机器学习在计算机体系结构设计中的应用综述

美国电气和电子工程师协会高级会员

摘要——机器学习已经在不同的领域带来了显著的好处，但是，除了少数例外，对计算机体系结构的影响有限。然而，最近的工作探索了设计、优化和模拟的更广泛的适用性。值得注意的是，基于机器学习的策略经常超越现有的最先进的分析、启发式和人类专家方法。本文回顾了机器学习在全系统模拟和运行时优化中的应用，以及在许多单个组件中的应用，包括内存系统、分支预测器、片上网络和图形处理器。该文件进一步分析了当前的实践，以突出有用的设计策略，并根据优化的实施策略、现有工作的适时扩展和雄心勃勃的长期可能性，确定未来工作的领域。总的来说，这些策略和技术为日益自动化的建筑设计提供了一个充满希望的未来。

1 导言

在过去的十年中，机器学习(ML)在许多领域迅速成为一个革命性的因素，从商业应用(如自动驾驶汽车)到医疗应用，改善了疾病筛查和诊断。在这些应用中的每一个中，通过发现数据中嵌入的模式或关系，ML 模型被训练来进行预测或决策，而无需显式编程。值得注意的是，ML 模型可以在任务/应用程序中很好地执行，在这些任务/应用程序中，关系过于复杂，无法使用分析方法进行建模。这些强大的学习能力继续推动着不同领域的快速发展。与此同时，摩尔定律预测的指数增长已经放缓，给建筑师带来了越来越大的负担，要求他们用建筑的进步来取代摩尔定律。这些相反的趋势表明了范式转变的机会，在这种转变中，计算机体系结构支持最大似然，同时，最大似然改进了计算机体系结构，为这两个领域关闭了一个具有巨大潜力的正反馈回路。

传统上，计算机体系结构和最大似然算法之间的关系相对不平衡，侧重于体系结构优化以加速最大似然算法。事实上，人工智能研究最近的复苏至少部分归功于处理能力的提高。这些改进通过利用可用并行性、数据重用、稀疏性等的硬件优化得到了增强。在现有的 ML 算法中。相比之下，应用 ML 来改进架构设计的工作相对有限，分支预测是少数主流的例子之一。这项新生的工作，虽然有限，但为建筑设计提供了一个吉祥的方法。

本文概述了最大似然法在建筑设计和分析中的应用。如图所示1，这一领域的成功和受欢迎程度都有了显著增长，尤其是在过去几年。这些作品确立了 ML 使能建筑设计的广泛适用性和未来潜力；现有的基于 ML 的方法，从具有简单分类树的 DVFS 到通过深度强化学习的设计空间探索，已经超过了它们各自最先进的基于人类专家和启发式的设计。基于最大似然的设计可能会

*对应作者。电子邮件：chenliz@oregonstate.edu

作者来自俄勒冈州大学科尔瓦利斯分校，

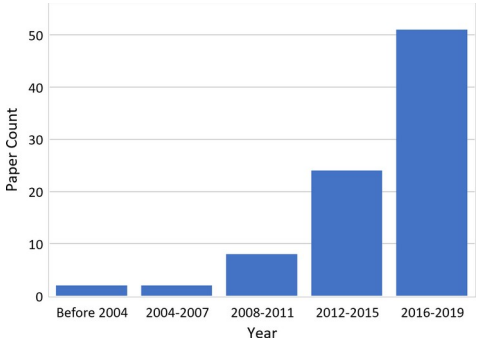

图 1。关于机器学习应用于建筑的出版物(对于在第节中检查的作品 3)

随着有前景的应用被探索，继续提供突破。

论文组织如下。第 2 节提供了关于 ML 和现有模型的背景，以建立 ML 对架构问题适用性的直觉。第 3 节介绍了应用于架构的 ML 的现有工作。第 4 节比较和对比了现有工作中的实施策略，以突出重要的设计考虑。第 5 节确定了现有工作的可能改进和扩展，以及未来工作的有希望的新应用。最后，第六节总结。

2 背景

2.1 基本适用性

机器学习作为一种解决各种问题的替代方法，在许多领域得到了迅速的应用。这种基本的适用性源于最大似然算法强大的关系学习能力。具体来说，ML 模型利用了一个通用框架，在这个框架中，这些模型从示例中学习，而不是从先前的编程中学习，使得应用程序能够在许多任务中运行，包括那些难以用标准编程方法表示的任务。此外，使用这个通用框架，对于任何给定的问题，可能有许多可能的方法。例如，在预测处理器的 IPC 的情况下，可以使用简单的线性回归模型进行实验，该模型学习特性(例如核心频率和高速缓存大小)之间的线性关系

和每周期指令。这种方法可能有效，也可能无效。如果效果不好，可以尝试不同的特征、非线性特征组合(如核心频率乘以缓存大小)，或者完全不同的模型，另一个常见的选择是人工神经网络。这种可能方法的多样性使得能够调整模型、模型参数和训练特征以匹配手头的任务。

2.2 学习方法和模式

学习方法和模型都是将机器学习应用于任何问题的基本考虑因素。一般来说，有四种主要的学习方法:监督学习、非监督学习、半监督学习和强化学习。这些方法可以通过使用什么数据以及如何使用这些数据来促进学习来区分。类似地，对于给定的问题，可能存在许多适当的模型，从而基于学习方法、硬件资源、可用数据等实现应用的显著多样性。在下文中，我们将介绍这些学习方法以及每种学习方法的几个重要模型，重点介绍已经证明适用的方法。实施细节将在后面的章节中讨论 4。

监督学习:在监督学习中，使用输入特征和输出目标来训练模型，结果是一个可以预测新的、看不见的输入的输出的模型。常见的监督学习应用程序包括回归(预测一个值，如处理器 IPC)和分类(预测一个标签，如应用程序执行的最优核心配置)。功能选择，在第节中讨论 2.3，在这些应用中尤为重要，因为模型必须学会仅基于特征值进行预测。

监督学习模型可以概括为四类:决策树、贝叶斯网络、支持向量机和人工神经网络[1].决策树使用树结构，其中每个节点代表一个要素，分支代表该要素的一个值(或值的范围)。因此，根据给定节点上考虑的特征值，通过顺序跟随分支对输入进行分类。相反，贝叶斯网络将条件关系嵌入到图形结构中；节点代表随机变量，边代表这些变量之间的条件依赖关系。例如，性能预测模型可以根据其他应用程序中未观察到的变量(即影响性能的潜在因素)的已学习分布，对新应用程序进行条件预测，如[2].支持向量机通常以其功能而非特定的图形结构而闻名(如决策树和贝叶斯网络)。具体来说，支持向量机学习例子之间的最佳分割线(二维)或超平面(高维)，然后沿着这个超平面使用例子进行新的预测。支持向量机也可以用核方法扩展到非线性问题[3]以及多类问题。最后，人工神经网络(或简称神经网络)代表了一大类模型，这些模型又是由它们的结构定义的，让人想起人脑中的神经元；节点/神经元层使用具有学习权重的链接进行连接，使得特定节点能够响应特定的输入特征。简单的感知器模型只包含一个权重层，直接将输入的加权和转换为输出。更复杂总数。卷积神经网络(CNNs)等其他变体在某些层之间结合卷积运算来捕获空间局部性，而递归神经网络则重复使用以前的输出来学习序列和长期模式。所有这些监督学习模型都可以用于分类和回归任务，尽管存在一些明显的高级差异。各种支持向量机和神经网络往往对高维连续特征表现得更好，当特征可能是非线性时也是如此[1].然而，与贝叶斯网络和决策树相比，这些模型往往需要更多的数据。

无监督学习:无监督学习仅使用输入数据提取信息，无需人工努力。因此，这些模型可能是有用的，例如，通过找到适当的替代表示或把数据聚集到对人类来说不明显的类中来降低数据维度。

迄今为止，应用于建筑领域的两种主要的无监督学习模型是主成分分析和 k-均值聚类。主成分分析通过确定具有高方差的线性特征组合，提供了一种从数据集中提取重要信息的方法[4].因此，主成分分析可以作为构建降维模型的第一步，这是大多数应用中非常需要的特性，尽管代价是可解释性(在第节中讨论 4).取而代之的是使用 k 均值聚类来识别具有相似特征的数据组。这些组可以进一步处理，以概括行为或简化大型数据集的表示。

半监督学习:半监督学习代表监督和非监督方法的混合，有一些成对的输入/输出数据，和一些不成对的输入数据。使用这种方法，学习可以利用有限的标记数据和潜在的重要的未标记数据。我们注意到，到目前为止，这种方法还没有在架构中得到应用。然而，有一项关于电路分析的工作[5]提出了一个可能的战略，可在今后的工作中加以调整。

强化学习:在强化学习中，根据环境状态向代理依次提供输入，并学习执行优化奖励的操作。例如，在内存控制器的环境中，代理取代了传统的控制逻辑。输入可以包括未决的读取和写入，而操作可以包括标准的存储器控制器命令(行读取、写入、预充电等)。).然后，可以通过将吞吐量包含在奖励函数中来优化吞吐量。在这种设置下，随着时间的推移，代理可能会学会选择最大化吞吐量的控制操作。

应用于架构的强化学习模型，作为一个整体，可以使用基于状态、动作和奖励的表示来理解。代理试图学习一个策略函数 π，它定义了在给定状态 s 下，基于收到的奖励 r [6].遵循策略的学习状态值函数给出如下

$$v_{\pi}(s) = e\left[\sum_{t \geq 0} \gamma^{t} r_{t} \mid s_0 = s, \pi\right] \quad (1)$$

其中 γ 是贴现因子(1)，它规定了模型应该考虑未来回报的多少。然后，通过学习满足以下条件的最优策略 π*来最大化累积回报

$$\pi^{*}(s) = \arg\max_{\pi} e\left[\sum_{t} \gamma^{t} r_{t} \mid s_0 = s, \pi\right]。 \quad (2)$$

各种模型可能会采用不同的方法来学习这种最优策略，但在很大程度上解决了相同的最大化回报的问题。Q-learning 是一个值得注意的例子，它通过从给定状态估计单个动作的值来模拟动作值函数。

2.3 特征选择

监督(和半监督)学习方法依赖输入数据特征来建模关系和生成预测。因此，用于特征选择的方法会显著影响模型性能，包括诸如过度拟合和计算开销之类的问题，以及诸如特征可解释性之类的更抽象的问题。在一些作品中，特征选择完全基于专家知识。另外，更一般的方法可以取代或补充专家知识。

一组方法，称为过滤方法，使用涉及统计相关或信息论标准(如互信息)的度量来单独考虑特征。这些方法通常被认为是最不密集的，因此对于非常大的特征集可能是优选的，但是模型性能可能是次优的，因为过滤方法中的评估标准不考虑特征上下文[7];单独提供很少益处的两个特征可能一起有益。因此，许多替代方法考虑特征子集。

包装方法通过直接评估学习模型的性能，为特征选择提供了黑盒方法[7].常用的贪婪方法包括向前选择和向后消除。在正向选择中，基于对整体学习模型的改进，特征被逐步添加到所选特征子集。相反，向后消除会逐渐移除几乎没有好处的功能。

嵌入式方法将特征选择集成到学习模型中，以提供过滤器和包装方法之间的折衷[8].正则化是一种广泛使用的嵌入方法，它允许学习模型被拟合，同时迫使特征系数变小(或为零)。然后可以移除具有零系数值的特征。这种方法消除了包装器方法中存在的迭代特征选择，包装器方法可能有很高的计算要求[7].

3 文献评论

本节回顾了将机器学习应用于架构的现有工作。工作由子系统(如适用)或主要目标组织。我们专注于设计和优化，但也介绍一般的性能预测工作。

3.1 系统模拟

周期精确模拟器通常用于系统性能预测，但需要比本机执行多几个数量级的时间。ML 可以通过在模拟时间和精度之间进行权衡来抵消这种损失。一般来说，ML 可以将执行时间减少 2-
3 精度相对较高的数量级(取决于任务，通常> 90%)。伊佩克等人的早期工作[9]使用人工神经网络集合(一组人工神经网络预测器)对建筑设计空间进行建模。模型在大约 1%的设计空间上进行训练，然后预测随机点的 CMP 性能，误差为 4-5%，尽管只是在特定配置空间中。当与简单点结合时，预测显示出稍高的误差，但模拟指令数进一步减少。Ozisikyilmaz 等人[10]此外，还预测了未来系统的 SPEC 性能，现有模拟器可能无法很好地模拟这些系统。评估仅限于使用相对简单的线性回归和神经网络模型的随机抽样数据，但与标准单层模型相比，修剪神经网络显示出优势(如[9]).其他几种 ML 方法也已经过测试。艾克曼等人[11]提出了一个用于处理器 CPI 预测的机械经验模型。在本研究中，他们使用了一个通用的机械模型，其参数由回归模型推断。他们的模型仅限于单核性能预测，但提高了准确性、易实现性(与纯机械模型相比)和可解释性(与纯经验模型相比)。郑等[12]，[13]探索了使用线性回归从英特尔/AMD 到 ARM 处理器的跨平台预测。他们的第一种方法 12]基于目标点周围示例的局部邻域进行预测，以近似非线性行为。他们后来 13 假设相级行为近似线性，估计相级预测。值得注意的是，使用阶段级分析，周期计数预测的平均误差小于 1%。然而，这种方法仅限于单个目标架构，并且需要源代码进行阶段级分析，为未来的工作留下了大量的机会。最后，Agarwal 等人最近的工作[14]介绍了一种利用单线程执行特性预测并行执行加速比的方法。他们使用应用程序级性能计数器为每个线程计数训练单独的模型。尽管由于数据有限而省略了神经网络，但评估发现高斯过程回归仍然提供了有希望的结果，尤其是对于高线程数。

3.2 绘图处理器

设计空间探索:由于设计空间高度不规则，GPU 设计空间探索已被证明是 ML 特别有利的应用；一些内核表现出相对线性的缩放，而其他内核则表现出配置参数、功率和性能之间非常复杂的关系[15]，[16]，[17].贾等[15]提出了 Stargazer，这是一个基于自然三次样条的回归框架。Stargazer 为每个应用程序从目标设计空间随机采样大约 300 个点(评估中为 933 千个点)，然后对这些点应用逐步回归。值得注意的是，该框架实现了低于 3.8%的平均性能预测误差。吴等。
[16]而是为计算单元、内核频率和内存频率显式建模缩放。使用 k-均值聚类处理来自训练核的缩放数据，以通过缩放行为对核进行分组。然后，人工神经网络将内核分类到这些集群中，允许使用集群缩放因子对新内核进行分类和预测。这种方法与贾等人的方法相反[15】，因此新应用只需要几个样本。Jooya 等人[17】，类似于贾等[15]，考虑了每应用性能/功率预测模型，但另外提出了预测每应用帕累托前沿的方案。许多基于人工神经网络的预测器被训练，最精确的子集被用作预测的集合。随后，通过在先前预测的帕累托最优曲线的阈值内采样点，提高了预测精度。

林等[18]将性能预测 DNN 与遗传搜索方案相结合，探索记忆控制器的位置。DNN 被用作替代适应度函数，避免了缓慢的系统模拟。由此产生的布局将系统性能提高了 19.3%。

跨平台预测:移植应用程序以在图形处理器上执行是一项具有挑战性的任务，与中央处理器执行相比，它可能具有不确定的优势。因此，我们已经研究了仅使用 CPU 执行行为来预测加速或效率提高的方法。Baldini 等人[19]将问题转化为分类任务，训练改进的最近邻和支持向量机(SVM)模型，以基于阈值确定 GPU 实现是否有益。使用这种方法，他们在 91%的时间里预测接近最优的配置。相比之下，Ardalani 等人[20]训练了一大群回归模型来直接预测代码段的 GPU 性能。尽管几个代码段显示出高误差，但相对误差绝对值的几何平均值仍然仅为 11.6%，并且该模型成功地识别了几个被人类专家错误预测的代码段(有益的和不有益的)。Ardalani 等人的后期工作[21]引入了一个完全基于静态分析的框架，使用随机森林模型进行二进制分类。这种方法消除了动态分析和人为指导，而是使用指令混合、分支发散估计和内核大小等特性来为二进制加速分类提供 94%的准确性(使用加速阈值 3)。

GPU 特定预测与分类:奥尼尔等。[22]介绍了一种用于 DirectX 应用的下一代图形处理器性能预测方法，即每帧周期(CPF)。他们专注于英特尔图形处理器，分析早期架构(如哈斯韦尔 GT2)来训练下一代预测器。他们发现，不同的模型(即线性与非线性)可以根据预测目标产生更准确的结果(布罗德韦尔 GT2/GT3 vs 斯凯莱 GT3)，表现最好的模型实现的 CPF 预测误差小于 10%。李等近期工作[23]对 GPU 流量模式的公认知识进行了重新评估。他们在热图转换的交通数据上使用了美国有线电视新闻网和 t 型分布的随机邻居嵌入，以 94%的准确率识别了八种独特的模式。

调度:图形处理器内存处理(PIM)架构可以受益于高内存带宽和减少的数据移动能量。尽管有这样的好处，但是当在各种资源上调度执行时，PIM 计算能力的潜在限制可能会在性能和能量之间引入复杂的权衡。为此，Pattnaik 等人[24]提出了一种方法，该方法使用回归模型对核心亲缘关系进行分类，从而划分工作负载，并提出了一种额外的回归模型来预测执行时间，从而实现动态任务迁移。性能和能效分别比基准图形处理器架构提高了 42%和 27%。通过提高核心相似性分类的准确性(与回归相比)，进一步的改进是可能的。

3.3 存储系统和分支预测

高速缓存:启发式的高速缓存方法会因为巨大的工作负载差异而导致性能下降。ML 方法可以学习这些错综复杂的东西，并提供卓越的性能。Peled 等人[25]提出了一种预取器，该预取器使用上下文标签(一种简单的 RL 变体)来探索语义局部性(数据结构)，将上下文信息和候选地址相关联以进行预取。实现使用两级索引方法动态控制状态信息，允许在线特征选择，但会有一些额外的开销。曾与郭[26]提出了一个长短期内存(LSTM)模型(递归神经网络变体)，用于基于本地历史和偏移增量表进行预取。评估表明，与以前的工作相比，LSTM 模型能够在更长的序列和更高的抗噪性上进行准确预测。解决了与开销和预热时间相关的几个问题，并为未来的工作留下了潜在的解决方案。同样，布劳恩等人[27]广泛探索了几种常见访问模式下的 LSTM 预取准确性。实验考虑了回看大小(访问历史窗口)和 LSTM 模型大小对几个噪声水平和独立访问流计数的影响。Bhatia 等人最近的工作[28]将传统预取器与基于感知机的预取过滤器合成，允许积极的预测，而不会降低准确性。评估证实了所提议的方案提供的大量覆盖和 IPC 优势，当参考无预取四核基线时，IPC 比次优预取器加速 9.7%。ML 同样被应用于数据重用策略。例如，特兰等人[29]用感知器模型预测 LLC 重用。在这种方法中，输入特征被散列以访问基于正确/不正确的重用预测而递增/递减的饱和权重表。这些特性是根据经验选择的，并且会对性能产生显著影响，因此为进一步优化提供了一个选项。王等[30]在缓存条目之前预测重用，只有在预测重用的情况下才在缓存中存储数据。他们使用决策树作为集成模型的低成本替代品，实现了 60- 80%的写入减少。额外的研究探索了翻译后备缓冲区(TLBs)中不断增长的性能瓶颈。Margaritov 等人[31]提出了一种基于学习索引的 TLB 虚拟地址转换方案[32].评估显示，预测指数的准确率接近 100%，但实际实施将需要专用硬件来减少计算开销(并留待未来工作)。

调度器和控制:内存和存储系统的控制器会影响设备性能和可靠性，因此与启发式算法相比，它代表了 ML 模型的另一个强大应用。伊佩克等人[33]首先为内存控制器提出了一种 RL 方法，以获取并发性、延迟和其他几个因素之间的平衡。该模型预测了最佳操作(预充电、激活、行读/写)，与之前的工作相比，系统性能提高了 15%(在双通道系统中)。穆昆丹和马丁内斯[34]后来在伊佩克工作的基础上，推广了奖励函数以优化能量、公平等。他们还增加了上电和掉电操作，进一步提高了 8.6%的性能和能效。相关工作使用 ML 优化了内存/存储和其他系统之间的通信能量。Manoj 等人[35]提出了一种用于硅中介层传输线动态电压摆幅控制的 Q 学习方法。对功率和误码率的预测被量化，然后作为输入提供给模型以确定新的电压电平。尽管他们的方法需要大量的量化来最小化

尽管开销很大，但与静态电压基线相比，它们仍然实现了15.1%的节能。王和[36]通过在线聚类和编码减少数据移动能量。几个集群使用该集群中数据的多数投票在位级持续更新。然后，通过将新数据与最近的学习群集中心进行异或运算，将传输的1总数降至最低。康和刘[37]应用Q-learning，通过确定最佳不活动时段来管理固态硬盘中的垃圾收集。使用LRU替换将关键状态保存在Q表中，从而获得了巨大的状态空间，并最终比基线平均减少了22%的尾部延迟。然而，许多状态在每个工作负荷中只观察到一次，这表明使用深度Q学习的潜在好处(DQL)。其他直接考虑系统可靠性的工作。例如，邓等[38]提出了一个基于回归的框架来动态优化非易失性存储器的性能和寿命。他们的方法使用基于阶段的应用程序统计来管理几个冲突的策略，如写延迟、写取消、持久性等。，保证最低的使用寿命，并适度提高性能/能耗。肖等[39]提出了一种使用在线随机森林预测磁盘故障的方法。他们使用磁盘状态窗口来训练他们的模型，以说明记录的故障日期的不精确性，从而能够准确预测即将出现故障的驱动器。与其他随机森林更新方案的比较(例如，每月更新一次)突出了一致性训练的准确性优势，这些优势可能会扩展到相关领域。

分支预测:分支预测是当前最大似然估计在工业中应用的一个值得注意的例子，其准确性超过了现有的最先进的非最大似然预测器。基于感知器的分支预测器最早是由和林提出的40]作为使用模式历史表的两级方案的有前途的高精度替代方案。St. Amant等人后来的研究引入了SNAP [41】，这是一种基于感知器的预测器，使用模拟电路实现，以实现高效且实际可行的设计。感知器权重和分支历史用于驱动电流控制DAC，该DAC将点积作为电流之和。姬广亮内斯[42]使用每个分支的历史表、历史重要性的动态系数和动态学习阈值进一步优化了该设计。优化设计实现了比L-TAGE低3.1%的MKPI。Garza等人最近在基于感知机的预测器方面的工作[43]探索了间接分支的位级预测。可能的分支目标使用它们的相似性(点积)和结合本地和全局历史的八个特征表的组合权重进行评估，最终与ITTAGE相比将MKPI降低了5%。目前，最先进的条件分支预测器(例如，TAGE- SC-L [44])仍然在几个难以预测(H2P)的条件分支中隐藏了显著的IPC增益(英特尔Skylake架构为14.0%)[45].Tarsa等人[45]因此提出了“有线电视新闻网助手”预测器，使用简单的两层有线电视新闻网针对特定的H2Ps。结果表明，它可以很好地应用于各种不同的工作负载，并为未来的工作提供了一个充满希望的领域。

3.4 片上网络

DVFS &链路控制:现代计算系统采用复杂的功率控制方案，以支持日益并行的架构设计。启发式方案可能无法利用所有节能机会，尤其是在动态片上网络(NoC)工作负载中，从而导致通过基于ML的主动控制获得显著优势。Savva等人[46]使用多个人工神经网络实现了动态链路控制，每个人工神经网络监控一个片上网络分区。这些人工神经网络仅使用链路利用率来学习启用/禁用链路的动态阈值。尽管节省了能源，但他们的方法在维度有序路由下会导致高延迟。DiTomaso等人[47]将微片缓冲器重新定位到链路，并通过每路由器分类树动态控制链路方向和功率门控。与集中网状网络相比，使用简单的三级树来限制开销，整体NoC功率降低了85%，延迟降低了14%。Winkle等人[48]探索了光子互连中基于ML的功率缩放。即使是一个简单的线性回归模型也提供了有希望的结果，可以忽略不计地降低吞吐量(相对于没有功率门控)，同时将激光功耗降低42%。Reza等人[49]提出了一种多级人工神经网络控制方案，该方案同时考虑了任务分配、链路分配和节点DVFS的功率和热约束。单个人工神经网络为本地网络控制分区分类合适的配置，而全局人工神经网络分类最优的整体资源分配。该方案以高精度(88%)识别全局最优片上网络配置，但使用了可能影响实施的复杂人工神经网络。克拉克等人[50]提出了一种适用于DVFS的路由器设计，并评估了几种基于回归的控制策略。变量预测缓冲区利用率、缓冲区利用率的变化或能量和吞吐量的组合度量。费特等人扩展了这项工作51】，他介绍了一种RL控制策略。回归模型和RL模型都支持有益的权衡，尽管RL策略最灵活。

准入和流量控制:与DVFS国家奥委会一样，准入和流量控制都可以从积极的预测中受益。伯颜和利特曼的早期作品52]在网络中引入了基于Q学习的路由，使用来自相邻节点的传递时间估计，注意到在高流量强度下，吞吐量优于传统最短路径路由。有几项工作扩展了Q路由，观察了在动态变化的片上网络拓扑中的应用53]，改进了无缓冲NoC容错路由的功能[54]，以及高性能拥塞感知非最小路由[55].最近的工作集中在注入节流和热点预防上。例如，Daya等人[56]提出了SCEPTER，一种使用单周期多跳路径的无缓冲NoC。他们使用Q学习来控制注入节流，以最大化多跳性能，并通过减少竞争微片来提高公平性。未来的工作可能会减少Q-table over head，这种over head在其实现中随片上网络的规模而扩展。王等[57]而是使用人工神经网络来预测标准缓冲NoC的最佳注射速率。额外的预处理(捕捉空间和时间趋势)和节点分组实现了高精度预测(90.2%)，与未优化的基线相比，执行时间减少了17.8%。Soteriou等人[58]类似地探索了基于ANN的注入节流，以减少NoC热点。人工神经网络被训练预测热点，同时识别建议的注入节流和动态路由的影响，提供整体缓解策略。该模型为合成流量下的吞吐量和延迟提供了最先进的结果，但在真实基准下的改进有限，这表明了进一步优化的潜力。尹等提出的另一种Q学习方法[59]，利用DQL仲裁国家奥委会的流量。他们考虑了广泛的功能和奖励，同时指出

由于过头，所提出的DQL算法是不切实际的。无论如何，与循环仲裁相比，评估显示吞吐量略有提高。

拓扑和总体设计:一些作品也将ML应用于更高级别的片上网络拓扑设计，涉及功率和性能之间的权衡，一些作品进一步考虑了热。Das等人[60]使用基于ML的STAGE算法高效地探索小世界的in-spire 3D NoC设计。在这种方法中，设计在基础/本地搜索(在爬山方法中添加/移除链接)和元搜索(使用先前结果预测本地搜索的有益起点)之间交替。Das等人再次使用了相同的模型[61]平衡链路利用率并解决TSV可靠性问题。Joardar等人改进了STAGE算法62]优化异构3D NoC设计。这些模型探索了中央处理器延迟、图形处理器吞吐量和热/能量约束之间的多目标权衡。三部作品[60], [61], [62]仍然依靠爬山来优化。林等近期工作[63]相反，我们探索了无路由器NoC设计中的深度强化学习。他们使用蒙特卡罗树搜索来有效地探索搜索空间，并使用深度卷积神经网络来逼近动作和策略函数，从而优化环路配置。此外，所提出的深度强化学习框架可以严格执行可能被先前的启发式或进化方法违反的设计约束。Rao等人[64]研究了广泛的SoC特性空间(从带宽要求到SoC区域)中的多目标NoC设计优化。ML模型使用来自数千个片上系统配置的数据进行训练，以预测基于性能、面积或两者的最佳片上系统设计。与人类专家设计的有限比较没有考虑替代技术(例如AMOSA [65])，但展示了一些有希望的结果，激发了对有效特征和模型的研究，以及与替代技术的进一步比较。

性能预测:现有的基于排队论的NoC模型通常是准确的，但是依赖于对流量分布的估计，这可能不适用于实际应用[66].钱等[66]强调了基于ML的方法如何放松排队论模型所做的假设。他们构建了一个基于通信图的机械经验模型，使用支持向量回归(SVR)将几个特征和排队延迟联系起来。评估显示，与现有分析方法相比，误差更低(3%误差对10%误差)。Sangaiah等人[67]考虑了用于性能预测和设计空间探索的NoC和内存配置。按照标准方法，他们对设计空间的一小部分进行采样，然后训练回归模型来预测最终的系统消费价格指数。评估通常显示较高的准确性，但对于高流量工作负载准确性较低(中位误差为24%)。额外的设计空间探索展示了有希望的结果，将设计空间从2.4M点减少到1000点以下。

可靠性和纠错:在片上网络中纠错带来的开销可能很大，尤其是在需要重新传输时。因此，一些工作已经探索了基于ML的控制方案。DiTomaso等人[68]训练了一个决策树，使用包括温度、利用率和设备损耗在内的各种参数来预测NoC故障。这些预测允许对可能有错误的传输进行主动编码(在基线循环冗余校验的基础上)。王等[69]采用了类似的动态错误缓解策略，但使用了基于RL的控制策略，以消除对标记训练示例的需求。他们的方法平均节省了46%的动态功耗(比决策树方法好17 %[68])与静态CRC方案相比。在这两种情况下，基于最大似然的主动控制都选择了比仅使用循环冗余校验更有效的方案。王等[70]随后提出了一个整体的NoC设计框架，包括动态错误缓解、路由器电源门控和多功能自适应信道缓冲器(MFAC缓冲器)。他们强调通过多种架构创新的协同集成/控制带来的综合优势，从而与SECDED基准相比，在延迟(32%)、能效(67%)和可靠性(平均故障时间提高77%)方面实现了显著改善。

3.5 系统级优化

能效优化:必须考虑工作负载执行受总能耗而非处理资源约束的系统，这是一项重要的工作。采用最大似然控制的控制方案在通过最小性能降低来优化能效方面显示出了希望，与争用到空闲方案相比，通常能够将能量延迟产品降低60-80%。Won等人[71]介绍了一种用于非核心DVFS的混合神经网络+比例积分控制器方案。他们首先离线训练神经网络，然后使用PI控制器在线优化预测。与单独的比例积分控制器相比，这种混合方案将能量延迟积降低了27%，与最高电压/频率水平相比，性能下降不到3%。Pan等人[72]使用多级RL算法实现了电源管理方案。他们的方法将单个核心状态传播到一个树形结构中，同时在每一层聚集Q学习表示。全局分配在根处进行，然后决策沿着树向下传播，实现有效的每核心控制。Bailey等人[73]解决了异构系统中的能效问题。与吴等相似16]中，他们根据核的缩放行为对核进行聚类，以训练多个线性回归模型。运行时预测使用两个示例配置来确定最佳配置，一个来自CPU执行，另一个来自GPU执行。Lo等人[74]专注于实时交互式工作负载的能效优化。他们使用线性回归来基于注释和代码特性对执行时间进行建模，从而在源代码不可用时以牺牲适用性为代价来实现更严格的服务级别保证。Mishra等人[75]还解决了实时工作负载，结合了控制理论和几个基于ML的模型。他们的框架是通过将学习卸载到服务器上来实现的，允许低开销的DVFS，与以前最好的方法相比，降低了13%的能耗。Mishra等人的相关工作[2]应用了一个相对复杂的分层贝叶斯模型来结合离线和在线学习。在这种方法中，他们接受了较高的执行时间惩罚(0.8秒)，以提供比单独在线或离线训练更精确的预测。因此，这种方法针对的是执行时间较长的工作负载，但与次优方法相比，可以节省24%以上的能源。白等[76]采用片外开关调节器和片内线性调节器，实现了基于RL的DVFS控制策略，适用于新型电压调节器体系。独立的反向链路代理适应由启发式竞价方法确定的动态分配的功率预算。这

使用自适应卡内瓦编码增强了设计[77]限制区域/电源开销和经验分享，以加速学习。陈和马库列斯库[78】(后来陈等人[79])为基于 RL 的 DVFS 探索了另一种两级策略。类似于白等人[76]，他们在细粒度核心级别使用 RL 代理，根据分配的全球电力预算份额选择 V/F 级别。他们通过使用考虑相对应用性能要求的性能感知型(尽管仍然是基于启发式)变体来分配功率预算，从而实现了进一步的改进。艾姆斯等人[80]探索了单应用系统能源优化，以获得更广泛的配置选项，包括插槽分配、超线程使用和处理器 DVFS。他们确定了几个有用的模型，同时指出进一步的工作可以优化模型和参数。分析还提供了对单模型多资源优化的益处的洞察，特别是对于神经网络。最后，Tarsa 等人最近的工作。
[81]考虑了一个 ML 框架，用于使用微控制器实现的模型的固件更新进行后硅 CPU 适配。针对人员盲点的重大调整限制了违反服务级别协议的比率，同时优化了通用和特定应用部署的性能功耗比。

任务分配和资源管理:除了能量控制，ML 还提供了一种方法，通过预测各种配置对长期性能的影响，将资源分配给任务或任务分配给资源。陆等[82]提出了一种用于多核任务分配的热感知 Q 学习方法。代理只考虑当前温度(即，没有应用程序分析或硬件计数器)，因任务分配而获得更高的奖励，从而获得更大的散热空间。估价
与启发式方法相比，峰值温度平均降低了 4.3℃。Nemirovsky 等人[83]
介绍了一种在异构体系结构上进行 IPC 预测和任务调度的方法。他们使用人工神经网络预测所有任务安排的 IPC，然后选择 IPC 最高的安排。突出评价
使用深度(但高开销)神经网络的显著吞吐量增益(> 1.3 倍)，表明了一种可能的应用
修剪操作(在第节中讨论 5.2).最近的工作还探索了混合 CPU-GPU 集群中的多级调度。张等[84]提出了深度强化学习(DRL)框架来划分视频工作负载，首先在集群级别(选择一个工作节点)，然后在节点级别(CPU vs GPU)。这两个 DRL 模型分别运行，但仍然一起优化整体吞吐量。将资源分配给任务是另一种可能的方法。Bitirgen 等人的早期工作[85]考虑了一个具有四个内核和四个并发应用程序的系统。在他们的方法中，每个应用的人工神经网络集成预测了每个间隔(500K 周期)2000 个配置的 IPC。然后聚合 IPC 预测，以选择性能最高的整体系统配置。在未来的工作中，可以解决每个应用程序集成的扩展问题和呈指数级增长的配置空间。re cent 研究还考虑了涉及多个组件/资源的低级别协同优化。例如，Jain 等人[86]探索了核心 DVFS、非核心 DVFS 和动态 LLC 分区的并发优化。这些选项由单个代理以相对较大的间隔(1B 指令)进行优化(可能会限制协同优化机会)。然而，评估表明，通过以下方式，能源延迟产品显著减少
多资源优化。最后，丁等人的工作。
[87]基于对数据稀缺性和模型偏差的改善，在模型精度和系统优化目标之间建立了某种矛盾的趋势。具体来说，他们发现最先进的模型在准确性方面的回报越来越少，反而受益于领域知识(例如，在最佳前沿集中采样)。

芯片布局:吴等作品[88]演示了 ML 在芯片布局中的用途，偏离了包括控制、预测和设计空间探索在内的常见应用。他们在物理布局过程中使用 k-means 对触发器进行聚类，以信号线长度为代价最小化时钟线长度，并指出时钟网络可以消耗超过 40%的芯片功率。它们包括对最大触发器位移和簇大小的限制，与现有技术方法相比，产生位移减少 28.3%、总导线长度减少 3.2%和总开关功率减少 4.8%的设计。

安全性:恶意软件检测，一个传统的基于软件的任务，已经被探索使用机器学习来实现可靠的基于硬件的执行中检测。例如，Ozsoy 等人[89]测试在低级硬件计数器上训练的逻辑回归和神经网络分类器。基于降低的精度和特征选择的优化以最小的开销(0.04%的核心功率和 0.19%的核心逻辑面积)为 LR 模型提供了高精度(100%的恶意软件检测和不到 16%的误报)。

3.6 支持最大似然的近似计算

近似计算有许多方面，包括电路级近似(如降低精度的加法器)、控制级近似(放松时序等)和数据级近似。使用最大似然法的方法通常属于最后一类，它提供了一种强大的函数/循环逼近技术，通常可以提供 2-3 倍的应用加速和能量降低，但对输出质量的影响有限。Esmaeilzadeh 等人[90]引入了 NPU，这是一种使用神经网络进行可编程逼近的新方法。他们开发了一个框架来实现 Parrot 转换，将注释代码段转换为神经网络近似。在所研究的应用中，将 NPU 与中央处理器紧密集成可以实现平均 2.3 倍的加速和 3.0 倍的节能。这个框架后来被 Yazdanbakhsh 等人扩展。
[91]在图形处理器上实现神经逼近。Neu- ral 近似被集成到现有的 GPU 流水线中，实现了组件重用、大约 2.5 倍的加速和 2.5 倍的节能。Grigorian 等人
[92]为多级神经加速器提出了一种不同的方法。输入首先通过相对低精度/开销的神经加速器发送，然后检查质量；可接受的结果被提交，而低质量的近似被转发到附加的、更精确的近似阶段。这些工作的问题是误差要么是常数[90]，[91]或者需要几个具有潜在冗余近似值的阶段[92].因此，马哈詹等人[93]介绍了 MITHRA，一个联合设计的用于神经逼近的软硬件控制框架。MITHRA 通过统计保证实现可配置的输出质量损失。最大似然分类器预测单个近似误差，允许与质量阈值进行比较。奥利维拉等人最近的工作[94]还探索了使用低开销分类树的近似。即使是基于软件的执行，他们也实现了

与 NPU 相当的应用程序加速[90]硬件实现。最后，ML 也被用来减轻现有近似加速器中故障的影响。塔赫尔等人[95]观察到，在许多输入测试向量中，故障倾向于以类似的方式出现。这种观察能够使用分类/回归模型进行有效的误差补偿，从而基于给定输入的预测故障校正输出。

4 当前实践分析

本节考察了现有工作中采用的各种技术。这些比较强调未来工作中潜在有用的设计实践和策略。

工作分为两类，代表设计约束和操作时间表的自然划分，因此对应于不同的设计实践。第一类，在线 ML 应用程序，包括在运行时直接应用 ML 技术的工作，即使培训是离线进行的。因此，这项工作的设计复杂性固有地受到功率、面积和实时处理开销等实际限制。第二类，离线 ML 应用，而是应用 ML 来指导架构实现，涉及设计和模拟等任务。因此，离线 ML 应用程序的模型可以利用更高的复杂性和更高的开销选项，代价是训练/预测时间。

4.1 在线语言应用

模型选择:在线 ML 应用程序主要使用决策树或人工神经网络(在监督学习模型的情况下)，以及 Q 学习或深度 Q 学习(在 RL 模型的情况下)。注意，这些学习方法的任务不一定是分离的，特别是对于控制。Fettes 等人[51]将 DVFS 同时视为监督学习回归任务和强化学习任务。监督学习方法预测缓冲器利用率或缓冲器利用率的变化，以确定合适的 DVFS 模式。相反，RL 方法直接使用 DVFS 模式作为动作空间。这两种模型都可以表现良好，但是 RL 模型更普遍适用，因为能量/吞吐量的权衡可以根据应用需求进行调整，并且不需要阈值调整。这当然不意味着 RL 是一个放之四海而皆准的解决方案。监督学习模型在函数逼近中有很强的应用 90]，[91]，[92]，[94]和分支预测任务[41]，[42]，这远不太适合(如果不是不可能的话)使用 RL，因为这些任务不能很好地表示为一系列动作。

实施和开销:在线 ML 应用程序的实施突出了数据可用性、模型存储空间等方面的限制。，表明需要一个高效且通常较低复杂性的模型。随着越来越多的研究走向现实世界的实施，这些限制可能会变得更加重要。

几部基于国家奥委会的作品[46]，[56]，[71]已经应用了不同的全局数据收集方法来支持 ML 模型。Daya 等人[56]使用单独的无缓冲饥饿网络实现了自学习注入节流，该网络携带饥饿标志，对于具有 N 个节点的网络，该标志被编码为一个热的 N 位向量。这些饥饿向量被传播到所有节点，允许基于单个节点的 Q 学习代理确定适当的注入节流。Soteriou 等人[58]类似地使用专用网络来收集缓冲区利用率和 VC 占用率统计数据。由 Won 等人提出的基于人工神经网络的 DVFS 控制 71]通过将数据编码为标准数据包报头中未使用的位，避免了额外的状态/数据网络。当数据包通过路由器时，中央控制单元会适时收集数据。这种方法引入了对数据过时的潜在担忧，但是以前的工作[96]对于足够大(50K 周期)的控制窗口，观察到与全知数据收集几乎相同的性能。可以通过发送专用数据包来适应更小的时间窗口，正如 Savva 等人所做的那样[46].

实现也可以考虑使用硬件或软件模型。使用专用硬件的实现通常会经历较低的执行时间开销，但还有其他考虑因素。Esmaeilzadeh 等人[90]使用专用硬件模块实现了用于函数逼近的神经处理器(NPU)。他们还考虑了软件实现，但是与基线 x86 函数相比，软件执行的指令数量增加得令人望而却步。奥利维拉等人的后期工作[94]发现使用简单分类树的函数逼近可以获得与 NPU 相似的结果[90]几个应用程序的应用程序加速和错误率(尽管平均来说稍差)。他们的纯软件实现突出了面积/功率和精度/性能之间的权衡。Won 等人[71]观察到类似的权衡，选择使用片上微控制器而不是专用硬件在软件中实现人工神经网络。这种实现比硬件实现多消耗几个数量级的周期(推断为 15K 个周期)，但平均功率需要减少 50mW。

硬件实现的方法也可能因任务而异。Savva 等人在工作中观察到了“标准”人工神经网络实现[46].他们结合了一个用于控制的有限状态机、一个用于计算的乘法累加单元阵列、一个用于加载和存储结果的寄存器阵列以及一个基于查找表的激活函数。媒体访问控制阵列宽度和计算精度都可以调整，以平衡功率/面积和精度/速度。相比之下，圣阿曼特等人[41]使用混合信号设计实现了 percep- tron 分支预测器。他们在模拟电路中实现了点积，利用电阻调整和电流求和来实现可行的开销。RL 模型的硬件也存在差异。“标准”Q 学习实现需要一个查找表来存储状态-动作值。伊佩克等人[33]以及穆昆丹和马丁内斯[34]改为使用 CMAC [97]，用多个粗粒度重叠表替换潜在的大量 Q 学习表。这种方法还包括散列，使用散列状态属性来索引 CMAC 表。综合起来，这两种方法平衡了泛化和开销，尽管根据任务的不同可能会引入碰撞/干扰。进一步将散列、CMAC 表查找和计算相结合，允许每个周期评估更多可能的动作值。

优化:在线训练的在线 ML 应用程序受益于对运行时工作负载特性的适应性。尽管有这些好处，低模型精度会对系统性能产生负面影响，最明显的是在执行开始时或在工作负载特性高度变化期间。可以考虑采用控制和学习来避免这些不利影响。一些基于 RL 的工作[25]考虑通过引入“影子”操作来减轻勘探期间不良行动的影响。这些操作是低可信度的操作

由模型建议，但仍在模型更新中使用，但未被系统执行。因此，该模型在不负面影响系统的情况下获得了关于行为良好性的反馈。在基于监督学习的控制任务中 71]使用 PI 控制器做出的控制动作在线训练人工神经网络，这表现出少得多的启动延迟。在训练之后，使用基于误差和一致性的混合组合做出控制决策，允许互补控制。在最简单的情况下，检查默认配置的性能，如[38】，提供了一个保证，ML 模型的性能不会比默认的差，但可以表现得更好。

在大多数作品中，ML 模型取代了现有的方法(通常是启发式的)。然而，最近的几部作品[28]，[45]通过将传统(非最大似然)和最大似然方法结合起来，已经显示出显著的优势。这些改进源自两种方法的正交预测/决策能力，从而实现了协同性能改进。该方法还可以通过关注传统方法中的特定缺点来实现低成本的 ML 应用。两部近期作品[28]，[45]仅考虑分支预测，因此存在探索这种潜在协同设计范式的重大机会。

4.2 离线多媒体应用程序

模型/特征选择:离线 ML 应用程序通常表现出大量的模型/特征多样性，因为模型本身并不依赖于特定的体系结构。因此，模型和特征选择更侧重于最大化模型精度，同时最小化整体学习/预测时间。特别地，设计空间探索可以使用直接优化的迭代搜索方法或者基于设计的预测最优性选择最佳点的监督学习方法来进行。几部作品 60]，[61]，[62]使用了迭代的 STAGE [98]一种算法，通过学习评估函数，从给定的起点预测本地搜索结果，从而优化 3D NoC 链接的本地搜索。最近的工作转而应用了深度强化学习[63]到无路由器的 NoC 设计。提议的蒙特卡罗树搜索，连同卷积神经网络建议的动作，提供了高效的搜索过程。随着计算资源的增加，并行线程也被用来扩展设计空间探索。系统级设计空间探索更倾向于标准的可视化学习方法[17]，[64]，[67].具体型号选择因线性而异[17]，[64]和非线性[67]回归模型，以及随机森林和神经网络
[64]寻找实现。如在线 ML 应用程序，在第节中讨论 4.1，有些任务自然局限于监督学习方法。跨架构预测就是一个范例[12]，[13]，[15]，[19]，[20].

优化:ML 模型在离线 ML 应用程序中的有用性很大程度上取决于相对于传统设计方法的开销。因此，优化主要集中在提高数据效率和整体模型精度上。

集成方法已经在在线 ML 应用中被提出 38]，但主要在离线 ML 应用中找到应用，因为集合可以任意大(与可用的计算资源相关)。为了提高效率，已经提出了几种优化方法。Jooya 等人
[17]使用稍微不同的配置训练了许多神经网络，并使用这些模型具有很好的通用性，对输入噪声最不敏感。他们进一步引入了异常值检测，通过过滤其性能和/或功率预测与训练数据中最接近的配置有很大差异的预测。Ardalani 等人[20]相反，保留他们训练的所有 100 个模型，注意模型在一个应用中可能是非常强的预测器，但在另一个应用中是弱的预测器。他们通过只选择 60 个最接近预测中值的个体来弥补这个困境。

抽样方法优化虽然不是体系结构任务所独有的，但在提高模型精度时仍然需要考虑。Sangaiah 等人[67]在他们的非核心绩效预测模型中考虑了潜在的系统偏差。具体来说，他们观察到均匀随机采样可能无法充分捕获非均匀配置空间中的性能关系(如使用 2 的幂进行大小调整的缓存配置)。因此，他们使用了一种低差异采样技术，SOBOL [99]，以消除这种系统偏差，并防止低端配置的性能过度预测。

4.3 领域知识和模型解释

最大似然算法提供的强大关系学习能力支持许多任务中的黑盒实现(即，不考虑任务特定的特性)，但可能无法利用可以提高可解释性或整体模型性能的额外领域知识。此外，在一些应用中，do- main 知识可以帮助识别异常行为，并再次提高整体模型的有用性。这些主题在几个具体的作品中得到了强调，但对于应用于建筑的 ML 来说可以普遍适用。

一种方法使用机械经验模型，将基于领域知识的机械框架与基于经验最大似然学习的特定参数相结合。与纯机械模型相比，这些模型简化了实现[11]，可以避免在纯机械模型中作出不正确的假设[66]，并且通过避免过度拟合，可以提供比纯经验模型更高的精度[11].艾克曼等人[11]还演示了如何使用这些模型构建 CPI 堆栈，从而允许有意义的替代设计比较。

邓等[38]在他们预测最优 NVM 写策略的工作中，提出了一个基于任务特定领域知识调整 ML 模型的案例。在初步分析之后，他们发现了单个配置参数(磨损定额)如何导致更高的复杂性和对 IPC 和系统能量的次优预测精度，即使使用二次回归和梯度增强模型。从配置空间中排除磨损定额，然后将其应用于预测的最佳配置，可使预测精度提高 2-6%。Ardalani 等人[20]类似地检查了他们跨平台性能预测学习模型中的固有缺陷。有些预测对学习模型来说很容易，对人类来说很难，代表了 ML 应用的理想场景；反之亦然。在这两种情况下，通过考虑任务特征来加强 ML 应用。

5 未来的工作

这一节综合了第二节的观察和分析 3 和科 4 发现机会并详细说明未来工作的需求。这些机会可能会到来

在模型级别，利用改进的实现策略和学习能力，或者在应用程序级别，解决对通用工具的需求或者探索全新的领域。

5.1 研究模型和算法

现有的作品通常在单个时间尺度或抽象层次上应用 ML。这些限制激发了对模型和算法的研究，这些模型和算法从系统设计和执行特性两个方面捕捉了体系结构的层次特性。

执行阶段级预测:使用基本块进行应用分析[100]长期以来一直是一种有用的模拟方法，通过识别程序执行中的独特和代表性阶段而成为可能。相位级预测为应用于建筑的最大似然法提供了类似的好处。特别是最近的一些工作已经证明了有希望的结果，对于两种性能预测都有很高的准确性[13]以及能量和可靠性(寿命)[38].一般来说，大多数工作[2]，[17]，[67]尚未采用阶段级预测技术(或未明确提及其方法)。具体来说，未来的工作可以探索基于相级行为的控制和系统重构的预测，而不是静态窗口[85]或应用程序级行为[75]，[101].

开发纳秒级:在许多 DVFS 控制方案中使用的粗粒度最大似然比基于标准控制理论的方案提供了显著的优势，然而细粒度控制可以提供更高的效率。具体而言，白等人的分析[76]表示能耗变化非常快，对于某些应用来说大约为 1K 指令。利用这些短暂的时间间隔需要仔细考虑模型和算法。未来的工作可能会优化现有的算法，如经验分享[102]和混合/串联控制[71]，或者考虑更适合新模型(例如，分层模型)的方法。这些方法还可以带来额外的纳秒级协同优化机会，如动态有限责任公司划分，以进一步提高效率。

应用分层和多代理模型:计算机系统中的应用程序执行自然遵循分层结构，在顶层，任务被分配给内核，然后内核被分配动态功率和资源预算(例如，有限责任公司空间)，最后，在底层，数据/控制包在内核和内存之间发送。因此，单个机器学习模型可能难以学习合适的设计/控制策略。此外，在强化学习模型的情况下，基于它们对整体执行时间、能量效率等的影响，精确地将信用分配给特定的低级动作可能是极其困难的。最近工作中一个很有前途的方法是层次模型103].Hier 架构的强化学习模型支持目标导向的学习，这在反馈稀疏的环境中(例如，任务分配)尤其有益。因此，将分层学习应用于架构可以实现更有效的多层次设计和控制。多智能体模型是机器学习研究的另一个有前途的领域。这些模型倾向于关注强化学习代理对其环境只有部分可观察性的问题。尽管部分可观测性可能不是单个计算机系统的主要关注点，但最近的工作[104]已经将这一概念应用于互联网分组路由器-

ing，并通过改善个体代理之间的合作展示了融合优势。

5.2 加强实施战略

越来越复杂的模型需要有效的策略和技术来减少开销和实现。如下所述，模型修剪和权重量化是两种特别有效的技术，在加速器中已被证实具有优势，同时许多其他有前途的方法也正在被探索[105].

探索模型修剪:模型复杂性可能是在线 ML 应用程序的一个限制因素。标准的 Q 学习方法需要一个潜在的大表来存储动作值。基于神经网络的学习方法(在深度 Q 网络中)和监督学习都需要网络权重存储和额外的处理能力。特别是神经网络，因此在现有的工作中通常被限制在几层，许多只使用一个隐藏层[46]，[71]，[85]，[93]有些使用一两个隐藏层[90]，[91].

最近对神经网络的研究已经证明了通过剪枝降低模型复杂性的有前途的方法[106]，[107].一般的直觉是，许多连接是不必要的，因此可以被修剪。迭代地修剪一个高复杂度的网络，然后在稀疏架构上从头开始重新训练可以获得良好的结果，一些工作证明了非常高的稀疏性(> 90%)和很少的精度损失[107].

应用于神经网络的剪枝，无论是深度 Q 学习还是监督学习回归/分类，都提供了一种训练复杂模型以获得高精度，然后剪枝以实现可行的方法。深度 Q 学习应用程序到目前为止仅限于两个作品[51]，[59]，其中一项目前无法实施[59].未来的工作可能会考虑修剪深度 Q 网络，作为标准 Q 学习方法的有用替代。修剪还为性能预测(如 DVFS 控制)和函数逼近(如支持 ML 的近似计算)的未来工作提供了大量机会。系统级近似(在第节中讨论 5.4)可能特别受益于修剪高复杂性模型。

探索量化:现有的工作主要将量化应用于 Q 学习中的状态值，以实现实用的 Q 表实现。类似地，通过降低乘法累加器的精度，神经网络受益于执行时间、功率和面积的潜在减少。然而，最近的工作为基于降低精度模型的替代硬件实现提供了一系列新的机会。

例如，二进制神经网络将权重量化为+1 或-1，从而能够基于逐位运算而不是算术运算进行计算[108].另一种方法考虑将神经网络权重量化为有限(但非二进制)子集，以便用查找表访问取代乘法运算[109]，允许更高的精度和更低的执行时间，尽管可能会增加面积成本。ML 应用的未来工作可以利用类似的硬件实现，同时探索各种任务和控制方案的最佳量化级别。

5.3 开发通用工具

现有机器学习工具(例如 scikit-learn [110])已被证明对相对简单的 ML 应用程序有用。从来没有-

此外，复杂的设计和模拟任务需要更复杂的工具，以便使用通用框架实现快速的特定任务优化。

实现广泛的应用和优化:与启发式设计策略类似，专门构建的架构工具在实现满足通用用例的设计、探索和模拟方面非常有用。这些方法的应用可能仍然局限于特定的问题、优化标准、系统配置等。鉴于架构研究(和机器学习研究)的快节奏性质，需要开发更通用的工具和易于修改的框架来解决更广泛的应用和优化选项。

基于 ML 的设计工具特别有前途，最近的作品展示了在室内设计空间的成功应用(例如，超过 1012 英寸[63]).Oppor-然而，新设计工具的特性不限于
特定的架构组件。芯片布局是一个值得注意的例子，其中即使是简单的聚类算法也可以显著优于现有的启发式方法[88].未来的工作还可以继续开发更广泛适用的性能和功率预测工具。特别是最近关于跨平台性能预测的工作[21]表明纯静态特征具有高预测精度的可能性，因此代表了额外研究的另一个潜在领域。

实现广泛使用:通用工具通过促进快速设计和评估带来额外的好处。使用机器学习方法，可以简单地修改训练数据(在监督学习环境中)或动作/奖励表示(在强化学习环境中)，而不是探索模型、数据表示策略、搜索方法等。，可能没有先验的机器学习经验。例如，最近的工作
[63]设想将深度强化学习框架重新用于涉及插入物、小芯片和加速器的各种 NoC 相关设计任务。虽然该框架可能无法兼容所有工作，尤其是在新领域，但它可能为机器学习应用于架构提供更好的基础，尤其是对于机器学习背景有限的个人。

5.4 拥抱新应用

未来的工作有很多机会将 ML 应用于现有和新兴的体系结构，取代启发式方法来实现长期扩展，并提高自动化设计的能力。

探索新兴技术:若干建议[30]，[37]，[38]，[39]确定如何使用 ML 来优化标准(能量、性能)和非标准(寿命、尾部延迟)标准。这些非标准标准在新兴技术中尤其成问题，因为如果没有一些可靠性保证，这些技术很难得到广泛应用。因此，应用最大似然法来优化标准和非标准标准，为将来的工作提供了一种动态智能平衡控制策略的方法，而不是依赖启发式方法。

探索新兴架构:ML 应用于新兴架构通过支持快速开发提供了类似的好处，即使是有限的最佳实践知识，这可能需要时间来开发。长期存在的设计领域的工作，如任务分配和分支预测，可能会结合最佳实践领域知识来指导方法，无论是应用 ML 还是
一些其他的传统方法。新兴架构的最佳实践可能不会立即显现。例如，最大似然法在 2D 光子国家芯片上的应用[48]，2.5D 内存处理设计[24]和 3D 国家和地区奥委会[60]，[61]，[62]都显示出优于现有方法的强大性能。未来的工作可以探索 ML 应用于新的关注点，例如插入器和特定领域加速器中的连接性和可重构性。

展开系统级近似计算:如第节所述 3.6ML 在近似计算中的应用大多局限于函数逼近。然而，近似计算的许多其他方面已经在非最大似然工作中实现，这可以通过利用最大似然获得额外的好处。例如，约-NoC [111]使用近似和编码数据减少网络流量。另一项工作探索了智能相机系统的多面近似方案[112]使用近似动态随机存取存储器(较低的重新刷新速率)、近似算法(循环跳过)和近似数据(较低的传感器分辨率)。现有的基于编译器的工作[113]对于系统范围的近似，增强了确定可近似代码的先验能力，但依赖于具有代表性输入的启发式搜索。因此，这种方法不能提供统计上的保证，例如密特拉[93].未来的工作可能探索基于深度强化学习(或者可能是分层强化学习)的搜索，以将现有的近似技术结合到用于高维近似和协同优化的可扩展框架中。

实现系统级、组件级优化:最近的工作已经开始探索更广泛的基于 ML 的设计和优化策略。MLNoC [64]为 NoC 设计优化探索了广阔的 SoC 特性空间。核心和非核心 DVFS 在机器学习机器中结合在一起[86]，以及 LLC 动态缓存分区，以探索运行时的协同优化潜力。相关 DNN 加速器研究[114]提出了基于硬件(例如位宽)和神经网络参数(例如 L2 正则化)的协同优化。这些工作激发了对系统级、组件级 ML 应用程序的思考。

现有系统级优化方案(例如[80]，
[83]，[101】)仅在单个且非常高的抽象级别(例如，任务分配或大)考虑配置机会。小核心配置)。虽然这些工作可能包括低级功能，如在其最大似然模型中的片上网络利用率和动态随机存取存储器带宽，但它们没有考虑组件级优化技术的影响，如片上网络数据包路由、缓存预取等。相反，我们设想了一个基于 ML 的系统级、组件级运行时优化框架。在这个框架中，控制决策将涉及更大层次的组件级(或更低)特性和控制选项以及更高级别的决策，从而为运行时优化提供更全面和精确的视角。

高级自动化设计:虽然全自动化设计可能是最终目标，但日益自动化的设计仍然是未来工作的重要里程碑。具体来说，随着越来越多的任务被自动化，在机器学习和架构之间实现正反馈循环的潜力越来越大，为这两个领域提供了更多的实际好处。当然，有一些必须解决的中间挑战，每一个挑战都是未来工作的一个重要领域。一个挑战涉及到建筑构件的层次结构建模。这个模型可能会受益于跨

系统堆栈，从过程技术到全系统行为，从而为现实世界的系统生成高度精确的表示。另一个研究方向可以探索机器学习模型的方法，以确定潜在的设计改进方面。理想情况下，该模型不仅可以探索现有选项的重新配置(如[115])，还能生成新颖的配置选项。整合这些和潜在的其他能力可以提供一个框架来推进自动化设计。

6 结论

机器学习已经迅速成为架构中的一个强大工具，对设计、优化、仿真等都有既定的适用性。值得注意的是，ML 已经成功应用于许多组件，包括内核、缓存、片上网络和内存，其性能经常超过现有的最先进的分析、启发式和人工专家策略。多样化的训练方法和学习模式进一步促进了广泛的应用，允许基于任务需求在性能和开销之间进行有效的权衡。这些进步很可能只是建筑革命性转变的开始。

涉及修剪和量化的模型级优化机会通过提供更实际的实现提供了广泛的好处。类似地，使用越来越强大的 ML 模型扩展现有工作的机会比比皆是，从而实现更细粒度的系统级实现。最后，ML 可以应用于体系结构的全新方面，学习分层或抽象表示，以基于高层和低层细节来表征整个系统行为。这些广泛的机会，以及尚未预见的可能性，可能最终会结束高度(甚至完全)自动化的架构设计。

参考

[1] 南 Kotsiantis，“监督机器学习:分类技术的回顾”，2007 年计算机工程中新兴人工智能应用会议论文集:真实世界的人工智能系统及其在电子健康、人机交互、信息检索和普及技术中的应用，第 3-24 页，2007 年。

[2] 名词（noun 的缩写）张，拉弗蒂，霍夫曼，“一种基于图形模型的性能约束下能量最小化方法”，国际编程语言和操作系统体系结构支持会议

系统，2015 年 3 月。

[3] 动词 （verb 的缩写）统计学习理论概述，IEEE 神经网络学报，第 10 卷，1999 年 9 月。

[4] J.Shlens，“主成分分析教程”，2014 年。arXiv:1404.1100。

[5] 米（meter 的缩写））王，李，“基于贝叶斯协同学习的集成电路高效分层性能建模”，载于《设计自动化会议》，2017 年 6 月。

[6] R.萨顿和巴尔托，《强化学习:导论》。美国剑桥:麻省理工学院出版社，第二版。，1998.

[7] I. Guyon 和 A. Elisseeff，“变量和特征选择导论”，《机器学习研究杂志》，第 3 卷，第 1157-1182 页，2003 年 3 月。

[8] J.李、陈光楷、王树森、莫斯塔特、、唐俊杰

H.刘，“特征选择:数据视角”，ACM 计算调查，第 50 卷，2018 年 1 月。

[9] E.伊佩克，S. A .麦基，B. R .德苏平斯基，m .舒尔茨和 r .卡鲁-阿纳，“通过预测建模有效地探索建筑设计空间”，在国际编程语言和操作系统建筑支持会议上，2006 年 10 月。

[10] B.“预测计算机系统设计方案性能的机器学习模型”，国际并行处理会议(ICPP)，2008 年 9 月。

[11] 南艾克曼、霍斯特和艾克霍特，“用于在真实硬件上构建 cpi 堆栈的机械经验处理器性能建模”，在系统和软件性能分析国际研讨会上，2011 年 4 月。

[12] X.郑，拉维库马尔，约翰和格斯陶尔，“基于学习的分析性跨平台性能预测”，国际嵌入式计算机系统会议:体系结构，建模和仿真(SAMOS)，2015 年 7 月。

[13] X.郑，约翰 L. K .和葛斯特劳，“精确的相位级跨平台功率和性能估算”，载于设计自动化会议(DAC)，2016 年 6 月。

[14] 名词（noun 的缩写）贾恩·阿加瓦尔和扎赫兰，“多线程应用的性能预测”，在 2019 年 6 月与 ISCA 联合举办的建筑人工智能辅助设计国际研讨会上。

[15] W.贾，肖传国，马多诺西，“基于自动回归的 gpu 设计空间探索”，系统与软件性能分析国际研讨会(IS- PASS)，2012 年 4 月。

[16] G.吴，李雅舍夫斯基，贾亚泽纳

D.池鸥，“使用机器学习的 gpu 性能和功耗估计”，在高性能计算机架构国际研讨会(HPCA)上，2015 年 2 月。

[17] A.“多目标 gpu 设计空间探索优化”，国际高性能计算与仿真会议(HPCS)，2016 年 7 月。

[18] T.-林瑞麟、李彦宏、佩德拉姆和陈立荣，“深度学习吞吐量处理器中内存控制器布局的设计空间探索”，载于《IEEE 计算机体系结构通讯》，第 18 卷，2019 年 3 月。

[19] I .巴尔迪尼、S. J .芬克和 e .奥特曼，“使用机器学习从 cpu 运行中预测 gpu 性能”，载于《计算机体系结构和高性能计算国际研讨会》(SBAC-PAD)，2014 年 10 月。

[20] 名词（noun 的缩写） Ardalani， C. Lestourgeon， K. Sankaralingam 和 X. Zhu，“使用 cpu 代码预测 gpu 性能的跨架构性能预测(xapp)”，在国际微体系结构研讨会(MICRO)上，2015 年 6 月。

[21] 名词（noun 的缩写）阿尔达拉尼，U. Thakker，A. Albarghouthi 和 K. Sankaralingam，“基于静态分析的跨架构性能预测，使用机器学习”，在 2019 年 6 月与 ISCA 联合举办的建筑人工智能辅助设计国际研讨会上。

[22] K.奥尼尔，P. Brisk，E. Shriver 和 M. Kishinevsky，“Hal-wpe:GPU 的硬件辅助轻量级性能评估”，载于设计自动化会议(DAC)，2017 年 6 月。

[23] Y.李，彭尼，拉马穆尔蒂，陈，“通用 GPU 的片上流量模式表征:深度学习方法”，载于国际计算机设计会议()，2019 年 11 月。

[24] A.Pattnaik，X. Tang，A. Jog，O. Kayran，A. K. Mishra，M. T. Kandemir，O. Mutlu 和 C. R. Das，“具有内存处理能力的 gpu 架构的调度技术”，载于并行架构和编译技术国际会议(PACT)，2016 年 9 月。

[25] 长度佩莱德，曼诺，美国魏泽，和伊特西，“使用强化学习的语义局部性和基于上下文的预取”，在

高性能计算机架构国际研讨会(HPCA)，2015 年 6 月。

[26] Y.曾和郭，“基于长短期存储器的硬件预取器”，在存储器系统国际研讨会(Mem- Sys)上，2017 年 10 月。

[27] 页（page 的缩写）布劳恩和李兹，“理解预取的内存访问模式”，在与 ISCA 联合举办的建筑人工智能辅助设计国际研讨会上，2019 年 6 月。

[28] E.巴蒂亚、g .沙孔、s .普利斯利、e .特兰、P. V .格拉茨和 D. A .姬广亮·内兹，“基于感知机的预取过滤”，在计算机体系结构国际研讨会(ISCA)上，2019 年 6 月。

[29] E.王，和耐兹，“面向重用预测的感知器学习”，载于微体系结构国际研讨会，2016 年 10 月。

[30] H.王，易晓明，黄培平，程，周 k，“高效固态硬盘

通过使用机器学习避免不必要的写入来缓存

ing，“并行处理国际会议(ICPP)，2018 年 8 月。

[31] A.Margaritov，d .乌斯蒂古夫，E. Bugnion，和 B. Grot，“通过学习页表索引的虚拟地址翻译”，在神经信息处理系统会议(NeurIPS)，2018 年 12 月。

[32] T.菲利普·克拉斯卡，比特尔，迟，迪恩和多佐蒂斯，“学习型索引结构的案例”，国际数据管理会议，2018 年 6 月。

[33] E.伊佩克，o．穆特鲁，J. F．马丁内兹和r．卡鲁阿纳，“自寻优记忆控制器:强化学习方法”，在高性能计算机体系结构国际研讨会(HPCA)上，2008年6月。

[34] J.穆昆丹和马丁内斯，“莫尔斯:多目标可重新配置的自寻优内存调度程序”，在高性能计算机架构国际研讨会(HPCA)，2012年2月。

[35] 南俞海平，黄海平，徐德伟，“一种基于q学习的2.5d集成多核微处理器和存储器的自适应i/o通信”，IEEE计算机事务，第65卷，2015年6月。

[36] 南王，，“通过在线数据聚类和编码减少数据移动能量”，载于国际微体系结构研讨会，2016年10月。

[37] W.Kang和S. Yoo，“动态管理用于强化学习辅助垃圾收集的关键状态，以减少ssd中的长尾延迟”，在设计自动化会议(DAC)上，2018年6月。

[38] Z.邓，张，米沙拉，霍夫曼，钟，记忆鸡尾酒疗法:一种优化非易失性存储器动态权衡的基于学习的通用框架，载于国际微体系结构研讨会，2017年10月。

[39] J.肖，熊，吴，易，金，胡k，“基于在线学习的数据中心磁盘故障预测”，国际并行处理会议(ICPP)，2018年6月。

[40] D.和林，，“基于概念的动态分支预测”，高性能计算机体系结构国际研讨会(HPCA)，2001年1月。

[41] R.“低功耗、高性能的模拟神经分支预测”，载于微体系结构国际研讨会，2008年11月。

[42] D.姬广亮•内兹，“一种优化的缩放神经分支预测器”，载于*国际计算机设计会议(ICCD)，2011年10月。*

[43] E.“间接分支的比特级感知器预测”，国际计算机体系结构研讨会(ISCA)，2019年6月。

[44] A.在2016年与ISCA联合举办的第五届JILP计算机体系结构工作车间竞赛:冠军分支预测中，Seznec，“再次出现分支预测”。

[45] 南Tarsa，c-k．Lin，G. Keskin，G. Chinya和H. Wang，“通过用卷积神经网络建模全局历史来改进分支预测”，在与联合举办的建筑人工智能辅助设计国际研讨会上，2019年6月。

[46] A.G. Savva，T. Theocharides和V. Soteriou，“片上网络的智能开/关动态链路管理”，载于《电气与计算机工程杂志-片上网络专刊:架构、设计方法和案例研究》，2012年1月。

[47] D.DiTomaso，A. Sikder，A. Kodi和A. Louri，“机器学习使能的功率感知片上网络设计”，载于《欧洲设计、自动化和测试》(DATE)，2017年3月。

[48] 南文克尔、柯迪、布内斯库和卢瑞，“用机器学习扩展异构多内核光子互连的功率效率和性能”，在高性能计算机体系结构国际研讨会(HPCA)上，2018年2月。

[49] 米（meter的缩写））雷扎、勒、德、巴尤米和赵，“神经网络:在黑硅时代使用神经网络的异构众核网络中的能量优化”，国际电路与系统研讨会(ISCAS)，2018年5月。

[50] 米（meter的缩写））Clark，A. Kodi，R. Bunescu和A. Louri，“Lead:NOC中学习使能的能量感知动态电压/频率缩放”，载于设计自动化会议(DAC)，2018年6月。

[51] 费特茨，m．克拉克，r．布内斯库，a．卡兰思和a．卢里，“使用监督和强化学习技术的NOC中的动态电压和频率缩放”，IEEE计算机事务，第68卷，2019年3月。

[52] J.伯颜和利特曼，“动态变化网络中的分组路由:强化学习方法”，《神经信息处理系统进展》，第6卷，第671-678页，1994年。

[53] 米（meter的缩写））“动态变化的片上网络中的分组路由”，国际并行和分布式处理研讨会，2005年4月。

[54] C.2010年12月，在与MICRO合作举办的“片上网络架构国际研讨会”上，冯，陆振中，杨春奇，李俊杰，张敏，“一种基于片上网络强化学习的可重构容错偏转路由算法”。

[55] 米（meter的缩写））易卜拉希米，m．达内什塔拉布和f．法拉纳基安，“Haraq:高度自适应路由的拥塞感知学习模型片上网络中的算法”，在国际片上网络(NOCS)研讨会上，2012年6月。

[56] B.K. Daya，l-s．Peh和A. P. Chandrakasan，“寻求具有单周期快速路径和自学习节流的高性能无缓冲NOC”，载于设计自动化会议(DAC)，2016年6月。

[57] B.王，陆泽涛，陈素珊，“基于人工神经网络的片上网络接纳控制”，载于设计自动化会议，2019年6月。

[58] 动词（verb的缩写）Soteriou，T. Theocharides和E. Kakoulli，“在基于网络芯片的多计算机中实现智能热点预防的整体方法”，IEEE计算机事务，第65卷，2015年5月。

[59] J.尹，，车，奥斯卡金和陆，2018年6月，在与联合举办的“建筑人工智能辅助设计国际研讨会”上，“迈向更高效的noc仲裁:深度强化学习方法”。

[60] 南“优化3d noc设计以提高能效:一种机器学习方法”，载于国际计算机辅助设计会议(ICCAD)，2015年11月。

[61] 南达斯、多帕、潘德和查克拉巴蒂，“高能效和可靠的3d片上网络:架构和优化算法”，在国际计算机辅助设计会议(ICCAD)上，2016年11月。

[62] B.乔达尔，金，多帕，潘德，马库列斯库和马库列斯库，“异构多核系统的基于学习的应用不可知3d noc设计”，IEEE计算机事务，第68卷，2019年6月。

[63] T.-林瑞麟、彭尼、佩德拉姆和陈，“优化无路由器片上网络设计:基于学习的创新框架”，2019年5月。arXiv:1905.04423。

[64] 名词（noun的缩写）拉玛钱德朗和沙阿，“基于机器学习的片上网络设计方法”，国际计算机体系结构和高性能计算研讨会(SBAC-PAD)，2018年9月。

[65] 南基于模拟退火的多目标优化算法:阿莫萨，《进化计算的IEEE事务》，第12卷，2008年5月。

[66] Z.钱，，徐春义，马库列斯库，“基于学习支持向量回归模型的片上网络性能分析工具”，载于《欧洲设计、自动化与测试》(DATE)，2013年3月。

[67] K.桑盖亚，m．亨普斯特德和b．塔金，“非核心rpd:通过回归建模的非核心快速设计空间探索”，在国际计算机辅助设计会议(ICCAD)，2015年11月。

[68] D.DiTomaso，T. Boraten，A. Kodi和A. Louri，“使用智能预测技术在片上网络中动态减少错误”，载于国际微体系结构研讨会(MICRO)，2016年10月。

[69] K.王，A. Louri，A. Karanth，R. Bunescu，“使用强化学习的高性能、高能效、容错片上网络设计”，载于《欧洲设计、自动化与测试》(DATE)，2019年3月。

[70] K.王，A. Louri，A. Karanth，R. Bunescu，“Intellinoc:面向多内核的高能效、可靠片上通信的整体设计框架”，2019年6月，计算机体系结构国际研讨会()。

[71] J.-Y. Won，X. Chen，P. Gratz，J. Hu和V. Soteriou，“自启动:用于cmp非核心电源管理的人工神经网络在线学习”，在高性能计算机体系结构国际研讨会(HPCA)上，2014年2月。

[72] G.-潘永年、朱永年和赖百川，“使用多处理器多级强化学习的可扩展电源管理”，载于《电子系统设计自动化美国计算机学会会刊》，2014年8月。

[73] 页（page的缩写）E. Bailey，D. K. Lowenthal，V. Ravi，B. Rountree，M. Schulz，和
B.苏平斯基，“功率受限异构系统的自适应配置选择”，国际并行处理会议(ICPP)，2014年9月。

[74] D.罗传国、宋传国和苏国荣，“预测导向的性能——交互应用的能量权衡”，载于微体系结构国际研讨会，2015年12月。

[75] 名词（noun的缩写）米什拉，J. D．拉弗蒂和h．霍夫曼，“卡乐里:可预测延迟和低能量的学习控制”，在编程语言和操作系统架构支持国际会议(ASPLOS)上，2018年3月。

[76] Y.白，李伟伟，，“电压调节器效率感知电源管理”，在编程语言和操作系统架构支持国际会议(ASP- LOS)，2017年4月。

[77] 米（meter的缩写））艾伦和P. Fritzsche，“xpilot游戏ai的自适应kanerva编码强化学习”，在IEEE进化计算大会上，2011年6月。

[78] Z.陈和D. Marculescu，“用于功率受限多核心系统性能优化的分布式强化学习”，载于《欧洲设计、自动化和测试》(DATE)，2015年3月。

[79] Z.陈，D. Stamoulis和D. Marculescu，“利润:多核心系统的优先级和功率/性能优化”，《IEEE集成电路和系统计算机辅助设计学报》，第37卷，第2064-2075页，2018年10月。

[80] C.“使用机器学习分类器的节能应用资源调度”，国际并行处理会议(ICPP)，2018年8月。

[81] 南塔尔萨、乔杜里、塞伯特、经亚、高尔、K.桑卡拉那拉亚南、林春凯、沙佩尔、辛哈尔和H.王，“使用机器学习实现后硅cpu适配”，计算机体系结构国际研讨会()，2019年6月。

[82] 南卢，特斯耶和w .伯利森，“热感知多核心任务分配的强化学习”，载于第25届大湖超大规模集成电路研讨会论文集，2015年5月。

[83] D.“异构CPU性能预测和调度的机器学习方法”，计算机体系结构和高性能计算国际研讨会(SBAC-PAD)，2017年10月。

[84] H.张，唐碧波，耿，马，“混合cpu-gpu集群中大规模视频工作负载的学习驱动并行化”，国际并行处理会议(ICPP)，2018年8月。

[85] R.伊佩克和马丁内兹，“芯片多处理器中多种交互资源的协调管理:机器学习方法”，载于国际微体系结构研讨会，2008年11月。

[86] R.贾恩、潘达和苏布拉莫尼，“机器学习机器:高速缓存、内核和片上网络的自适应协同优化”，载于《欧洲的设计、自动化和测试》(DATE)，2016年3月。

[87] Y.丁，n .米什拉和h .霍夫曼，“计算机系统优化的生成性和多阶段学习”，国际计算机体系结构研讨会(ISCA)，2019年6月。

[88] G.吴永旭，吴德德，拉古帕提，莫永延，朱春华，“加权k-均值算法的触发器聚类”，载于设计自动化会议(DAC)，2016年6月。

[89] 米（meter的缩写））Ozsoy，K. N. Khasawneh，C. Donovick，I. Gorelik，N. Abu- Ghazaleh和D. Ponomarev，“使用低级架构特征的基于硬件的恶意软件检测”，IEEE计算机事务，第65卷，2016年3月。

[90] H.埃斯马伊尔扎德、桑普森、塞泽和伯格，“通用近似程序的神经加速”，载于国际微体系结构研讨会，2012年12月。

[91] A.亚兹丹•巴克什、朴槿惠、夏尔马、洛特菲•卡姆兰和H.Esmaeilzadeh，“gpu吞吐量处理器的神经加速”，在微架构国际研讨会(MICRO)上，2015年12月。

[92] B.“大脑互动体:将可靠的精确度带入神经实现的近似计算”，在高性能计算机架构国际研讨会(HPCA)上，2015年2月。

[93] D.《在控制近似加速的质量权衡中实现统计保证》，载于国际计算机体系结构研讨会(ISCA)，2016年6月。

[94] G.奥利维拉、冈萨尔维斯、布兰达罗、贝克和卡罗，“将基于分类的算法用于通用近似计算”，载于设计自动化会议，2018年6月。

[95] F.n .塔赫尔、J. Callenes-Sloan和B. C .斯查费，“近似硬件加速器的基于机器学习的硬故障恢复模型”，载于设计自动化会议(DAC)，2018年6月。

[96] X.陈，许志永，金，格拉茨，胡，基希诺夫斯基，和单位Ogras，“cmp片上网络和末级缓存dvfs的网络内监控和控制策略”，载于片上网络国际研讨会，2012年5月。

[97] R.萨顿，“强化学习中的泛化:使用稀疏粗编码的成功例子”，神经信息处理系统会议，1996年6月。

[98] J.伯颜和摩尔，“通过局部搜索改进优化的学习评估函数”，《机器学习研究杂志》，2001年9月。

[99] 页（page的缩写）“算法659:实现sobol的准随机序列发生器”，数学软件上的ACM事务，第14卷，1988年3月。

[100] T.舍伍德、佩雷尔曼和卡尔德龙，“在应用中寻找周期性行为和模拟点的基本块分布分析”，载于并行体系结构和编译技术国际会议，2001年9月。

[101] W.王，戴维森，索法，“预测大规模numa机器上多线程应用的内存带宽和最优内核分配”，载于高性能计算机体系结构国际研讨会()，2016年3月。

[102] R.多智能体系统的强化学习算法，在智能体和群体编程研讨会上，2003。

[103] T.D. Kulkarni，K. R. Narasimhan，A. Saeedi和J. B. Tenenbaum，“分层深度强化学习:整合时间抽象和内在动机”，在神经信息处理系统会议(NeurIPS)上，2016年12月。

[104] H.毛，龚振中，张振中，肖振中，倪永中，“有限带宽限制下的互联网分组路由多代理通信学习”，2019年2月。arXiv:1903.05561。

[105] 动词（verb的缩写）陈永海，杨天杰，埃默，“深度神经网络的有效处理:教程和调查”，2017年8月。arXiv:1703.09039。

[106] 南韩，池，陈，戴利，“学习有效神经网络的权值和连接”，2015年10月。arXiv:1506.02626。

[107] D.C. Mocanu，E. Mocanu，P. Stone，P. H. Nguyen，M. Gibescu和A. Liotta，“受网络科学启发的具有自适应稀疏连通性的人工神经网络的可扩展训练”，《自然通讯》，第9卷，2018年6月。

[108] 米（meter的缩写））库尔巴里奥，胡巴拉，索德里，亚尼夫和本吉奥，“二值化神经网络:训练权重和激活限制在+1或-1的深度神经网络”，2016年3月。arXiv:1602.02830。

[109] 米（meter的缩写））拉兹利吉、伊马尼、库尚法尔和罗辛，“Looknn:无乘法的神经网络”，载于《欧洲的设计、自动化和测试》(DATE)，2017年3月。

[110] F.Pedregosa，G. Varoquaux，A. Gramfort，V. Michel，B. Thirion，O.Grisel，M. Blondel，P. Prettenhofer，R. Weiss，V. Dubourg，J.范德普拉斯、帕索斯、库纳波、布鲁赫、佩罗特和杜切斯内，“Scikit-learn:Python中的机器学习”，《机器学习研究杂志》，第12卷，第2825－2830页，2011年。

[111] R.Boyapati，J. Huang，P. Majumder，K. H. Yum和E. J. Kim，“近似noc:片上网络体系结构的数据近似框架”，在计算机体系结构国际研讨会(ISCA)上，2017年6月。

[112] A.Raha和V. Raghunathan，“走向全系统能量-精度权衡:近似智能相机系统的案例研究”，载于设计自动化会议(DAC)，2017年6月。

[113] A.桑普森、拜索、兰斯福德、莫罗、叶、策泽和米（meter的缩写））Oskin，“接受:实用近似计算的程序员指导编译器框架”，华盛顿大学技术报告，第1卷，2015年1月。

[114] B.雷根、埃尔南德斯-洛巴托、阿道夫、盖尔巴特、瓦特-穆格、魏国勇和布鲁克斯，“通过贝叶斯优化进行有效加速器设计空间探索的案例”，载于国际低功率电子与设计研讨会(ISLPED)，2017年7月。

[115] A.瓦莱罗、萨维诺、波利塔诺、卡洛、齐达米特里，南茨罗尼斯、卡利奥拉基斯、吉佐普洛斯、里埃拉、卡纳尔，A.Gonzalez，M. Kooli，A. Bosio和G. D. Natale，“硬件故障的跨层系统可靠性评估框架”，国际测试会议(ITC)，2016年11月。