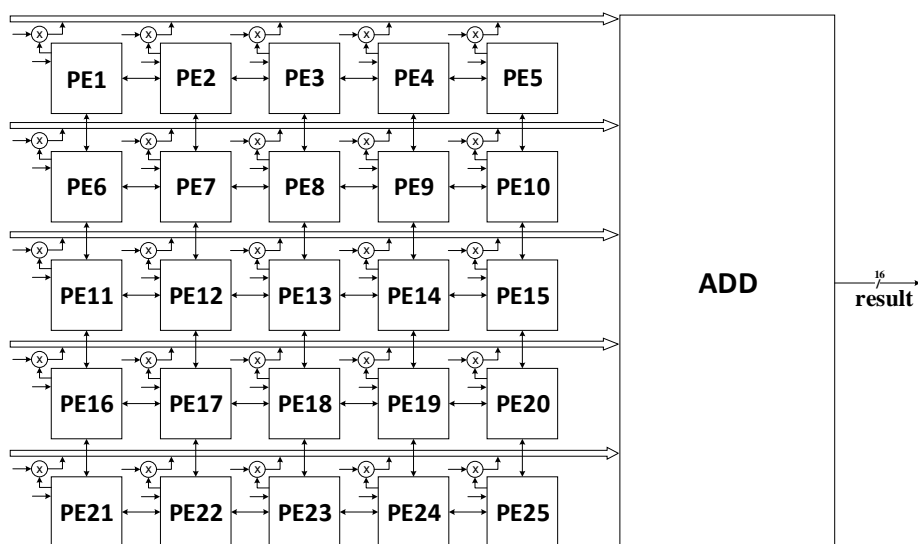


图 6 5x5 卷积运算阵列

● 优化思路

1. **优化乘法运算。**每一个 PE 单元带一个乘法器，能够实现一个 16 位的有符号数乘法运算。可以看出该计算过程中，实际上最慢的部分就是乘法运算，因此需要设计延迟较低的乘法器来使得整个运算速度加快。**可考虑使用基于 Booth 编码乘法器。**本次设计中输入数据为有符号 16 位（16 位定点数 15~0，15 位为符号位，14~12 为整数位，11~0 位小数位），因此乘法器输出结果本应该为 32 位，在这里需要截断，使得乘法结果依然为 16 位，请参考后面具体的截断方式。
2. **PE 内置存储单元。**从图 5 中可以看出邻近的两次卷积计算，实际上输入图片的数

据部分是重复的，因此可以**给 PE 内置寄存器或触发器**，使得 PE 可以缓存输入数据，并且可以**将缓存的输入数据传递给邻近的 PE**，减少从片外读取输入数据的次数，从而降低功耗，请**思考数据的传递过程**，从而**给出数据复用的方法**。

3. **提高流水线的效率**。设置合理的流水线方式，能够大大加快运算速度。
4. 可以通过调研，考虑其他更好的优化方式。

三、提交内容、时间及其它说明

- 1、**根据附件中给出的卷积阵列数据（3*3）和输入数据（7*7），计算出卷积之后的输出数据（5*5 阵列），所有数据转换成 16 位的二进制数。**
- 2、设计时综合考虑速度，面积，功耗等性能指标。
- 3、完成电路的前端设计(电路设计、Verilog 代码编写及 NC 仿真，DC 综合及综合后仿真) 及后端版图设计（布局布线）。
- 4、独立完成。
- 5、提交内容和时间：
 - 1) 设计报告：12 月 28 号之前网上提交。
 - 2) 结果展示：12 月 29 号下午 2:00~5:00，地点：集成电路学院上机机房。

Project 的结果展示注重学生分析问题解决问题的过程!!!

四、输入数据

输入数据有两个，一个是权重 weight（3*3），如表 1 所示。

表 1: weight 数据

FD01	01CD	FEFC
FFC4	FF30	FDB4
FEDD	FFBC	FCFD

第二个数据是输入图片的数据 pic(7*7)，如表 2 所示。

表 2: pic 数据

0000	0000	0070	0757	0F1F	1000	0FDF
01A1	0AAA	0BBB	0FCF	0FCF	0FDF	0FCF
0C8C	0FDF	0FCF	0FCF	0D6D	0A8A	0A8A
0E1E	0FDF	0FCF	0FCF	0282	0000	0000
0E2E	1000	0FDF	0A8A	0000	0000	0000

0969	0FDF	0FCF	0A8A	0000	0000	0000
0717	0FDF	0FCF	0A8A	0000	0000	0000

说明：

- 数据是以 16 位二进制**补码**的形式给出的，为了方便表述，写成 4 位十六进制。
- 16 位二进制与 16 位二进制相乘的结果是 32 位二进制，必须进行数据截断。16 进制最高位为符号位，有 3 位整数位，和 12 位小数位，所以对乘法运算所得结果的数据截断方式如下：

1101 0011 . 1100 0101 0111 1110 0101 0010 → **1011 . 1100 0101 0111**

相乘得到的数据首先进行截断，而后进行相加，并且相加过程中溢出位舍去。12 位小数相乘结果有 24 位小数，截取小数点后面的 12 位作为小数部分（绿色部分），蓝色表示符号位，小数点前面的 3 位作为整数部分（红色部分），其他部分直接删除，最终得到 16 位二进制。（卷积的方式是：strides=[1, 1, 1, 1], padding=**VALID**，感兴趣的同学可以参考 conv2 函数。）