**关键是,**

**我觉得就越大越好啊?**

**得到16\*16,**

**一些查到的资料:**

[**https://www.sohu.com/a/142237570\_505803 这个分析了 矩阵大小512\*512**](https://www.sohu.com/a/142237570_505803%20这个分析了%20矩阵大小512*512)**时没有效率. 好像挺有用的.**

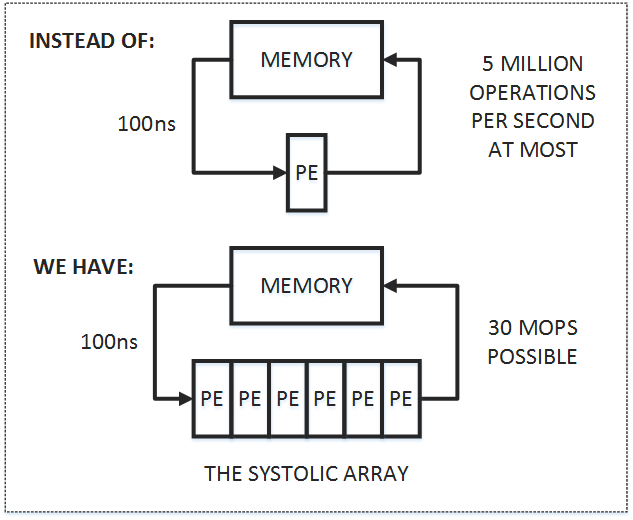
**1.延迟会随着阵列边长成线性增长**

**2. 加载权重的开销, 比如512 要有4个, 加载部分占用的总面积就比 256\*256 大.**

**3.** **输入带宽需求随阵列边长平方增加**

**4.** **阵列的利用率随着阵列的边长的增长而减小**

**systolic array**脉动矩阵中，每个单元具备一个乘累加器，一个寄存器用来缓存权值，通过两个方向脉动传输数据，一个方向（列方向）传输输入数据，其输入数据来自于vector uinit里的vector register，数据沿着脉动矩阵的列方向逐级下传，另一个方向（行方向）传输累加结果.



正如上图的下半部分所描述的，第一个数据首先进入第一个PE，经过处理以后被传递到下一个PE，同时第二个数据进入第一个PE。以此类推，当第一个数据到达最后一个PE，它已经被处理了多次。所以，脉动架构实际上是多次重用了输入数据。因此，它可以在**消耗较小的memory带宽的情况下实现较高的运算吞吐率**。当然，脉动架构还有其它一些好处，比如模块化的设计容易扩展，简单和规则的数据和控制流程，使用简单并且均匀的单元（cell），避免了全局广播和扇入（fan-in），以及快速的响应时间（可能？）等等。

Google选择复杂指令集(CISC)作为TPU指令集的基础，为了控制MUX、UB和AU进行计算，Google定义了十几个专门为神经网络计算而设计的高级指令。同时TPU又遵循极简设计方案，专门为深度学习运算加速服务，是一个单线程芯片，不需要考虑缓存、分支预测、多道处理等问题。

总结起来，脉动架构有几个特征：1. 由多个同构的PE构成，可以是一维或二维，串行、阵列或树的结构（现在我们看到的更多的是阵列形式）；2. PE功能相对简单，系统通过实现大量PE并行来提高运算的效率；3. PE只能向相邻的PE发送数据（在一些二维结构中，也可能有对角线方向的数据通道）。数据采用流水线的方式向“下游”流动，直到流出最后的PE。

到这里不难看出，脉动架构是一种很特殊的设计，结构简单，实现成本低。但它灵活性较差，只适合特定运算。

这里值得注意的是，要实现正确的矩阵运算，数据进入脉动阵列需要调整好形式，并且按照一定的顺序。这就需要对原始的矩阵进行一些reformat，这也增加了额外的操作。脉动架构适合的其它运算，主要包括信号和图像处理（signal and image processing），矩阵算术（matrix arithmetic）和一些非数值型应用（non-numeric application）。