中国科学: 信息科学 2019 年 第 49 卷 第 3 期: 334–341
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
面向智能应用的定制计算加速器技术专题 · 观点与争鸣

深度学习驱动的领域专用架构

马立伟

北京比特大陆科技有限公司, 北京 100192
E-mail: Liwei.Ma@bitmain.com

收稿日期: 2018–10–18; 接受日期: 2018–12–12; 网络出版日期: 2019–03–19

摘要 深度学习是人工智能近年来的新进展, 其对计算的新需求驱动新的计算架构. 本文首先通过分析人工智能的阶段和任务指出深度学习的需求实质, 然后从 3 个方面讨论深度学习领域专用架构, 分别是计算结构的评价标准、数字计算的数制基础和深度学习计算架构的研究方向. 本文首次提出使用 K-L 距离 (Kullback-Leibler divergence) 来评价深度学习结构的复杂度和准确度. 本文认为以 Posit 数制为基础, 不仅可以重新构造深度学习的计算架构, 而且可以重新构造科学计算的计算架构, 形成计算芯片设计的后发优势. 最后全文总结认为深度学习驱动的领域专用架构将是计算架构创新的重要组成部分.

关键词 深度学习, 熵, 互熵, 数制, 计算架构

1 引言

从摩尔定理的终结来推演领域专用架构的必要性, 已经获得了足够的重视, 这是供给侧的分析. 本文开篇试图从人工智能与深度学习的需求侧来论证新的领域专用架构的迫切性.

人工智能有 3 种研究方法, 分别是符号主义 (symbolism)、连接主义 (connectionism) 和行为主义 (behaviorism), 它们是人工智能的 3 个平行的支柱. 如图 1 所示, 从发展历史和内在逻辑来看它们之间也存在演进关系, 这也决定了对应的计算硬件架构的演进关系.

(1) 符号主义从逻辑推理机发展而来, 是人们发现可以用机械操作代替人类体力劳动后, 进而追求用机械推理代替人类脑力劳动而发展出来的产物. 它希望把大脑内的推理过程转换成计算机可以执行的程序和规则, 依赖专家编程. 歌德尔不完备定理 (Gödel's incompleteness theorems) 已经证明了永远存在一些命题是符号计算机无法解答的. 从硬件实现上看, 中央处理器 (CPU) 是这种计算范式最好的载体, 它强调在寄存器 (标量) 之间相互操作, 追求单核的极致性能.

(2) 连接主义不模拟大脑内逻辑思考的过程, 而是直接模拟大脑本身的行为. 大脑由大量的相互连接的神经元组成, 而单个神经元的行为是可以分析和模拟的. 大规模连接的神经元模型通过反馈训练能够对输入做出期望的反应. 任意连接的网络结构很难训练, 目前也只是找到若干种有很明显规则结

引用格式: 马立伟. 深度学习驱动的领域专用架构. 中国科学: 信息科学, 2019, 49: 334–341, doi: 10.1360/N112018-00278
Ma L W. Domain-specific architectures driven by deep learning (in Chinese). Sci Sin Inform, 2019, 49: 334–341, doi: 10.1360/N112018-00278

© 2019《中国科学》杂志社

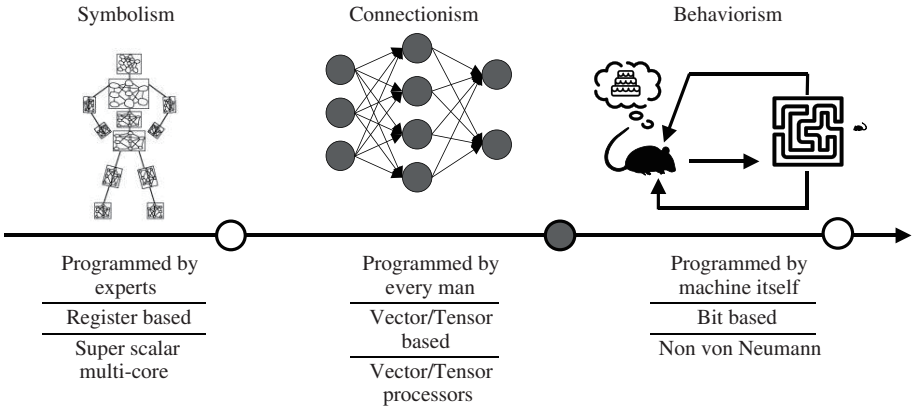

图 1 人工智能的 3 种研究方法

Figure 1 Three research methodologies for artificial intelligence

构的网络取得了一些进展, 如卷积神经网络 (convolutional neural network)、循环神经网络 (recurrent neural network)、长短期记忆网络 (long, short-term memory neural network) 等. 连接主义不关心内部逻辑, 只关注输入输出的拟合关系, 近年来在大数据和计算力提高的条件下获得长足的进步. 深度学习解决了莫拉维克悖论 (Moravec's paradox), 提供了从外部世界获取符号信息的有效途径. 采集大量普通人做某项任务的数据, 训练得到神经网络模型, 控制机器运行, 这也是一种编程, 因此可以认为连接主义依靠大家编程. 从硬件实现上看, 张量处理器 (tensor processing unit, TPU) 是这种计算范式的最佳载体, 它强调在原始采样数据 (张量) 之间相互操作, 追求数据并行的极致性能.

(3) 行为主义则把关注的对象从大脑直接转移到任务评价本身. 虽然它的很多具体实现还依赖前两种方法, 但是它提供了一个更加宏大的自主学习的框架. 它能够让机器自主地采集和标注数据, 在数据的规模、广度和深度方面远超过人的能力, 可以说是让机器自己编程. 它搜集到的数据维度之多, 多到只有维度本身变得有意义, 且每个维度的数据幅度不再重要, 因此它硬件结构的可能方向之一是单比特现场可编程异或门阵列 (field programmable xor-gate array) [1].

根据对研究客体和发展脉络的分析, 从符号主义到连接主义再到行为主义的发展是一个演进的进程. 目前深度学习方兴未艾, 人工智能处于连接主义后期向行为主义过渡的阶段. 这一阶段的主要任务是从环境的原始传感数据提取有价值的信息. 这是物联网市场的本质需求之一, 如在安防领域场需要从视频中提取结构化信息从而能够进行检索, 在智能家居领域需要从用户语音中提取文本信息从而能够人机互动. 这就需要从信息 (熵) 的角度来讨论、解释和评价深度学习以及它的算法和结构.

2 从熵的角度讨论深度学习专用计算架构

深度神经网络是目前发现的能够比较好地拟合输入输出关系的一种计算结构. 假设有一个映射关系 $y = f(x)$, 深度学习能够训练出一个神经网络模型 $y' = f'(x)$ 来拟合本来的映射关系. 那么需要考察如下问题:

(1) 如何衡量从 x 映射到 $y = f(x)$ 的困难程度?

(2) 如何衡量模型 $f'(x)$ 计算出来的 y' 的准确程度?

(3) 为什么深度神经网络结构可以取得更好的拟合效果?

(4) 如何衡量实现 $f'(x)$ 的算法或者结构的成本?

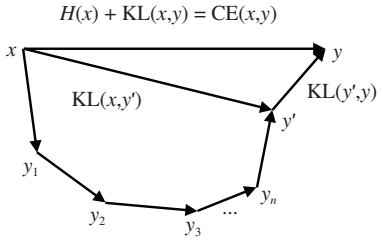

图 2　从输入 x 到输出 y 的可能的 K-L 链条

Figure 2　The possible K-L chains from input x to output y

(5) 什么是选择最佳实现方法的客观标准?

对于第 1 个问题, 一个映射的困难程度是可以被直观感受的. 如果 x 的输入是确定的一个, 自然的 y 也只有一个, 这个映射问题看起来是简单的 (例 A). 如果 x 的输入有概率均等的两种可能, 但是 y 永远输出一个呢 (例 B)? 例 A 与 B 哪一个更难?

对于随机变量 x, 它包含的熵是可以计算的,

$$H(x) = -\int_X p(x)\ln p(x)\,\mathrm{d}x, \quad x \in X.$$

同样可以计算 y 的熵 $H(y)$. 同时, 可以定义从 x 到 y 的互熵:

$$\mathrm{CE}(x,y) = -\int_X p(x)\ln q(y)\,\mathrm{d}x, \quad x \in X,$$

其中 $p(x)$ 和 $q(y)$ 分别是 x 和 y 的概率分布函数. 这里, $\mathrm{CE}(x,y)$ 与 $H(x)$ 的差被定义为 K-L 距离 (Kullback-Leibler divergence):

$$\mathrm{KL}(x,y) = \int_X p(x)\ln\frac{p(x)}{q(y)}\,\mathrm{d}x, \quad x \in X.$$

本文认为, 可以使用 K-L 距离来衡量一个映射的困难程度. 例 A 中 x 的熵为 0, 它到 y 的互熵为 0, K-L 距离为 0; 例 B 中 x 的熵为 1, 它到 y 的互熵为 0, K-L 距离为 -1. 根据这个衡量标准, 例 B 比例 A 更简单.

对于第 2 个问题, 从 y' 到 y 的 K-L 距离 $\mathrm{KL}(y',y)$ 也可以用来衡量 y' 的准确程度, 目前很多深度学习框架中的分类层就是使用了 K-L 距离作为代价函数.

虽然 K-L 距离被称为 "距离", 但是它并不满足三角不等式, 也就是说对于一个映射链条, $y_1 = f_1(x)$, $y_2 = f_2(y_1)$, $\mathrm{KL}(x,y_1) + \mathrm{KL}(y_1,y_2)$ 并不一定大于 $\mathrm{KL}(x,y_2)$, 这两者的关系可以是 "大于"、"等于" 和 "小于" 中的任何一种[1]. 如图 2 所示, 对于一个深度神经网络, 可以构造出一条映射链条, 使得

$$\mathrm{KL}(x,y_1) + \mathrm{KL}(y_1,y_2) + \cdots + \mathrm{KL}(y_{n-1},y_n) + \mathrm{KL}(y_n,y') + \mathrm{KL}(y',y) < \mathrm{KL}(x,y).$$

这在某种程度上解释了第 3 个问题, 之所以深度神经网络结构可以取得更好的拟合效果, 是因为映射链条越长, 总的 K-L 距离反而有机会变得更小.

对于第 4 个问题, K-L 距离同样可以用来判断具体实现一个映射的成本, 包括时间复杂度、空间复杂度和能量消耗. 每一个具体的实现都会涉及到硬件载体和软件调度, 并不能直观地和 K-L 距离联

1) https://en.wikipedia.org/wiki/Kullback-Leibler_divergence. 2018.

系起来, 这里做一个定性的分析. 一个电路, 一段程序, 或者一个函数, 它输入输出之间的 K-L 距离, 可以定义为它的转换熵. 考察双输入的与门、或门和异或门, 假定每个输入的 "0" 和 "1" 的概率是相同的, 可以计算出与门和或门的转换熵是 $-\frac{3}{4}\times\log_2 3$, 而异或门是 -1, 也就是说异或门比与门和或门都复杂, 这是符合直观判断的. 推而广之, 一个软硬件系统也可以计算转换熵. 转换熵小的软硬件系统, 实现的综合成本一般来说会更低.

现在可以回答第 5 个问题, 在完成相同功能的深度学习的不同实现之间, 需要权衡两个维度, 一是它的实现成本, 即各层的转换熵之和, 二是它的精度, 即电路输出与真实输出之间的 K-L 距离.

在保证功能 (精度) 的前提下, 如何设计转换熵最小的电路和算法? 在电路和算法实现中应该剔除冗余保留最有价值的信息, 这就需要重新思考计算的数制基础.

3 重新思考计算的数制基础

深度学习算法研究中一个很重要的分支是研究模型的量化与稀疏. 通过对深度模型中的权重参数进行量化, 缩减模型中参数的数量, 可以减少深度学习需要的计算量, 从而大幅降低算法和硬件实现的难度, 降低电路运行的功耗, 这方面的研究已经很多, 不赘述.

为什么深度学习中需要的数可以进行大幅度的量化与压缩? 这是由两方面的原因决定的. 第一, 真实的物理世界是有结构而非均一的, 信息主要是保存在不同的位置上, 而不是在相同位置的不同幅度上. 深度学习算法为了拟合真实世界, 需要更多的维度来表达信息, 需要更多的深度来表达信息之间的转换关系, 而不需要在单一维度上保留更高的精度. 第二, 在深度模型的参数和计算中间结果里, 数据的分布是不均匀的, 数据的重要性也是不均匀的. 在深度学习的计算中, 加法和乘法是最主要的两种计算形式, 而 "0" 和 "1" 分别是这两种运算的单位元, 这使得 0 附近的数的精度对加法结果的精度影响更大, 1 附近的数的精度对乘法结果的精度影响更大. 因此数据分布和重要性上的不均匀性为数据的压缩编码提供了可能性.

目前这类的量化方案有均匀定点化, 如使用 INT16, INT8 等, 有浮点数的一些变种, 如 Flexpoint 等, 但都是现有方案的一些改进. 一个能够灵活编程的硬件架构, 需要找到一种统一的可以适配多种精度的数制方案, 可以在浮点和定点之间恰当地分配比特. 这里着重介绍一下由 Gustafson 等 [2] 发明的 Posit 数制方案. Posit 是用来替代 IEEE Float 的一种浮点数制方案, 有很多优良的性质. 一般来说, 计算硬件能够表达的数是有限的, 因此任何一种编码方案都是在实数轴进行采样, 得到一组点的集合, 用来表示计算的结果. 如果计算的结果超出了这组点的集合, 就需要对结果进行舍入, 转换到集合中的一个. 定点数是在一定范围内进行均匀采样, 超出这个范围的数就计作溢出. 浮点数为了扩大数制的表达范围, 对不同区间的数进行不同精度的采样. 数制设计就是要在表达范围和精度之间权衡. 根据前面的分析, 一个合理的数制应该在加法和乘法的单位元附近分配更多的比特, Posit 数制就是根据这个原则设计的. Posit 的数制编码也是在实数轴上进行采样, 根据所在区间的数的重要性确定采样精度. Posit 在实数轴上的采样密度函数是

$$\sigma(x)(1-\sigma(x)),$$

其中 $\sigma(x)$ 就是该采样密度函数的积分, 也就是深度学习中非常重要的 Sigmoid 函数:

$$\frac{1}{1+\mathrm{e}^{-x}}.$$

图 3 展示了 Posit 的采样密度函数和累积分布函数.

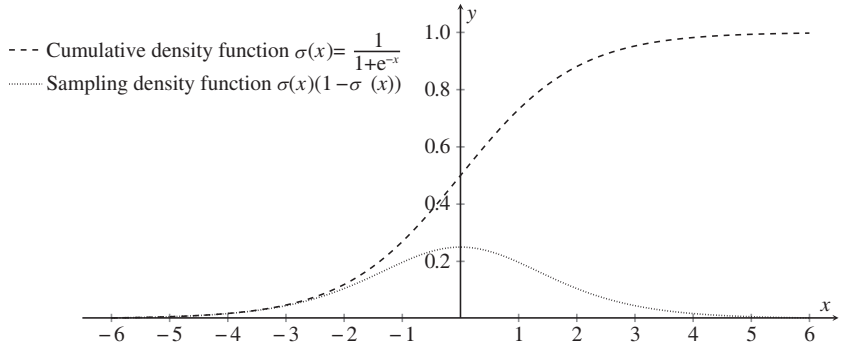

图 3　Posit 数制在实数轴的采样密度函数和累积分布函数

Figure 3　The sampling density and cumulative density function of Posit number system

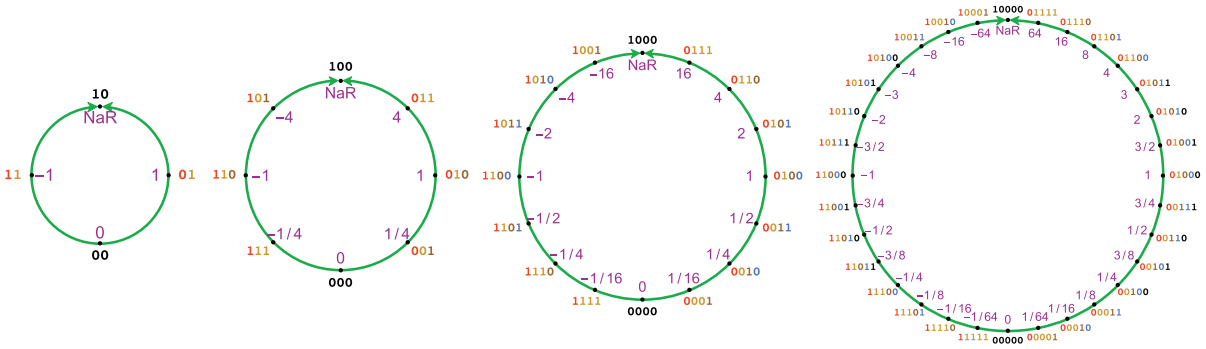

图 4　(网络版彩图) Posit 数制设计原则

Figure 4　(Color online) Principles of Posit design

如图 4 所示, 一个比特只能表示两个数. 其一选择超出实数的范围, 计为 “NaR (not a real)”, 其二选择一个最重要的实数, 计为 “0”. 两个比特只能表示 4 个数, 应该选择 “0”, “+1”, “−1” 和 “NaR”, 这时可以表达加法单位元, 乘法单位元和正负关系. 3 个比特能够表示 8 个数, 这时候可以增加 “±useed”, “$\pm\frac{1}{\text{useed}}$”. useed 是采样实数轴的粒度, 是可以任意选择的, 不影响最终数制设计的结构性质, 但是它应该与计算需要的粒度相匹配. 这里选择 useed=4 进行展开. 如果增加第 4 个比特, 那么新增加的 8 个数应该 “均匀” 地插入在既有的 8 个数中间, 即在 0 和 $\pm\frac{1}{4}$ 之间插入 $\pm\frac{1}{16}$, 在 $\pm\frac{1}{4}$ 和 ±1 之间插入 $\pm\frac{1}{2}$, 以此类推. 如果再增加第 5 个比特, 那么新增加的 16 个数还是应该 “均匀” 地插入在既有的 16 个数中间, 并以此类推. “均匀” 的含义与所在数字空档的位置是相关的, 其中有一套编码的规则, 有的指代表达范围的幂级扩展, 有的指代几何均值, 有的指代算术均值. 感兴趣的读者可以参考文献 [2] 了解更详细的编码规则.

从图 4 可以看出 Posit 数制的一些优点:

- 新增加的比特永远放在尾部, 表达在既有的数字之间插入的新数字, 这使得不同位宽编码的 Posit 数字拥有相同的头部比特序列. 这大大简化了不同编码宽度之间的转换和存储成本, 是现有的 IEEE Float 编码方式无法实现的.
- 新增加的尾部比特或者增加整个数制的编码范围, 或者增加某一区间的编码精度. Posit 能够自动灵活选择对应区间的采样密度, 最大限度地同时照顾编码的范围和精度.

以 Posit 数制为基础, 我们不仅可以重新构造深度学习的计算架构, 而且可以重新构造科学计算

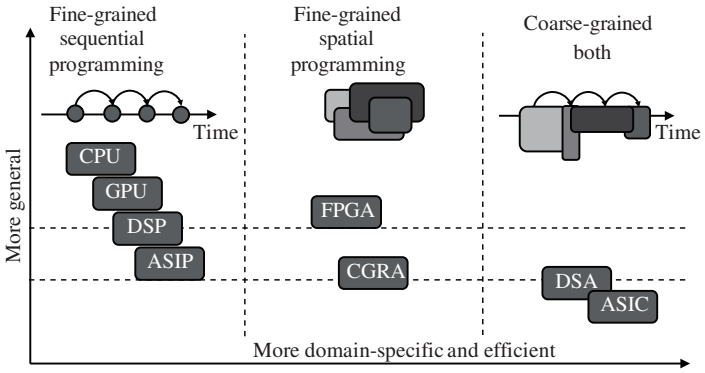

图 5 计算架构的比较

Figure 5 The comparison of computing architectures

的计算架构. 这可以打破以 IEEE Float 构造起来的计算产业格局, 形成计算芯片设计的后发优势. 目前, 一些研究已经证实了 Posit 数制的编码优势. Lindstrom 等 [3] 比较了一系列浮点编码的替代方案, Posit 数制在编码效率和计算精度方面都超过 IEEE Float, 在很多任务中是表现最好的. Langroudi 等 [4] 比较了使用 Posit 和 Float 实现嵌入式的深度神经网络模型的方案, 结果表明在保持相同识别精度的情况下, Posit 可以节省 2 bit 的编码长度. Johnson [5] 把 Posit 数制和对数域的乘加运算结合, 发现与 Integer8 的量化方法相比, 能够更好地保持模型精度, 不需要重新训练, 与 Float16 相比, 在 28 nm 的设计工艺下可以节省 40% 的功耗和 30% 的面积.

Posit 是一种能够使用较小转换熵完成数值计算的编码设计, 但是要想在深度学习芯片中发挥作用, 还需要在整个芯片的计算架构上进行创新.

4 展望深度学习专用计算架构

以深度神经网络芯片服务物联网市场是芯片行业从业者的一个历史机遇. 50 多年来受摩尔定律驱动, 集成电路产业有 3 次主要的通用计算的浪潮, 分别是英特尔 X86 架构代表的个人电脑 CPU, 以 ARM 公司 ARM 架构代表的移动互联网 CPU, 和近年来兴起的应用于物联网的神经网络芯片. 随着摩尔定律慢慢终结, 依靠工艺进步获取性能优势的机会不复存在, 产业界必须依靠计算体系结构创新来满足市场对性能和功耗的需求. 深度神经网络是一个通用型、计算密集型的计算范式, 引起了国内外众多学术机构、初创公司和行业巨头的重视. 下面按照芯片设计选择的内在逻辑分析深度学习专用计算架构的可能演化方向.

如图 5 所示, 在芯片的族谱上, 从 CPU 到 GPU, FPGA, DSP, ASIP, CGRA, DSA, ASIC, 通用性依次降低, 专用性依次升高, 能耗依次降低, 性能依次升高, 性能功耗比也是依次升高. 从编程的方式来看, CPU, GPU, DSP 和 ASIP 属于一个技术族, 它们共同的特点是根据一个程序指针 PC 进行时序编程; FPGA 和 CGRA 属于一个技术族, 它们共同的特点是对某种可编程的最小单元进行空间编程; DSA 和 ASIC 属于一个技术族, 它们共同的特点是根据应用设计特定的空间功能和时序功能, 在大粒度上对空间和时序进行组合. 在芯片领域, 性能和功耗不是绝对的. 更多的芯片资源能够堆砌出更高的性能, 其代价是更高的能耗和成本. 因此性能功耗比是比较公平的比较基准, 一般认为 GPU 和 FPGA 比 CPU 在性能功耗比方面有一个数量级的提升, DSP 和 ASIP 比 GPU 和 FPGA 有一个数量级的提升, CGRA, DSA 和 ASIC 比 DSP 和 ASIP 又有一个数量级的提升. 芯片行业一直在通用和专

用之间平衡取舍, 通用意味着更灵活的编程和更大的市场能够摊薄更低的成本, 专用意味着更强的性能带来更好的用户体验.

下面按照这个族谱做简要分析, 并着重分析每个族中性能功耗比最佳的方案. 在时序编程一族, GPU 是目前神经网络算法的主流实现方法, ASIP 则被很多初创公司选为技术路线. ASIP 为深度神经网络设计专门指令, 有明显的进步, 但是它很难跳出该族固有的一些限制, 如 Cache 争抢、循环展开的指令开销等等. 在空间编程一族, FPGA 因其灵活的硬件映射方案, 在某些特定的算法上有很强的功耗优势, 但是它可编程难度太大, 与设计一款芯片的难度相当, 注定是一个阶段性的解决方案. CGRA 改进了 FPGA 的可编程基础单元, 从逻辑运算上升到算术运算, 极大地提高了算术电路实现效率, 保持了阵列编程的灵活性, 因而实现的能耗效率远高于 FPGA, 也高于 ASIP, 但是它的编程方法比较复杂, 也没有现成的生态系统.

DSA 提取特定应用领域中可重复利用的空间结构和时间序列, 组织成电路结构. 从原始高维度张量数据中提取符号信息, 是深度学习要解决的人工智能任务, 这决定了适用于神经网络的计算结构是 TPU. 传统处理器的基础数据单位是标量, 最多在 SIMD 指令中扩展成向量, 它的指令是寄存器之间各种操作, 种类很多. 深度神经网络计算中的基础数据单位是张量, 张量之间进行的操作是有限的若干种. 前者是小数据, 操作种类多, 后者是大数据, 操作种类少, TPU 正是认识到这个关键区别, 实现了对神经网络计算结构的优化设计. 在 TPU 操作的具体实现上, 又可以借鉴 CGRA 的思路, 使用可重构的计算单元, 完成不同的操作.

综上分析, 我们可以得出以下结论:

(1) CPU, GPU, FPGA 和 DSP 性能功耗比不占优势, 并不是技术路线的首选, 但已经在各自的领域发展多年, 行业龙头已然形成. 专门固化的 ASIC 也不是可选方案.

(2) ASIP, CGRA 和 DSA 性能功耗比明显提升一个数量级, 属于新兴的技术方向, 是技术路线合理的选择, 但市场格局尚未形成.

(3) ASIP 过于保守, 未能突破时序编程的框架, 未来性能提升有很大的瓶颈; CGRA 过于激进, 编程方法和生态不够成熟, 风险较大; DSA 从神经网络计算的实际需求出发, 以张量为核心设计计算结构, 在粗粒度上保持可编程性, 兼顾了效率和灵活性, 是最合理的选择.

5 总结

本文首先讨论了深度学习在人工智能发展的阶段和任务, 由此引出它对计算结构的需求本质, 然后从 3 个方面讨论了适用于深度学习的领域专用计算架构. 近年来深度学习的算法和硬件实现层出不穷, 但是始终没有一个适当的深度学习计算结构的评价理论, 本文首次提出从熵的角度解释深度学习, 提供了计算结构评价的一个新视角. 深度学习近年来的发展模糊了数制的界限, 这使得有机会使用新的数制基础重新构造计算机体系架构. 最近出现的 Posit 数制不仅可以为深度学习的发展提供新的数制基础, 也可以替代现有的浮点编码方案为科学计算提供新的数制基础, 形成计算芯片设计的后发优势. 综合各方面因素, 针对深度学习领域专用计算架构最有可能成功的方向在哪里, 本文全面梳理了编程的时空范式, 并认为深度学习的专用架构应该朝着粗粒度时空编程的方向发展.

致谢 感谢 John L. GUSTAFSON 教授专门为本文制作了图 4.

参考文献

1 Ma L W. Intel Corporation. Method and apparatus for a binary neural network mapping scheme utilizing a gate array architecture. PCT/CN2016/112721. https://patentscope2.wipo.int/search/en/detail.jsf?docId=WO2018119785

2 Gustafson J, Yonemoto I. Beating floating point at its own game: posit arithmetic. J Supercomput Front Innov, 2017, 4: 71–86

3 Lindstrom P, Lloyd S, Hittinger J. Universal coding of the reals: alternatives to IEEE floating point. In: Proceedings of the Conference for Next Generation Arithmetic. New York: ACM, 2018

4 Langroudi S H F, Pandit T, Kudithipudi D. Deep learning inference on embedded devices: fixed-point vs posit. 2018. ArXiv: 1805.08624

5 Johnson J. Rethinking floating point for deep learning. 2018. ArXiv: 1811.01721

Domain-specific architectures driven by deep learning

Liwei MA

BitMain Technologies Holding Company, Beijing 100192*, China*
E-mail: Liwei.Ma@bitmain.com

Abstract Deep learning (DL) is one of the most exciting progresses in the field of artificial intelligence (AI); moreover, its new computational demands are driving new architecture researches. This paper firstly points out DL requirement essence by analyzing the stage and tasks in AI development, then discusses DL domain-specific architectures (DSAs) from three perspectives, which are the criteria of computational structures, the basics of a numerical system for computation, and DL DSA potential research directions. Furthermore, herein, the Kullback-Leibler divergence was utilized as the criteria for DL computation architecture complexity and accuracy. Besides, Posit was employed as a new number system to rebuild DL computation and scientific computation and to establish the late-development advantage of digital chips. Finally, it was concluded that DL DSAs are one of the critical DSA research areas.

Keywords deep-learning, entropy, cross-entropy, numerical system, computing architecture

Liwei MA was born in 1979. He received his Ph.D. degree in electronics engineering in the 2007 from the Tsinghua University, Beijing, China. He is now a product marketing manager at Bitmain. His academic and industrial interests include RISC-V based DSAs, deep-learning acceleration, and AI applications.