人工智能的模拟存内计算架构

Patrick Bowen*Neurophos, 212 W Main St. #301, Durham, North Carolina 27701, USA and Center for Metamaterials and Integrated Plasmonics and Department of Electrical and Computer Engineering, Duke University, P.O. Box 90291, Durham, North Carolina 27708, USA

Guy Regev† 和 Nir Regev‡AlephZero, 5141 Beeman Ave. Valley Village, CA 91607, USA andDepartment of Electrical and Computer Engineering, Ben-Gurion University of the Negev, David Ben-Gurion Blvd. 1, Be'er Sheva, Israel

Bruno Umbria Pedroni§AlephZero, 5141 Beeman Ave., Valley Village, CA 91607, USA and Department of Bioengineering, UC San Diego, 9500 Gilman Dr., La Jolla, CA 92093, USA

Edward Hanson¶ 和 Yiran Chen**Duke University, Electrical and Computer Engineering Department, Science Dr, Durham, NC 27710(日期：2023年2月14日)

本文分析了数字与模拟存内计算架构能效的基本极限，并在机器推理的具体场景中，将其性能与单指令、单数据（标量）机器进行了比较。分析重点在于计算能效如何随待执行计算的规模、算术强度和位精度而变化。研究表明，随着问题规模和处理器规模不断扩大，模拟存内计算架构的能效可以趋近于任意高。

arXiv:2302.06417v1 [cs.AR] 13 Jan 2023

I. 引言

本研究致力于最大限度地降低神经网络求值所需的能量，尤其关注构成绝大部分计算量的线性层。描述卷积神经网络层的线性算子通常具有三个特点：稀疏、维度高、算术强度高。算术强度定义为基本运算（即乘法和加法）的数量与读写字节数之比。本文表明，对于维度和算术强度都较高的算子，模拟存内计算设备能够大幅降低算子求值所需的能量，相较之下，冯·诺依曼机需要消耗更多能量。此外，模拟处理器能效的提升幅度与处理器规模有关。

在经典的冯·诺依曼机中，计算一个算子所需的能量可分为两部分：存储器访问能耗和计算能耗。在典型的中央处理器中，视工作负载而定，这两部分的能耗可能处于同一数量级。与内存访问相关的能耗很容易超过计算能耗，尤其是在用于计算神经网络推理中常见的序列化大型线性算子时。本文旨在寻找高层次体系结构，以将神经网络算法的能耗降低几个数量级；这需要同时解决存储器访问能耗和计算能耗问题。我们在此表明，对于算术强度较高的算子/算法，存内计算加速器架构能够降低存储器访问能耗；而针对特定类别的线性算子进行专门设计的模拟处理器/加速器，则能够降低计算能耗。原则上，兼具存内计算优势且本质上为模拟式的处理器体系结构，可将整体计算能耗降低几个数量级；具体降低幅度取决于待执行算法的规模和算术强度，以及模拟处理器对特定算子集的专门处理能力。

存内计算架构最初旨在加速可并行化且应用于大型数据集的算法处理。最早的实例之一可追溯到 20 世纪 60 年代的 Westinghouse Solomon 项目。该项目的目标是通过将单条指令应用于大型算术逻辑单元（ALU）阵列，使计算机速度提升至 1 十亿次浮点运算每秒。这或许是以下几个密切相关概念的首个实例：单指令多数据（SIMD）机器、向量/阵列

* ptbowen@neurophos.com
† guy@alephzero.ai
‡ nir@alephzero.ai§ bruno@alephzero.ai
¶ edward.t.hanson@duke.edu
** yiran.chen@duke.edu

处理器、脉动阵列以及存内/近存计算设备。

如今，利用并行硬件挖掘高算术强度算法中的并行性，仍是沿时间维度加速计算的一种众所周知的技术。不过，近来向量/阵列处理器已被用于降低计算能耗，而非像最初那样用于缩短计算时间；其实现方式是减少内存访问所带来的能耗。谷歌的 TPU 就是一个很好的例子：它采用由数字处理单元构成的脉动阵列，作为近存计算器件 [1, 2]。在第三节中，我们将解释，对于算术强度较高的线性算子，存内计算器件如何降低内存访问能耗。

另一方面，近来有人提出采用模拟计算来降低计算能耗，同样针对大型线性运算。在第四节中，我们将介绍一种模拟计算通用模型，重点探讨能耗如何随问题规模和位精度变化，并说明通过使用专门实现特定类别算子的模拟处理器，计算能耗可降低数个数量级。可重构模拟处理器本质上就是存内计算器件，因此我们将说明，对于特定算子，这类处理器能够将整体计算能耗降低数个数量级。

II. 中央处理器能耗

我们首先计算执行乘加运算（MAC）的计算机的能效。乘加运算是深度学习中所用线性算子的核心。执行线性运算所需的总能量可分解为存储器访问能耗和计算能耗：

$$E_{tot} = N_m e_m + N_{op} e_{op}, \tag{1}$$

其中， N_m 表示内存访问次数， e_m 表示每次访问的平均能量， N_{op} 表示计算整个算子所需的运算次数， e_{op} 表示每次运算（例如加法、乘法等）的平均能量。我们将计算效率定义为计算机每单位能量所执行的运算次数：

$$\eta \equiv N_{op}/E_{tot} = \frac{1}{(N_m/N_{op})e_m + e_{op}}. \tag{2}$$

在 Flynn 分类法中，采用单指令单数据（SISD）体系结构且内存层次结构扁平的简单中央处理器中，每执行一次运算，都会从内存中读取当前部分和、算子权重以及输入激活值。对这三个值进行运算后，再将结果写回内存。因此，无论权重或激活值的实际大小如何，每次运算的内存访问次数始终为 4 次（即三次读取和一次写入），而计算运算次数（乘法和加法）为 2 次。这使得 $N_m = 2N_{op}$，计算效率为

$$\eta = \frac{1}{2e_m + e_{op}}. \tag{3}$$

在现代互补金属氧化物半导体器件中， e_m 和 e_{op} 的量级均为 1 pJ， [3], 如表 IV 所示。这使大多数传统体系结构的计算效率大致受限于 0.1–1 每瓦万亿次运算，这与当前最先进的性能水平一致 [4]。

III. 通过存内计算最大限度地降低存储器访问能耗

单指令流单数据流机器的一大缺点是，在计算大型算子的过程中，可能会多次访问同一个存储器元素，从而浪费存储器访问能耗。这最终体现在比率 $N_{op}/N_m = 1/2$ 上，而该比率受单指令流单数据流机器的本质所决定。另一种设想是，寻找一种假想的体系结构，以某种能效最优的方式进行设计，使所有输入在计算过程中都只从存储器中读取一次，所有输出也都只写入存储器一次。若能做到这一点，就意味着计算该线性算子所需的总访问能耗达到了最低。换言之， N_m 将达到最小值，而比率 N_{op}/N_m 将达到最大值。

虽然某个处理器可能只能实现特定的 N_{op}/N_m 比率，但该比率也受到所执行算法的限制，通常称为算法的 算术强度：

$$a \equiv N_{op}/N_m. \tag{4}$$

如图1所示的存内计算器件 [5] 可以通过一次从存储器中读取大量算子数据和输入向量数据，并在将输出写回存储器之前对所有数据进行整体运算，充分利用算法的算术强度。如果存内计算器件足够大且复杂，那么涉及这些数据的所有必要运算都可以完成，而无需在之后再次从存储器中读取任何输入。

回到式（1），我们根据算术强度，为冯·诺依曼机执行该算子所必须消耗的存储器访问能耗设定了一个下界。这进而对计算效率形成了限制：

$$\eta = \frac{1}{e_m/a + e_{op}} \tag{5}$$

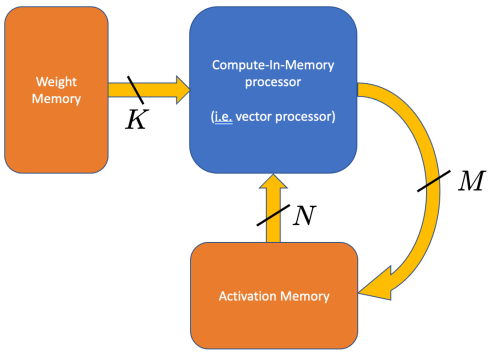

图1：数字存内计算处理器示意图。

因此，在实现算术强度任意高的算子时，内存访问能耗对计算效率的影响可以任意降低。式（5）中，随着算术强度增加，内存访问能耗对计算效率的影响随之减小，这反映了脉动阵列和张量处理单元节省的能量 [1, 2]。

我们指出，式（5）中的分析与处理器的 Roofline 模型 [6]类似；不过，本文重点关注能耗，而后者侧重于识别处理器速度的瓶颈。

为了了解存内计算设备能够带来多大优势，我们考察了几个线性算子的示例，并给出其算术强度。对于一个矩阵乘法，若一个矩阵的大小为 $L \times N$，另一个矩阵的维度为 $N \times M$，则内存访问总数为 $N_m = LN + NM + LM$，运算次数为$N_{op} = 2NML$，其中加法和乘法分别视为独立运算。在这种情况下，算术强度为：

$$a = \frac{2NML}{LN + NM + LM}, \tag{6}$$

当 N、M、$L \to \infty$ 共同变化时，其趋近于 ∞。

对于卷积，算术强度同样可以变得任意大，因为卷积可以实现为矩阵乘法。通常的做法是通过称为 im2col() 操作的方法，将输入数据重新排列成托普利兹矩阵。图 2 展示了在脉动阵列中使用矩阵乘法实现卷积的一般算法，其中 $n \times n$ 是一个输入通道的尺寸，C_i 是输入通道的数量，$k \times k$ 是一个卷积核通道的尺寸，C_{i+1} 是输出通道的数量（因此也是三维卷积核的数量）。通过复制和重新排列激活数据形成的托普利兹矩阵，其结果是一个 $(n-k+1)^2 \times k^2 C_i$ 矩阵。卷积通过将该矩阵与一个 $k^2 C_i \times C_{i+1}$ 矩阵相乘来完成其中包含权重。因此，使用矩阵乘法实现卷积时，通常会得到如下矩阵尺寸：

$$L = (n-k+1)^2 \approx n^2 \tag{7a}$$
$$N = k^2 C_i \tag{7b}$$
$$M = C_{i+1}. \tag{7c}$$

由此得到的算术强度为：

$$a = \frac{2n^2 k^2 C_i C_{i+1}}{n^2 k^2 C_i + k^2 C_i C_{i+1} + n^2 C_{i+1}}. \tag{8}$$

然而，由于激活数据被复制了约 k^2 次以构成输入矩阵，与原生实现卷积而非一般矩阵乘法的处理器相比，算术强度显著降低。为说明这一点，再考虑一个输入图像为 $n \times n$、具有 C_i 个输入通道、C_{i+1} 个输出通道且卷积核大小为 $k \times k$ 的卷积层。输入向量大小为 $N_i = n^2 C_i$，卷积核权重数量为 $K = k^2 C_i C_{i+1}$。如果只需读取必要的权重和激活数据，则该 i^{th} 层的算术强度将变为

$$a \approx \frac{2n^2 k^2 C_i C_{i+1}}{n^2 (C_i + C_{i+1}) + k^2 C_i C_{i+1}}. \tag{9}$$

在 $n^2 >> k^2 C_i$ 的极限情况下，与使用矩阵乘法实现卷积相比，其算术强度大约高出 k^2 倍。

无论是直接实现卷积，还是使用矩阵乘法实现卷积，公式 (9) 都表明，当 n，k，$C_i \to \infty$ 时，算术强度会变得任意大，使得公式 (5) 中内存访问能耗的贡献任意小。事实上，在大多数现代卷积神经网络中，这些参数都很大，因此具有较高的算术强度，如表 I 所示。根据存储体的大小（其决定了内存访问能耗），并参考表 IV 中给出的静态随机存取存储器访问能耗和数字乘加运算的数值，采用高算术强度算法的存内计算处理器，其内存访问能耗可以做到相对于计算能耗可忽略不计。

IV. 利用模拟计算降低计算能耗

遗憾的是，根据阿姆达尔定律，即使将存储器访问能耗降至任意低，逻辑单元消耗的计算能耗仍会限制整体性能提升幅度。若要将整体效率提升数个数量级，就必须同时解决这两方面的问题。

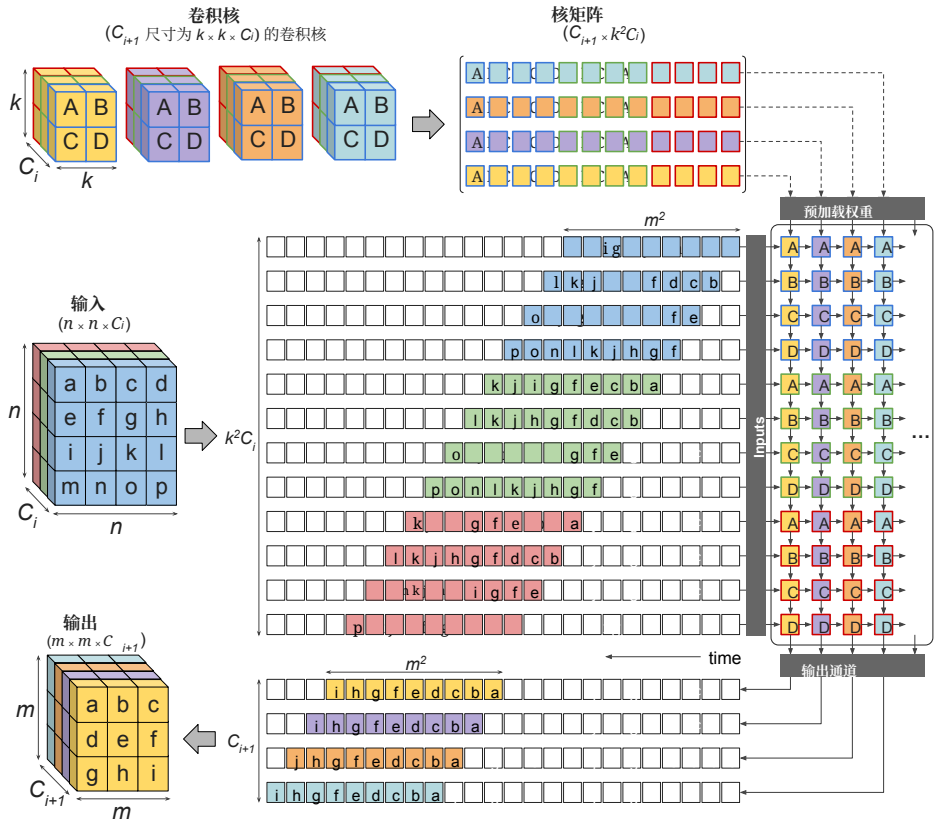

图 2：使用矩阵乘法在权重驻留脉动阵列中实现卷积的算法。输入数据被转换为托普利兹矩阵并馈入脉动阵列，其中每一行相对于其上方的行延迟一个时间步。

网络	层数中位数 n 中位数 C_i			最大值 N	平均值 k	总计 K	中位数 C_{i+1} 中位数 a	
DenseNet201	200	62	128	1.6e+07	2.0	1.8e+07	128	292
GoogLeNet	59	61	480	3.9e+06	2.1	6.1e+06	128	200
InceptionResNetV2	244	60	320	8.0e+06	1.9	8.0e+07	192	291
InceptionV3	94	60	192	8.0e+06	2.4	3.7e+07	192	295
ResNet152	155	63	256	1.6e+07	1.7	5.8e+07	256	390
VGG16	13	249	256	6.4e+07	3.0	1.5e+07	256	2262
VGG19	16	186	256	6.4e+07	3.0	2.0e+07	384	2527
YOLOv3	75	62	256	3.2e+07	2.0	6.2e+07	256	504

表 I：各类知名神经网络在输入图像为 1 百万像素（每通道）时的卷积层参数摘要。

近年来，人们提出了各种模拟计算技术，从电学到光学，旨在降低计算能耗。电子模拟计算通常以阻变存储器（或 ReRAM）交叉阵列为核心 [7–9]。光学模拟处理器通常基于硅光子学 [10–13]。自 20 世纪 80 年代以来，人们一直在探索光学4F系统，将其作为一种更高维度的计算形式 [14, 15], ；此外，利用光学表面的简单散射进行计算也正受到探索[16–18]。

支持模拟计算的论据本质上在于其可扩展性：模拟计算具有独特的优

在应用于低位精度的大型线性算子时，优势尤为明显 [19]。为了解这一点，请看一个通用模拟处理器（如图 3(a) 所示）：它接收 N 个精度为 B位的输入数据，生成 M 个精度为 B 位的输出数据，并由B位精度的 K 个权重进行配置，这些权重表示该矩阵。首先，通过数模转换器（DAC）转换 K 个权重，并将这些值施加到模拟处理器中的调制器上，从而完成模拟处理器的配置。随后从存储器中读取 N 个输入，并使用 DAC 将 N 个模拟输入施加到处理器上。根据处理器的物理特性，这自然会产生 M

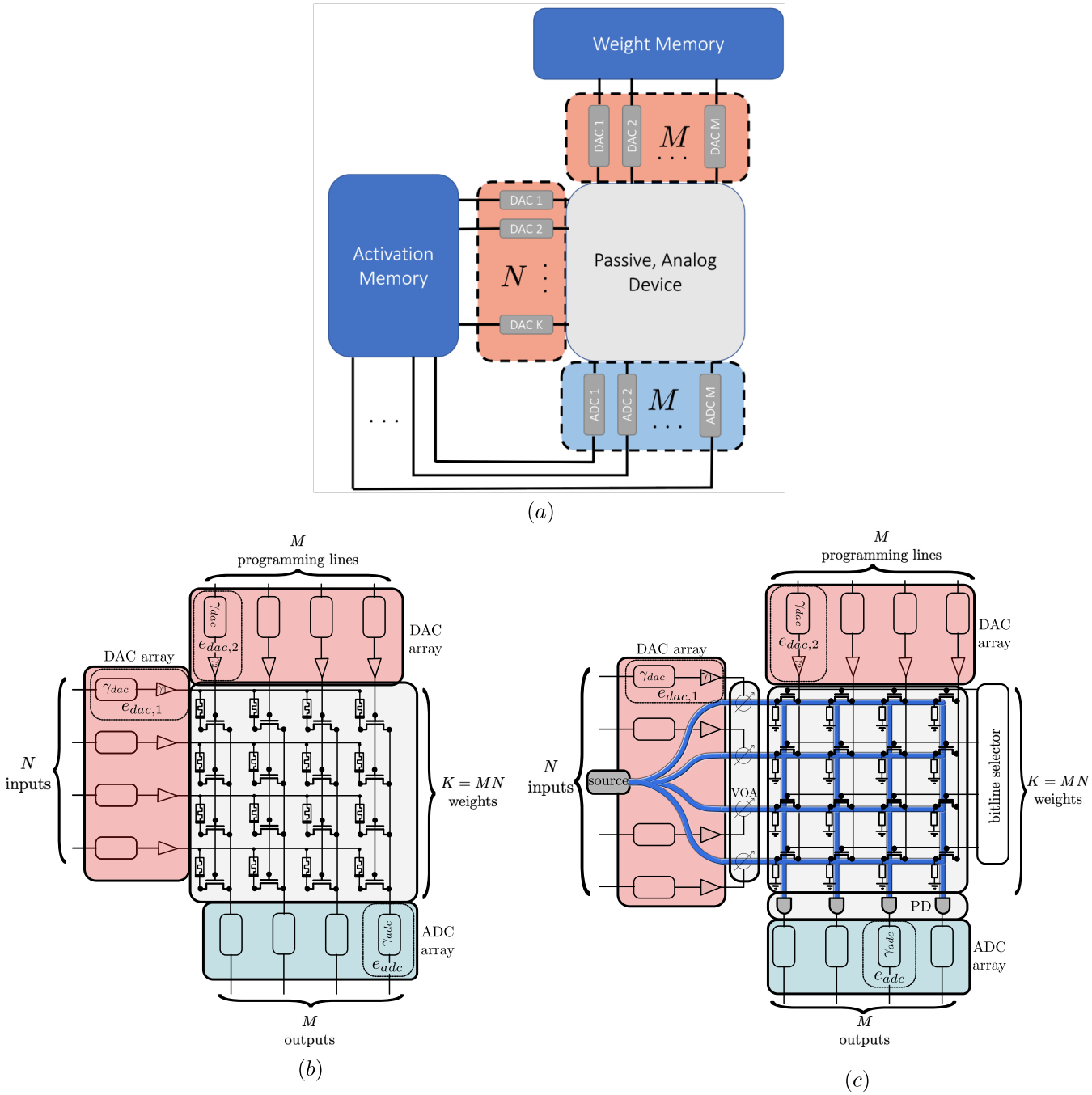

图 3：（a）模拟存内计算处理器的系统级视图。模拟器件通过数模转换器进行配置，以存储激活值或权重，而另一者则作为输入提供。（b）ReRAM 忆阻器交叉阵列模拟电子存内计算处理器的详细视图。每个晶体管都连接到一个可重构电阻器，其电导决定矩阵中每个元件的有效权重。（c）硅光子存内计算处理器的详细视图。每个晶体管都连接到一个电光元件，该元件会改变每个交叉点处的散射参数。

模拟输出，再通过模数转换器（ADC）转换回数字域。如果模拟处理器已经完成配置，或者无需重新配置，那么消耗的总能量就仅包括输入所用数模转换器和 ADC 的能量

对于输出：

$$E_{op} \equiv N_{op} e_{op} = N(e_{dac,1} + e_{adc}), \tag{10}$$

其中，为简化起见，我们假设 $N = M$ 。而

式（10）右侧表示模拟处理器消耗的计算能耗，左侧则表示数字计算机要达到与模拟计算机相同的效率时所需执行的等效数字运算次数（N_{op}）乘以每次运算所需的能量（e_{op}）。由于矩阵乘法的 $N_{op} = 2N^2$，如果以数字方式执行这一运算，所消耗的计算能耗将与运算次数成正比：$E_{op} = 2e_{op}N^2$。由此得出结论：模拟计算将矩阵乘法的能耗从 $\mathcal{O}(N^2)$ 降至 $\mathcal{O}(N)$。此外，这也意味着模拟计算的有效单次运算能耗与问题规模成反比，即

$$e_{op} \propto 1/N. \tag{11}$$

我们注意到，在实际应用中，缩放 N 由处理器的尺寸或问题的尺寸定义，取两者中较小者。

A. 向量-矩阵乘法

对于涉及神经网络的大多数问题，所构建的模拟处理器往往不足以存储整个神经网络。在这种情况下，重新配置模拟处理器中的权重可能会破坏 $\mathcal{O}(N)$ 缩放优势。为说明这一点，考虑将一个长度为 N 的向量与一个尺寸为 $N \times M$ 的矩阵相乘。此时，我们有，

$$N_{op}e_{op} = 2Ne_{dac,1} + 2MNe_{dac,2} + 2Me_{adc}. \tag{12}$$

我们还将数模转换器的能量 $e_{dac,\ 1}$ 和 $e_{dac,\ 2}$ 分开，因为有时配置模拟计算机和向其中馈入模拟输入时，会采用不同的物理机制和负载。这里，$e_{dac,\ 1}$ 表示每个输入所需的能量，而 $e_{dac,\ 2}$ 表示每次重新配置所需的能量。

通常，在模拟计算技术中，模拟存内计算器件只能存储正定数值（如忆阻器的情况）或完全复数（如耦合式 Mach-Zender 干涉仪的情况）。如果只能生成正数，那么为了同时考虑矩阵值的正负，必须将整个计算执行两次，然后求两次结果之差。另一方面，当允许使用复数值时，例如在硅光子 MZI 中，每个耦合式 MZI 调制器都需要两个电压（因而需要两次数模转换操作）来进行配置。此外，对于相干光学测量，必须采用干涉技术，才能从光电探测器中恢复场的正、负分量，因为光电探测器只能测量场的模平方。因此，无论采用哪种模拟计算方案，实际上都必须将式（12）中的每一项乘以 2，以处理正值和负值。

将式（12）应用于向量-矩阵乘法，可得：

$$e_{op} = e_{dac,1}/M + e_{dac,2} + e_{adc}/N, \tag{13}$$

在这种情况下，中间项既不与 $1/N$ 成正比，也不与 $1/M$ 成正比。

B. 矩阵乘法

在矩阵乘法的情况下，上述情形得到了缓解。在这种情况下，模拟计算机本身的配置会针对输入矩阵的每一行重复使用，从而使每次运算的能耗恢复为与问题规模成反比。在一个 $L \times N$ 矩阵乘以一个 $N \times M$ 矩阵的情况下，我们有

$$e_{op} = e_{dac,1}/M + e_{dac,2}/L + e_{adc}/N \tag{14}$$

因为 $N_{op} = 2NML$ 在这种情况下成立。由于能耗的三个独立组成部分中的每一个，都会按照与相乘矩阵相关的三个不同尺寸成比例的因子降低，因此，随着问题规模增大，每次运算的有效能耗会降低。对于尺寸有限的模拟处理器，后两个组成部分最终将受到模拟处理器本身的两个尺寸（输入和输出的数量）的限制。

此时，需要区分神经网络架构中所涉及矩阵的尺寸与模拟处理器的物理尺寸。我们用撇号标记矩阵尺寸，即 M'、N' 和 L'；用帽号标记处理器的物理尺寸：$\hat{M}$、$\hat{N}$。能耗实际节省的倍数（即方程（14）中的 M 和 N）取这两组数中较小者：

$$M = \min\{\hat{M}, M'\} \tag{15a}$$
$$N = \min\{\hat{N}, N'\}. \tag{15b}$$

C. 卷积

与数字处理器一样，模拟处理器也可以通过矩阵乘法实现卷积。卷积核和激活数据到矩阵尺寸的映射保持不变，即

$$L' = (n-k+1)^2 \approx n^2 \tag{16a}$$
$$N' = k^2 C_i \tag{16b}$$
$$M' = C_{i+1} \tag{16c}$$

采用权重驻留方案时。这些数值在采用激活驻留方案时会重新排列。与数字处理器一样，将卷积表示为纯矩阵乘法的一个遗憾之处是

网络	层数	L'	N'	M'
DenseNet201	200	3844	1152	128
GoogLeNet	59	3721	528	128
InceptionResNetV2	244	3600	432	192
InceptionV3	94	3600	768	192
ResNet152	155	3969	1024	256
VGG16	13	62001	2304	256
VGG19	16	38688	2304	384
YOLOv3	75	3844	1024	256

表 II：根据公式 (16)，各种知名神经网络卷积层中 L'、N' 和 M' 的中位数。这些数值是在考虑每通道 1 百万像素的输入图像后得到的。

输入激活值会被重复 k^2 次，这意味着与原生实现卷积、而非通用矩阵乘法的处理器相比，需要进行 k^2 次额外的数模转换操作（也可能产生更多内存访问）。因此，M 远小于式（16c）中的其他数值，所以将卷积实现为矩阵乘法的模拟处理器，在式（14）中对其输入数模转换器的摊销程度最低。表 II 列出了各种神经网络中 L'、N' 和 M' 的中位数。

V. 算子专用模拟处理器

截至目前，我们已经看到：1）通过在专用处理器上实现具有较大算术强度的网络，存储器访问能量对计算效率的贡献可以任意降低；2）模拟处理器在执行矩阵乘法时，还可以进一步降低计算能耗。计算能耗的降低幅度与模拟处理器能够处理的矩阵规模成正比。

基于平面矩阵乘法的处理器在执行卷积时存在一个固有缺点：为输入构造的矩阵尺寸为 $(n-k+1)^2 \times k^2 C_i$，比实际的激活数据大k^2 倍。以数字方式执行卷积时，这影响不大，因为该矩阵乘法所需的乘加运算数量与卷积所需的相同：$(n-k+1)^2 k^2 C_i$。然而，使用模拟处理器执行矩阵乘法时，若所用矩阵的行数比必要行数多 k^2 ，就意味着所需的数模转换操作比理论上必要的数量多 k^2 。更糟的是，除非使用额外逻辑在静态随机存取存储器与处理器之间设置矩阵（这也会消耗能量），否则还会比原则上必要的次数多进行 k^2 次内存读取，从而显著增加存储器访问能耗。此外，由于每层的通道数量往往彼此相关（一层的输出通道会成为下一层的输入通道），加载到模拟处理器中的权重数据尺寸为 $k^2 C_i \times C_{i+1}$ ，呈现高度矩形的形状，这会使 M 相对于 N增大，进而提高输入数据数模转换器对每次运算能耗的贡献。

与为通用矩阵乘法而设计的模拟处理器不同，有一类模拟处理器专门用于实现卷积。实现模拟处理器的一种方法，是将其限制为仅在算子的某个特定特征空间中运行。虽然任何线性算子都可以表示为矩阵，但矩阵 A 可以通过特征分解因式分解为三个矩阵的乘积：

$$X = U \Lambda U^T, \tag{17}$$

其中， U 是由 X的特征向量构成的酉矩阵（即无损矩阵），而 Λ 是由 X的特征值构成的纯对角矩阵。卷积的特征向量是波，因此，当 X 是表示卷积的矩阵时，特征向量矩阵 U 表示傅里叶变换，而 U^T 表示傅里叶逆变换。

构建算子专用处理器 的一种方法是静态实现矩阵 U 和 U^T，仅动态重新配置特征值 Λ。在这种情况下，要将线性算子从一个变为另一个，只需更改 Λ 的对角元素。换言之，如果矩阵 X 的大小为 $m \times m$，那么要将该矩阵变为另一个卷积矩阵，只需调制模拟处理器中的 m 个权重，而非 m^2 个权重。对于 X 表示卷积的特殊情况，这些特征值就是卷积核数据的傅里叶变换。通过调整这组 m 个元素，模拟处理器实现的矩阵 X 便可覆盖特征向量由 U给定的线性算子范围。

平面模拟处理器可以进行特征分解，事实上，硅光子处理器已实现了这一点 [11, 13]。不过，实现卷积专用处理器还有一种不同于硅光子学的方案，称为光学 *4F系统*；与平面卷积处理器相比，它具有一系列独特优势。

在平面模拟处理器中，数据以一维阵列的形式输入处理器，并在沿第二个维度传播时接受处理。与平面处理器不同，光学4F系统是一种体积处理器，因此数据以二维阵列的形式表示，而计算则在光沿第三个维度传播时进行。虽然这确实大幅提高了信息密度和计算密度，但最显著的区别在于，它使处理器能够扩展到平面处理器完全无法实际处理的输入数量。公式（11）表明，模拟计算的效率与模拟处理器的尺寸成正比（在算术强度趋于无穷大的极限下），因此，光学4F系统理论上可达到高出数个数量级的计算效率

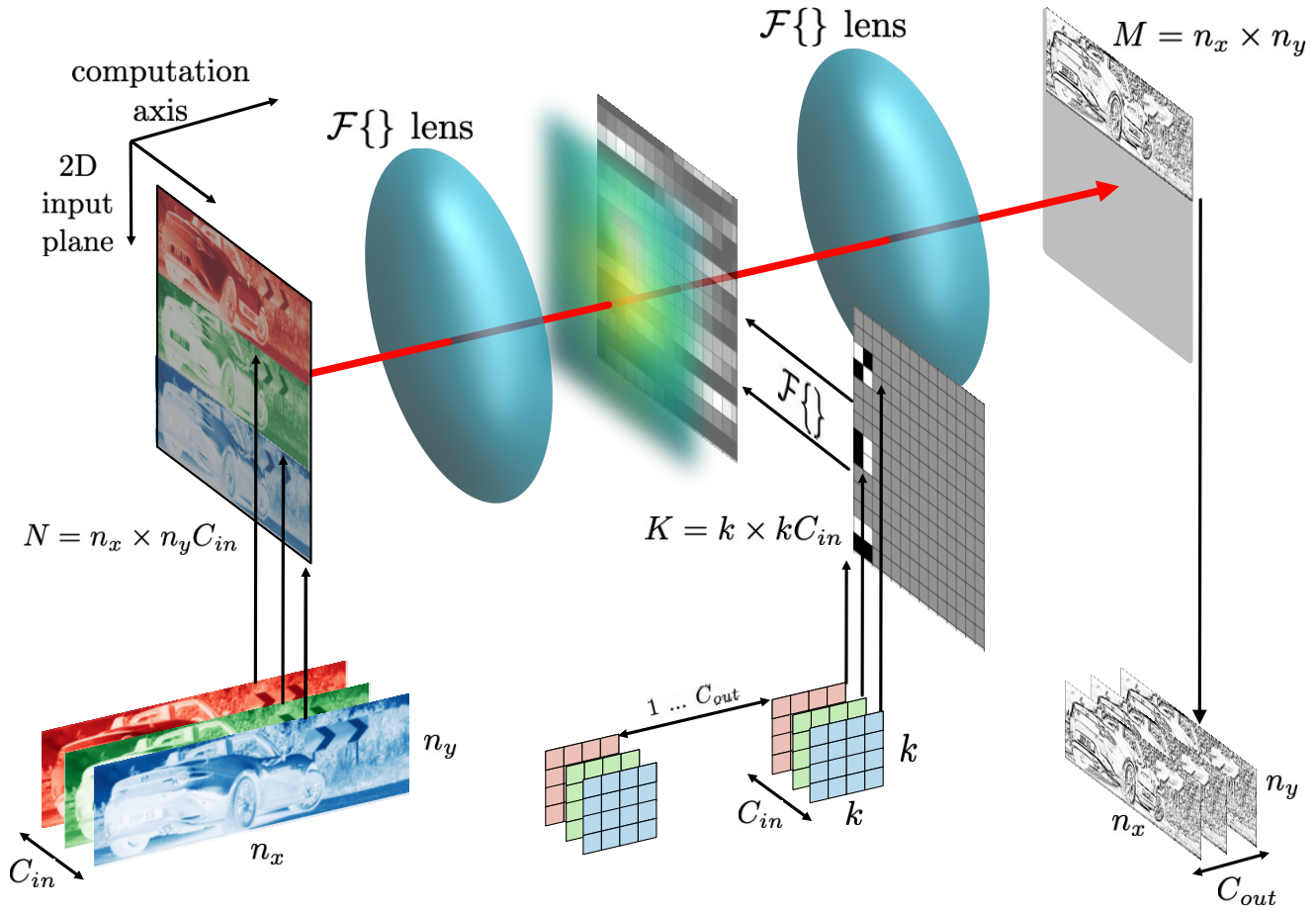

图4：透射式光学4F系统通过并行化输入通道执行卷积的示意图。输入激活数据可平铺在物平面上，而输入滤波器可在进行傅里叶变换并将数据加载到傅里叶平面上的第二个空间光调制器之前，采用适当的填充进行平铺。在这种配置下，每次测量都会产生一个完整的输出通道。

于平面处理器。

图4展示了一个光学4F系统处理器的示例。它由两个空间光调制器（空间光调制器）组成，可采用液晶单元或动态超表面实现。它们分别放置在透镜两侧，距透镜一个焦距。透镜会自然地在这两个平面之间执行傅里叶变换，因此，光线穿过第一个空间光调制器后，经过透镜时会发生傅里叶变换。因此，第一个空间光调制器提供输入数据，而第一个透镜表示与酉傅里叶矩阵 U 相乘。第二个空间光调制器加载卷积核数据的傅里叶变换；光线透过它后，得到输入数据的傅里叶变换与卷积核数据的傅里叶变换的乘积。因此，第二个空间光调制器表示与对角特征值矩阵 Λ 相乘。

随后，在第二个空间光调制器后方一个焦距处放置第二个透镜，这代表乘以第二个特征值矩阵 U。最后，在距第二个透镜两个焦距处放置一个探测器，因此，照射到探测器上的光就是输入数据与卷积核数据的卷积。探测器本身只对入射场的强度（即模平方）敏感。不过，仍可通过干涉测量方法恢复场的复数值。或者，正如其他人指出的那样，半导体的光吸收所执行的非线性测量，也可以自然地用作神经元的非线性激活。

如图 4 所示，如果在进行傅里叶变换并将数据应用于第二个空间光调制器之前，对卷积核数据进行适当填充，则可以并行处理多个输入通道。使用较小卷积核时，这样可以提高空间光调制器利用率。

遗憾的是，从计算系统的角度来看，传统光学4F系统存在一个致命缺陷：卷积的输出数据是在距离输入数据四个焦距的位置测量的，而输入数据据推测

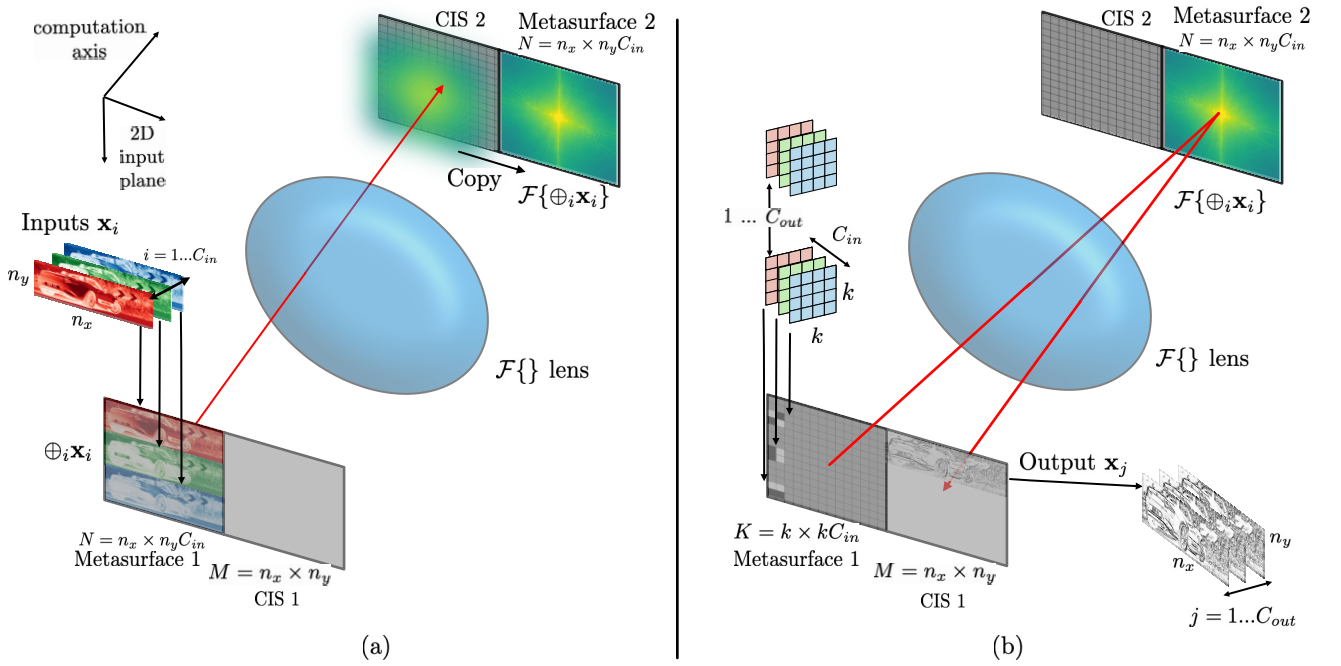

图5：反射式光学4F系统（折叠后总长度为2F）示意图。该系统分两个阶段处理包含所有输入和输出通道的完整卷积层：(a) 第一阶段，对输入激活数据进行光学傅里叶变换，并将结果加载到傅里叶面空间光调制器中；(b) 第二阶段，将输入通道平铺到物平面空间光调制器上，并并行测量所有输入通道的卷积。对每个输出通道重复此过程。

必须在独立的芯片上以物理方式实现。由于这次卷积运算仅表示两个神经元层之间的连接，因此，要实现具有两个以上神经元层的深度神经网络，就必须以某种方式将探测器芯片的输出数据送回输入空间光调制器。将如此海量的数据传送到芯片外会带来巨大的能耗，从而抵消大规模模拟计算带来的所有优势。

然而，如图5所示，可以利用反射式空间光调制器折叠光学4F系统，将第一个空间光调制器和CMOS图像传感器并排整合到单个芯片中，并且仅使用一个透镜。在这种体系结构中，两个芯片之间所有重要的数据传输均以光学方式而非电子方式进行。透镜两侧各有一个芯片，每个芯片分为两半：一半是空间光调制器（或超表面），另一半是CMOS图像传感器。两个芯片分别放置在透镜两侧、距透镜一个焦距的位置，因此每当光线在两个芯片之间传播时，透镜都会执行一次傅里叶变换。

该系统分两个阶段计算卷积：加载阶段和计算阶段。第一个阶段，即加载阶段，如图 5(a) 所示，其目的是对激活数据进行傅里叶变换，并将变换结果加载到第二个超表面上。一组输入滤波器映射被写入第一个芯片中的输入空间光调制器（SLM），并对其进行照明。反射光的傅里叶变换被传送到第二个芯片中的 CMOS 图像传感器（CIS），然后通过数模转换器和模数转换器运算，将这些数据以电子方式传输到同一芯片中的第二个空间光调制器（SLM）。与图 4 中的透射式展开4F系统一样，图 5 所示的反射式4F系统也可用于并行计算多个输入通道的卷积。因此，该阶段的最终结果是：第二个芯片中的空间光调制器（SLM）已配置为激活数据的傅里叶变换。

在第二个计算阶段，输入卷积核权重数据被施加到第一个空间光调制器上。随后以略微倾斜的角度对其进行照明，使反射光照射到第二块芯片中的空间光调制器上。当这束光反射时，透镜会再次进行傅里叶变换，而照射到第一块芯片中 CIS 上的光，就是输入滤波器映射数据与卷积核数据的卷积。

如果输入数据需要 $n^2 C_i$ 个像素，那么加载激活数据的光学傅里叶变换将耗费

$$E_{fft} = n^2 C_i (2e_{adc} + 4e_{dac}) \tag{18}$$

能量。每个像素需要一次数模转换操作，才能将输入数据写入第一个超表面；而要从强度数据重建复场数据并将其施加到第二个空间光调制器上，则需要两次数模转换操作和两次模数转换操作。

由于输入通道可以并行执行，然后遍历输出通道，因此第二阶段包-

含两次 $K = k^2 C_i C_{i+1}$ 数模转换操作，以及 CIS 中的两次 $n^2 C_{i+1}$ 模数转换操作，以恢复该场。

$$E_{conv} = 2k^2 C_i C_{i+1} e_{dac} + 2n^2 C_{i+1} e_{adc} \tag{19}$$

因此，该层模拟计算所涉及的总能量为 $E_{fft} + E_{conv}$，

$$E_{op} = 2n^2(C_i + C_{i+1})e_{adc} + 2C_i(2n^2 + k^2 C_{i+1})e_{dac}. \tag{20}$$

$N_{op} =$ 执行的运算总数为 $2n^2 k^2 C_i C_{i+1}$ 。

因此，该方法的效率为：

$$\eta = \frac{1}{e_m/a + e_{adc}/\left(\frac{k^2 C_i C_{i+1}}{C_i + C_{i+1}}\right) + 2e_{dac}/k^2 C_{i+1} + e_{dac}/n^2}. \tag{21}$$

在超表面足够大、能够处理所有激活数据或权重数据的情况下。

为了考虑超表面的有限尺寸（它们可能不足以同时容纳所有通道的全部激活数据），我们首先确定实际上一轮可处理的输入通道数量。对于尺寸为 $n_x \times n_y \equiv N$ 的超表面，一轮可纳入的输入通道数量 C' 为：

$$C' = \lfloor \hat{N}/n^2 \rfloor. \tag{22}$$

用该数量替代软件定义的实际输入通道数量后，我们可以推导出在 $C' \geq 1$ 的情况下，光学4F系统节省能量的倍数：

$$L = n^2 \tag{23a}$$

$$N = \frac{k^2 C' C_{i+1}}{(C' + C_{i+1})} \tag{23b}$$

$$M = k^2 C_{i+1}/2. \tag{23c}$$

用这些参数表示，光学4F系统的效率可按通常方式表示为：

$$e_{op} = e_{dac}/M + e_{dac}/L + e_{adc}/N. \tag{24}$$

对于光学4F系统，各种神经网络的 L、N 和 M 的中位数如式 (23) 所示，结果列于表 III。

VI. 解析结果

式 (3)、(5)、(14) 和 (24) 中给出的公式可用于估算在上述四种计算平台中的任意一种上评估给定卷积神经网络层时的效率。这些公式取决于内存访问、数模转换器/模数转换操作以及数字乘法的能量值。其中许多量的估算值列于表 IV，

网络	层数	L	N	M
DenseNet201	200	3844	272	136
GoogLeNet	59	3721	128	64
InceptionResNetV2	244	3600	224	112
InceptionV3	94	3600	240	120
ResNet152	155	3969	1024	512
VGG16	13	62001	2304	1152
VGG19	16	38688	3456	1728
YOLOv3	75	3844	512	256

表 III：各种知名神经网络的卷积层中 L、N 和 M 的中位数

将光学4F系统视为计算基底的网络。数值是在考虑每通道 1 百万像素的输入图像和无限大的超表面（即 $C' \to$inf）时得出的。

e_m (96kB 静态随机存取存储器)[3]	4.3pJ
e_{mac} [3]	0.23pJ
e_{adc} [20]	0.25pJ
e_{dac} [21]	0.01pJ
e_{opt} [式（A8）]	0.01pJ
e_{load} 对于 $4\mu m$ 间距，$N = 256$ [式（A6）]	0.08pJ
e_{load} 对于 $250\mu m$ 间距，$N = 40$ [式（A6）]	0.8pJ
e_{load} 针对 $2.5\mu m$ 间距，$N = 2048$ [式 (A6)]	0.04pJ

表 IV：数字和模拟计算机各种运算的能量。这些数据假设工艺节点为 45nm，电压为 0.9V，且每次运算处理 8 位数值。内存访问能耗示例假设存储体大小为 96kB，因为构建 TPU 静态随机存取存储器存储体时采用的正是这一存储体大小。

附录中还给出了用于推导负载的公式，以估算各种模拟计算平台的数模转换器能耗。

这些数值均取决于互补金属氧化物半导体工艺节点，但可以使用缩放定律在不同工艺节点之间进行插值 [22]。我们通过考察一个参数如表 V 所示的卷积神经网络层来比较各种计算平台，并在图 6 中绘出了其能效随工艺节点变化的曲线。

虽然所有处理器都会随着工艺节点的进步而提升，但数字存内计算处理器与硅光子处理器之间大致存在一个数量级的差距，而硅光子处理器与光学4F系统之间预计还将存在另一个数量级的差距。

输入通道	C_i	128
输出通道 C_{i+1} 128		
滤波器尺寸	k	3
输入规模	n	512
算术强度	a	230

表 V：用于估算图 6 中各种处理器效率的卷积参数。算术强度可根据公式 (9) 由其他参数得出。

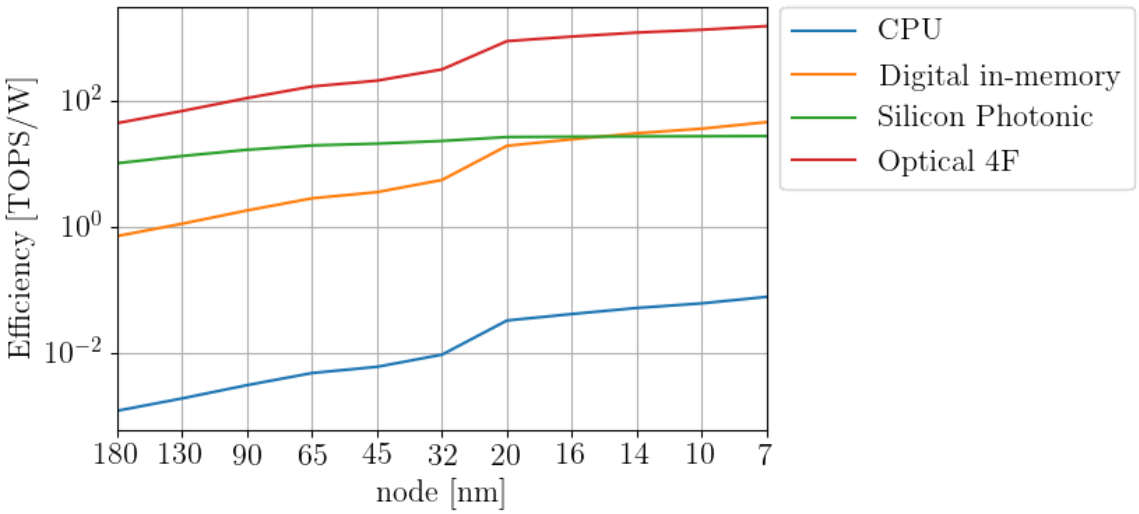

图 6：不同计算架构的解析模型效率随工艺节点的变化。

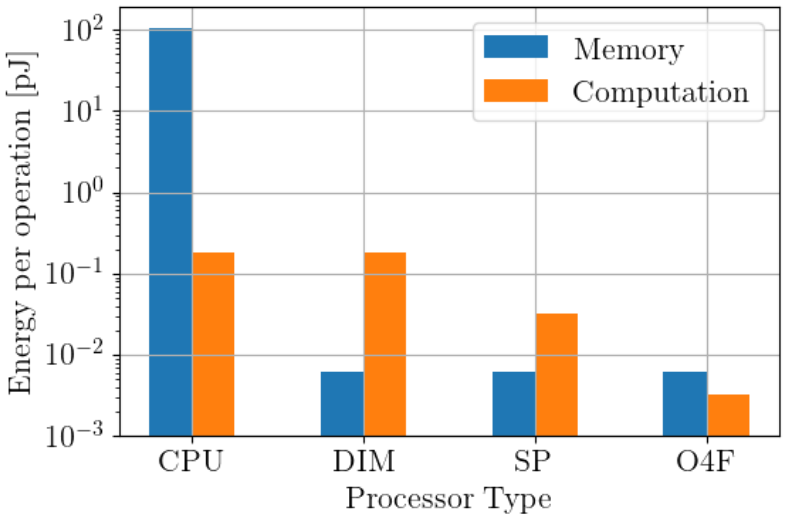

图7：不同处理器类型的单次运算能耗构成。DIM 指数字存内计算，SP 指硅光子，O4F 指光学4F系统架构。CNN层参数见表 V，架构细节的假设见正文。所有处理器类型的工艺节点均假设为 32纳米。

尽管这种差异显然与算法相关，但模拟计算系统的底层硬件必须足够庞大，才能充分利用潜在的算法优势；而从二维硅光子处理器转向一种本质上类似于光学4F系统的三维处理器，正是实现这一点的途径。

图7展示了改进效果在存储能耗降低和计算能耗降低方面的分解，并分别列出各类处理器中存储和计算元件对每次运算能耗的贡献。利用存内计算实现高算术强度，首先大幅降低中央处理器与其他平台之间的能耗差异，使存储能耗远低于计算能耗。模拟处理器则进一步降低了计算能耗；对于每次运算而言，输入更多的模拟处理器计算能耗更低。

值得注意的是，图 6 中报告的数字存内处理器能效显著高于 Google TPU。对于采用 28纳米工艺节点制造的芯片，Google TPU 的能效因 CNN架构而异，为 0.3-2 每瓦万亿次运算。此处建模的存内计算数字处理器与 TPU 具有相同的架构参数：一个 256×256 的脉动阵列，以及由 256 个 96-KB 存储体组成的 24-MiB 静态随机存取存储器。我们预测，其能效应约为 5 每瓦万亿次运算，

这比文献 [1] 报告的能效高得多。不过，我们指出，这一估算简化了数字乘法以及在脉动阵列各处理元件中存储和传输数据所涉及的能耗。

图 6 和图 7 中建模的硅光子处理器采用了 40×40 的阵列大小，这对于文献中报告的大多数处理器而言是典型配置 [10–13]，因为各种调制器技术通常要求阵列间距处于 100–400 µm 范围内。计算能耗在很大程度上受光调制器技术限制，目前约为 7 pJ/字节，具体讨论见 A 1 节。我们在模型中假设，随着时间推移，这一数值将改善至 0.5 pJ；但即使考虑这一假设优势，从图 6 可以清楚看出，除非能够扩大处理器尺寸，否则硅光子学很难在内存技术中的数字计算效率方面保持优势。我们还假设硅光子处理器采用 24 MiB SRAM，并按照 TPU 架构将其划分为 40 个 600 KB 的 SRAM 存储体。

光学4F系统基于图5中的体系结构，采用4百万像素空间光调制器和一个划分为2048个、每个容量为12 KB的存储体的24 MiB SRAM，同样遵循TPU体系结构。用于空间光调制器有源矩阵寻址的数模转换器负载的空间光调制器像素间距假设为2.5 µm，由此得到0.9 fF的线电容和40 fJ的负载能耗，如表 IV所示。每个像素的光学能量基于1550纳米光，每次运算贡献10 fJ/像素，如表 V所示。现实空间光调制器尺寸所支持的大阵列大小能够将计算能耗降低到甚至低于图7中的存储能耗。

VII. 计算结果

到目前为止，我们已经给出了简单的解析公式，用于根据各种 AI 推理平台的扩展方式估算其效率。这些公式只是近似值，存在若干局限，其中最主要的一点是，它们没有考虑矩阵规模过大、超出存内计算器件或其输入容量的情况。在这种情况下，需要将问题拆分为多个较小的矩阵乘法。为克服这一局限，我们开发了脉动阵列和反射式光学 4F 系统的周期精确模型，并在给定输入图像尺寸下评估各种卷积神经网络时对这些模型进行了测试。随后，我们将更精确的计算结果与前几节中的解析模型进行了比较。

A. 脉动阵列效率估算

为了分析脉动阵列的能效，我们考虑了一种与 Google TPU [ref:TPU], 类似的体系结构，其中采用了由 256 x 256 个单元组成的权重驻留脉动阵列。阵列的 256 个端口各自可访问一个 96 KB 的 SRAM存储块，共计提供 24 MiB 的缓冲存储器，用于存储激活值（即卷积层的输入/输出）。权重存储在动态随机存取存储器中，并根据所执行的卷积层进行访问。激活值和权重均采用 8位定点表示。

在能耗方面，我们以 [3]中0.9 V、45纳米工艺下的SRAM和乘加运算能耗值作为参考：SRAM读写能耗为 1.25 pJ/字节（8 KB存储器），8位乘加运算能耗为 0.23 pJ/字节。为与TPU中的96 KB SRAM存储块大小保持一致，依据公式（A2），将8 KB SRAM的能耗按大小缩放$\sqrt{96\text{K}/8\text{K}} = 3.46$，得到4.33 pJ/字节。对于每次乘加运算，我们还计入了每个阵列瓦片中的负载能耗和存储器读写能耗（用于存储/传播8位输入和32位累加 $= 40$ 位）。利用公式A6计算得到负载能耗为2.82 fJ/位，其中阵列瓦片之间的距离根据256×256阵列在整个TPU芯片（331 mm^2）中的面积占用率（24%）进行近似，得到瓦片间距离为34.8 µm。内部阵列存储器能耗通过将8 KB SRAM存储块缩放至40位获得，结果为1.25 pJ/字节$\times\sqrt{5/8\text{K}} = 31.25$ fJ/字节。

最后，我们采用 [22], 中介绍的技术，将所有能量值（负载除外，因为它不直接依赖于工艺）从 45纳米工艺缩放到相应的工艺节点，范围从 180 nm 到 7 nm。结果如图 8所示。解析表达式和周期精确模型呈现相同的趋势，随着工艺节点缩小，两者略有偏离。这可以归因于 e_{load} 不依赖于工艺节点；随着节点尺寸缩小，其他能量来源逐渐减少，其能耗开始成为总体能耗中的主导因素。

B. 光学计算机效率估算

对于光学4F系统，我们考虑了 4 百万像素空间光调制器，并采用与系统阵列分析中相同的 24 MiB SRAM。这样，SRAM 被划分为 2048 个相等部分（每个部分对应一行超表面），得到经尺寸缩放后的 SRAM读写能耗为 1.55 pJ/byte。数模转换器、模数转换器和激光器的能耗取自表 IV中的数值，采用 2.5微米间距。

图 9 展示了光学4F系统的解析表达式与周期精确模型之间的比较。该图给出了效率的整体曲线，显示采用更小工艺节点构建设备可带来显著提升

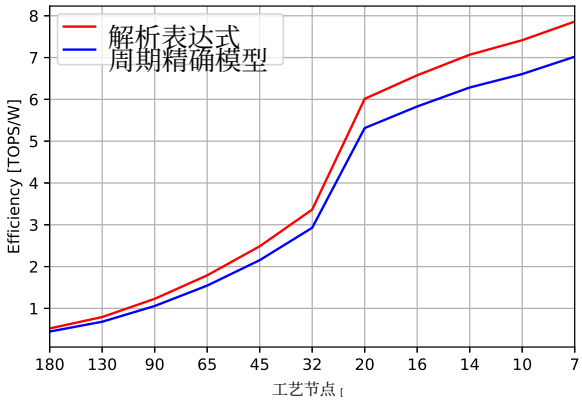

图 8：周期精确模型与方程（8）给出的解析表达式及表 I 中数值的效率比较。两个模型均使用 256×256 权重驻留脉动阵列和 24 MiB SRAM（与 Google TPUv1 中相同）运行 YOLOv3（1 百万像素输入图像）。

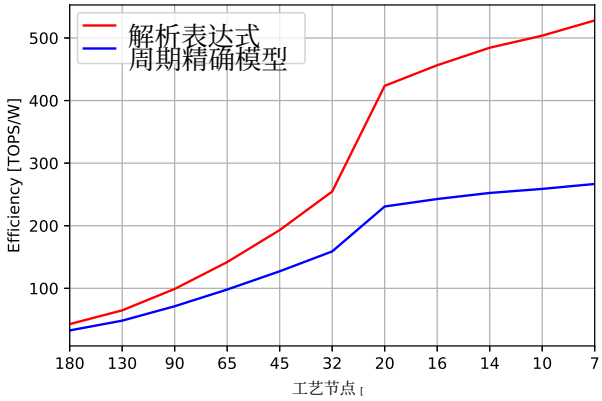

图 9：将方程（24）与运行 YOLOv3（1 百万像素输入图像）的光学 4F 处理器的周期精确模型进行比较，该处理器使用 4 百万像素空间光调制器和 24 MiB SRAM。

。导致解析模型与周期精确模型之间出现差异的主要因素包括：

- 周期精确模型会考虑超表面的确切执行次数，以统计输出探测器 ADC 读取操作、输出存储器访问次数以及消耗的激光器总能量。
- 方程（23）假设输出尺寸与输入尺寸相同（即 $m{=}n$），因此自然无法考虑大于 1 的步幅。
- 方程（24）中 e_{dac} 的值由 $e_{dac,1}+$ 构成 $e_{load}+e_{opt}$，因此能耗取决于超表面中的有效像素数量。然而，周期精确模型通过将逐像素能耗（$e_{dac,1}+e_{load}$）与取决于超表面尺寸的激光器能耗（e_{opt}）分开，更准确地估算能耗。

C. 光学计算机能耗分布

光学4F系统的周期精确模型可以基于四种不同的系统组件：数模转换器、模数转换器、SRAM和激光器，对能耗分布进行详细汇总。这些针对不同工艺节点下VGG19和YOLOv3的结果如图10所示，数值以每次乘加运算对应的皮焦耳表示。

随着工艺节点尺寸减小，模数转换器和SRAM的能耗自然会降低。另一方面，数模转换器能耗中包含占主导地位的 e_{load} ，由于后者与工艺节点无关，因此在不同工艺节点下，数模转换器的总体能耗几乎没有降低。与 e_{load}一样，激光器能耗 e_{opt} 也不会随工艺节点变化，因此保持不变。

比较VGG19（左）和YOLOv3（右）的能耗分布，可以有趣地发现，算术强度高得多的网络（如VGG19，参见表 I）每次乘加运算对应的SRAM能耗反而更高。这可以解释为，周期精确模型考虑了空间光调制器和输入的尺寸；由于VGG19（平均每层）具有更大的输入图像和更多通道，其输入图像像素在超表面上的放置效率略低。这样会导致更多的超表面执行次数，并因此需要更多的输出激活缓冲（SRAM读写），才能完成网络中的卷积。如果考虑一个无限大的超表面，那么这一人为因素自然会消失；就每次乘加运算对应的SRAM能耗而言，VGG19也会比YOLOv3更高效。

VIII. 结论

存内计算和模拟计算技术都是有效的技术，可用于应对处理器总能耗中的不同组成部分。对于算术强度较高的算法，存内计算能够降低每次运算的存储器访问能耗；而模拟计算则能够按模拟处理器的规模降低计算本身的能耗。卷积神经网络非常适合采用此类模拟存内计算体系结构，因为它们具有较高的算术强度和大型线性算子，并且前向传播通常只需较低的位精度。

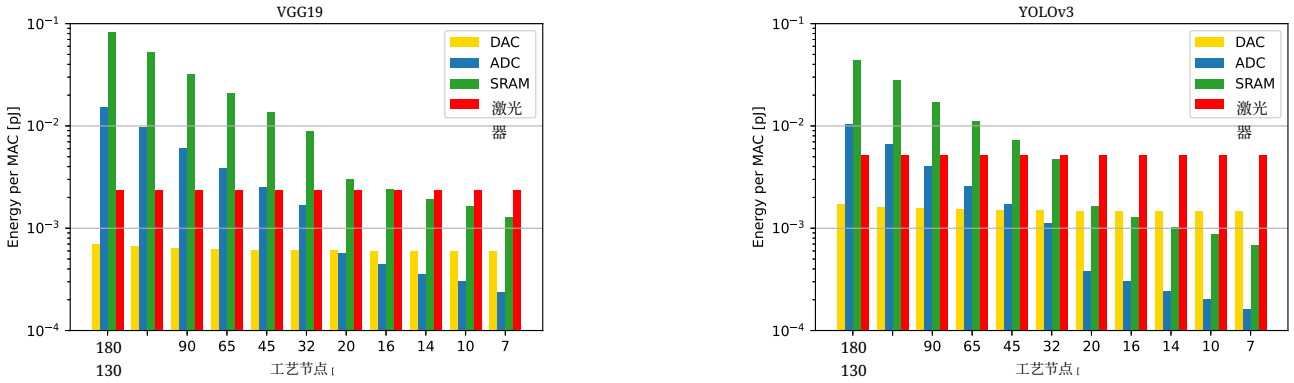

图10：4F光学系统运行 VGG19（左）和 YOLOv3（右）时，周期精确模型的能耗分布。

为了直观说明采用其中一种或两种技术能够在多大程度上提升能效，我们给出了简单的解析公式，用于估算多种处理器类型的能效，包括数字存内处理器（如脉动阵列）、模拟存内计算处理器，以及专用于卷积的一类模拟存内处理器——光学4F系统。将这些解析公式应用于表I至表II中给出的平均神经网络参数后，所得结果与图8和图9中TPU架构的周期精确模型吻合良好。

如图 6 所示，在任何现代工艺节点上，这些方法的性能都比中央处理器高出几个数量级。性能提升最大的一项来自存内计算处理器每次运算的存储器访问能耗降低，如图7所示。然而，这种技术如此有效，以至于计算能耗成为主要能耗来源，而模拟计算又进一步降低了这部分能耗。由于模拟计算的能耗优势与模拟处理器的规模成正比，光学4F系统具有独特优势：它们可以扩展到足够大的规模，使每次运算的计算能耗低于存内计算处理器所需的最低存储器能耗，从而能够运行现代卷积神经网络算法。

附录A：处理器能耗模型参数

本节将推导内存访问、数字乘加、模数转换器和数模转换器的典型单次运算能耗；为了恰当地比较本文讨论的各种计算方案，这些推导都是必不可少的。

数字乘加运算消耗的能量会随着逻辑单元中所涉及的逻辑门数量而增加：事实上，数字乘加运算的下限由朗道尔极限决定，该极限与逻辑门数量成正比。对于串并行乘法器，逻辑门的数量为

门 G 为 $G = 6B^2$；对于其他乘法器实现方式，面积或逻辑门数量仍与 B^2[23], 成正比，其中 B 是操作数的位数。全加器每位还需要额外的九个逻辑门，因此我们可以写成

$$e_{mac} = \gamma_{mac}(6B^2 + 9B)kT \tag{A1}$$

其中， k 是玻尔兹曼常数， T 是温度，而γ_{mac} 是无量纲常数。朗道尔极限指出，每次乘加运算的能量下限为 $\gamma_{mac} > \ln(2)$。[24] 通常，45纳米工艺下 $\gamma_{mac} \approx 122,500$ [3], 因此，当前数字乘法器理论上仍有几个数量级的提升空间。

所有加速器都需要内部存储器来存储神经网络参数和中间变量（除非模拟处理器具有足够大的容量来存储整个网络，但目前这并不切实际）。数字电子SRAM存储体每次运算的能量随用于寻址和写入数据的位线和字线长度而变化，因为大部分功耗都消耗在为这些线路形成的等效电容器充放电上。因此，通常可以将每次内存访问的能量表示为：[3]

$$e_m = e_{m0}\sqrt{N_m} \tag{A2}$$

其中， N_m 是存储体的大小， e_{m0} 是一个能量单位的常数。这里所示的缩放关系并不反映朗道尔极限所规定的能耗下限，因为目前导致静态随机存取存储器能耗的，是电容性线路的充放电，而非逻辑门的切换；这也是其能耗随阵列大小的平方根变化的原因。在仅有一个位单元的极限情况下，可以令$e_{m0} = \gamma_m kT$，将 e_{m0} 与朗道尔极限进行比较。由此得到的 γ_m 与朗道尔极限相差的数量级，比数字乘加运算还要大得多：对于45纳米CMOS工艺， $\gamma_m \approx 3 \times 10^6$

对应于 $e_{m0} \approx 5$ 飞焦耳。可以说，与朗道尔极限相比，e_{m0} 的数值极大；此外，静态随机存取存储器中寻址线路的电容所产生的功耗，使能耗与阵列大小的平方根成正比。这两点大体上正是计算面临的最严峻能耗问题的根源。[3]幸运的是，对于实现卷积等高算术强度运算的专用处理器，我们能够显著缓解这一问题。[5]

对于模拟计算，模数转换器的能量随位精度呈指数级变化，根本原因在于必须具备足够的信噪比才能区分各个电平。当这些电平以线性电压步长定义时，模数转换器每次采样的能量为[20, 25]

$$e_{adc} = \gamma_{adc} kT 2^{2B}, \tag{A3}$$

其中，k 为玻尔兹曼常数，T 为温度，γ 为无量纲常数。已有研究[20]指出，受热噪声影响，γ_{adc} 的下限为 $\gamma_{adc} > 3$ ，并通过一项实证调查表明，65纳米工艺下片上模数转换器的先进水平为 γ_{adc} ≈1404，缩放至45纳米工艺后约为927。

数模转换器的缩放方式与模数转换器相同：

$$e_{dac} = \gamma_{dac} kT 2^{2B} \tag{A4}$$

理由也类似。然而，γ_{dac} 的先进水平为 $\gamma_{dac} \approx 39$。[21]

然而，式（A4）中的表达式仅考虑了数模转换器电路本身消耗的功率，没有考虑驱动模拟处理器负载所消耗的功率。例如，图3(b)中与ReRAM处理器相关的位线负载，与光学模拟处理器中图3(b)所示可变光衰减器（VOA）的负载会有很大差异。光学处理器还会产生额外的能量消耗，即光学激光功率本身；这部分可以视为负载能量的一部分。因此，我们可以写为：

$$e_{dac,i} = \gamma_{dac} kT 2^{2B} + e_{load,i} \tag{A5}$$

对于 $e_{dac、1}$ 和 $e_{dac、2}$，情况均是如此。在以下小节中，将分别估算模拟忆阻器处理器和硅光子处理器中的这些量。

不过，我们注意到，对于物理尺寸较大的阵列，负载通常可能主要由行列寻址线的电容决定。位线和字线电容导致的能量耗散公式为：

$$e_{load,i} = (1/2)\mathcal{C} L V^2 \tag{A6}$$

其中，$\mathcal{C}$ 是该互连线单位长度的电容，L 是互连线长度。作为参考，典型的CMOS铜互连线电容约为0.2 fF $/\mu m$[26]，因此对于电压为0.9伏的工艺，每次运算通常消耗0.08 fJ$/\mu m$ 。

1. 硅光子学模拟处理器

对于光学计算机，驱动输入的负载既包含光学部分，也包含电学部分：

$$e_{load,1} = e_{elec} + e_{opt}. \tag{A7}$$

电气组件将涉及驱动某种电光调制器，而每次运算的能量通常取决于该组件的电容。在硅光子学领域，数据输入端可能采用可变光衰减器（VOA），而阵列中通常使用马赫-曾德尔干涉仪、MEMS 调制器或相变调制器来存储权重数据。一些用于电光调制器的低能耗方案是等离激元谐振器。其中，迄今记录的等离激元调制器每次调制的最低能量约为$e_{elec} \approx 9$ pJ。[27, 28] 这些数值与由掺杂硅微环谐振器制成、通过载流子等离子体色散进行调谐的电光调制器相当；后者已实现约 0.9pJ/bit，即 7pJ/B。[29] 未来或许可以设计出每个样本能耗低于这些数值的光调制器。[30]

由于散粒噪声是光学噪声的主要来源，负载本身的光学贡献将随位精度呈指数变化。因此，对于光学元件，我们可以写作：

$$e_{opt} = \frac{\hbar\omega}{\eta_{opt}} 2^{2B} \equiv \gamma_{opt} kT 2^{2B} \tag{A8}$$

其中，$\hbar$ 是以角频率为单位的普朗克常数，ω 是光的频率，η_{opt} 是光学系统和光电探测器的效率。值得一提的是，由于噪声通常主要来自散粒噪声，光功耗也按 2^{2B} 缩放。以此为例，对于1550纳米光和80%的光学效率，我们有 $\gamma_{opt} \approx 39$，对应于 $e_{opt} \approx 10$ 飞焦耳。考虑到当前电光技术下每个样本对应的能量，光学部分的能量可以忽略不计。

与权重重配置相关的负载，$e_{dac,\ 2}$，只有电学部分，其中涉及电光调制器以及用于阵列寻址的电比特线。超低能耗电光调制器通常尺寸也很小，以尽量降低电容；其尺寸约为几微米，因此沿阵列长度方向的每个元件会额外消耗几飞焦耳的能量。与电光调制器本身相关的能量相比，这部分能量也可忽略不计。因此，$e_{dac,\ 1}$ 和 $e_{dac,\ 2}$ 均由电光调制器能耗主导。

2. 忆阻模拟处理器

在ReRAM处理器中，负载有两部分：阵列中导电线路相关的电容，以及忆阻器中的电荷耗散。ReRAM阵列的间距通常受每个工艺节点上所放置晶体管的尺寸限制，这意味着阵列的位线和字线相对较短，电容也较低。尽管如此，大型阵列中的能耗仍可能由电容主导，其计算公式见式（A6）。

另一方面，在 ReRAM 阵列中，忆阻器本身每次运算消耗的能量也可能相当高，因为能量既与阵列大小成正比，也与其平均电导成正比，而电导受到 $G_0 = 2e^2/h$ 的量子电导限制，其中 e 是电子的电荷，h 是普朗克常数。因此，这些元件的电导受到限制，其范围为 $G = G_0$ 至 $G = G_0 2^B$，适用于 B 位精度的元件。

忆阻器是高度非线性的元件，因此通常通过脉宽调制来提供输入数据，而不是改变电压。因此，整个阵列消耗的能量可以表示为所有忆阻器之和

$$\langle E_{ReRAM} \rangle = \delta t \sum_{i=1}^{M} \sum_{j=1}^{N} \langle G_{ij} \rangle \langle V_j^2 \rangle. \tag{A9}$$

其中 δt 是采样周期。使用每个忆阻器的电导和电压标称值，我们可以将方程简化为

$$\langle E_{ReRAM} \rangle = \delta t MN \langle G \rangle V_{rms}^2. \tag{A10}$$

阵列的一次操作中，乘加运算的数量为 MN，因此，在 ReRAM 阵列中，阵列每次运算消耗的平均能量实际上是一个常数，不会因扩大阵列规模而降低：

$$e_{ReRAM} \equiv \frac{\langle E_{ReRAM} \rangle}{MN} = \langle G \rangle V_{rms}^2 \delta t \tag{A11}$$

如上所述，忆阻器的电导只有在高于量子电导时才表现良好。假设服从均匀分布，其平均值将是动态范围的一半，因此 $\langle G \rangle = 2^{B-1} G_0$。

能量与电压的平方成正比，因此我们假定电压受到限制，以确保相对于忆阻器中约翰逊-奈奎斯特热噪声 V_{noise} 限值，精度达到 B 个有效位。对于 dt 的时钟周期，热噪声为：

$$V_{noise}^2 = \frac{4kT}{G_0 \delta t}. \tag{A12}$$

由于最大噪声由最小电导决定，将 $V_{rms}^2 = (3/2)\, 2^{2B} V_{noise}^2$ 设为维持 B 位精度所需的最小电压，则

有源ReRAM [8, 31]	1-4μm
光学相变材料 [10]	250 μm
光学马赫-曾德尔干涉仪（MZI）[13]	100 μm

表 VI：各种模拟计算调制技术的典型间距。

γ_m	3×10^6
γ_{mac}	1.2×10^5
γ_{adc}	583
γ_{dac}	39
γ_{opt}	105

表 VII：各种运算的无量纲常数值。假设工艺节点为 45nm，电压为 0.9V。假设 γ_{opt} 的光学效率为 50%。

忆阻器阵列每次运算所吸收的最小能量为：

$$e_{ReRAM} = 3kT 2^{3B}. \tag{A13}$$

虽然这是理想的解决方案，但在实际中，可施加的电压存在一个最小值，通常远高于热噪声极限，约为 $V_{rms} \approx 70$ mV。采用这一估算值和 $\delta t = 1$ ns 的采样周期，忆阻器每次运算的能量为 $e_{ReRAM} \approx 0.05$ pJ，约为商用忆阻器阵列每次运算能量的五分之一；尽管如此，这仍将效率上限限制在 $\eta \approx 20$ TOPS/W。

[1] N. P. Jouppi、C. Young、N. Patil、D. Patterson、G. Agrawal、R. Bajwa、S. Bates、S. Bhatia、N. Boden、A. Borchers、*et al.*，《数据中心张量处理单元的性能分析》，载于 *Proceedings of the 44th annual international symposium on computer architecture* (2017)，第 1–12 页。[2] A. Samajdar、J. M. Joseph、Y. Zhu、P. Whatmough、M. Mattina 和 T. Krishna，《一种系统化方法，用于表征 DNN 加速器的可扩展性》ing scale-sim，在 *2020 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*（IEEE，2020）第 58–68 页。[3] M. Horowitz，《1.1 计算的能耗问题（以及我们能对此做些什么）》，载于 *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)* 2014 10 14（IEEE，）第 – 页。[4] A. Reuther、P. Michaleas、M. Jones、V. Gadepally、S. Samsi 和 J. Kepner，《机器学习加速器的调查与基准测试》，

机器学习加速器，载于 *2019 IEEE high performance extreme computing conference (HPEC)*（IEEE, 2019）第 1–9 页。[5] A. Sebastian、M. Le Gallo、R. Khaddam-Aljameh 和 E. Eleftheriou，《用于存内计算的存储器件及其应用》，《自然·纳米技术》**15**，529（2020）。[6] S. Williams、A. Waterman 和 D. Patterson，《Roofline：一种富有洞察力的多核体系结构可视化性能模型》，《ACM通信》**52**，65（2009）。[7] M. Demler，《Mythic 在闪存中实现乘法》，《微处理器报告》（2018）。[8] Q. Xia 和 J. J. Yang，《面向类脑计算的忆阻器交叉阵列》，《自然·材料》**18**，309（2019）。[9] A. Shafiee、A. Nag、N. Muralimanohar、R. Balasubramonian、J. P. Strachan、M. Hu、R. S. Williams 和 V. Srikumar，《Isaac：具有交叉阵列原位模拟运算的卷积神经网络加速器》，《ACM SIGARCH计算机体系结构新闻》**44**，14（2016）。[10] J. Feldmann、N. Youngblood、M. Karpov、H. Gehring、X. Li、M. Stappers、M. Le Gallo、X. Fu、A. Lukashchuk、A. S. Raja、*et al.*，《使用集成光子张量核心进行并行卷积处理》，《自然》**589**，52（2021）。[11] J. R. Ong、C. C. Ooi、T. Y. Ang、S. T. Lim 和 C. E. Png，《使用集成衍射光学元件的光子卷积神经网络》，《IEEE量子电子学专题期刊》**26**，1（2020）。[12] M. Y.-S. Fang、S. Manipatruni、C. Wierzynski、A. Khosrowshahi 和 M. R. DeWeese，《具有组件误差的光学神经网络设计》，《光学快报》**27**，14009（2019）。[13] Y. Shen、N. C. Harris、S. Skirlo、M. Prabhu、T. Baehr-Jones、M. Hochberg、X. Sun、S. Zhao、H. Larochelle、D. Englund、*et al.*，《利用相干纳米光子电路进行深度学习》，《自然·光子学》**11**，441（2017）。[14] P. Ambs，《光计算：60 年的探索之旅》，《光学技术进展》（2010）。[15] J. Chang、V. Sitzmann、X. Dun、W. Heidrich 和 G. Wetzstein，《采用优化衍射光学元件进行图像分类的混合光电卷积神经网络》，《科学报告》**8**，1（2018）。[16] X. Lin、Y. Rivenson、N. T. Yardimci、M. Veli、Y. Luo、M. Jarrahi 和 A. Ozcan，《使用衍射深度神经网络进行全光机器学习》，《科学》**361**，1004（2018）。[17] C. Qian、X. Lin、X. Lin、J. Xu、Y. Sun、E. Li、B. Zhang 和 H. Chen，《通过衍射神经网络执行光学逻辑运算》，《光：科学与应用》**9**，1（2020）。[18] T. Zhou、X. Lin、J. Wu、Y. Chen、H. Xie、Y. Li、J. Fan、H. Wu、L. Fang 和 Q. Dai，《利用可重构衍射——式处理单元，《自然·光子学》**15**, 367 (2021). [19] R. Hamerly, L. Bernstein, A. Sludds, M. Solja˘ci´c, and D. Englund，基于光电乘法的大规模光学神经网络，Physical Review X **9**, 021032 2019( ). [20] B. E. Jonsson，寻找高能效模数转换器体系结构的经验方法，载于 *Proc. of 2011 IMEKO* ( ) pp. – *IWADC & IEEE ADC Forum* 2011 1 6 . [21] P. Palmers and M. S. Steyaert，一款采用电流转向架构的 10 位、1.6 GS/s、27 mW 数模转换器，在 130 nm 互补金属氧化物半导体工艺下实现 550 MHz、54 dB SFDR 带宽，IEEE Transac- tions on Circuits and Systems I: Regular Papers**57**, 2870 (2010). [22] A. Stillmaker and B. Baas，用于准确预测 180 nm 至 7 nm 互补金属氧化物半导体器件性能的缩放方程，Integration **58**, 74 (2017). [23] I. Koren, *Computer Arithmetic Algorithms* (PRENTICE HALL, 1993) 第 6 章。高速乘法。[24] R. Landauer，计算过程中的不可逆性与热量产生，IBM journal of research and devel- opment **5**, 183 (1961). [25] M. Saberi, R. Lotfi, K. Mafinezhad, and W. A. Serdijn，用于逐次逼近模数转换器的电容式数模转换器的功耗与线性度分析，IEEE Transactions on Circuits and Systems I: Regular Papers **58**, 1736 (2011). [26] N. H. Weste and D. Harris, *CMOS VLSI design: a circuits and systems perspective* (Pearson Education India, 2015). [27] G. Dabos, D. V. Bellas, R. Stabile, M. Moralis-Pegios, G. Giamougiannis, A. Tsakyridis, A. Totovic, E. Li- dorikis, and N. Pleros，神经形态光子技术与体系结构：扩展机遇与性能前沿，Optical Materials Express **12**, 2343 (2022). [28] C. Haffner, D. Chelladurai, Y. Fedoryshyn, A. Josten, B. Baeuerle, W. Heni, T. Watanabe, T. Cui, B. Cheng, S. Saha, *et al.*，低损耗等离激元辅助电光调制器，《自然》**556**, 483 (2018). [29] C. Sun, M. T. Wade, Y. Lee, J. S. Orcutt, L. Al- loatti, M. S. Georgas, A. S. Waterman, J. M. Shainline, R. R. Avizienis, S. Lin, *et al.*，可直接通过光进行通信的单芯片微处理器，《自然》**528**, 534 (2015). [30] D. A. Miller，面向低能耗信息处理与通信的阿焦级光电子学，Journal of Lightwave Technology **35**, 346 (2017). [31] A. Khakifirooz, K. Cheng, Q. Liu, T. Nagumo, N. Lou- bet, A. Reznicek, J. Kuss, J. Gi mbert, R. Sreenivasan, M. Vinet, *et al.*，面向片上系统应用的超薄 SOI，载于 *Proceedings of the IEEE 2012 Custom Integrated Circuits Conference* 2012 1 4 (IEEE,) pp. – .