通过超维计算对认知地图学习器进行模块化与组装

Nathan McDonald美国空军研究实验室，信息局 美国纽约州罗马市
Nathan.McDonald.5@us.af.mil

摘要— **生物体必须学会如何控制自身身体，以实现有意运动，即根据当前身体位置和所选择的动作预测下一个身体位置。这种学习与最大化（最小化）环境奖励（惩罚）信号的目标无关。认知地图学习器（CML）由三个彼此独立但协同训练的人工神经网络组成，用于学习为任意双向图的节点状态和边动作构建表示。通过这种方式，CML 学会如何遍历图节点；然而，CML 不会学习何时以及为何从一个节点状态移动到另一个节点状态。**

本研究创建了节点状态以适用于超维计算（超维计算，一种符号机器学习（ML）形式）的高维向量表示的CML。通过这种方式，图知识（CML）与目标节点选择（超维计算）相分离，使每种机器学习方法都能独立训练。第一种方法使用超维计算构建任意数量的层级 CML，其中每个图节点状态都为下一级 CML指定要遍历到的目标节点状态。其次，针对每个 CML，演示了一种基于超维计算的刺激-反应经验模型。由于超向量可以彼此叠加，因此可以将多个经验模型相加并行运行，而无需任何重新训练。最后，将一个 CML-HDC机器学习单元模块化：使用代理符号进行训练，使其能够处理任意的、特定应用刺激符号，而无需重新训练 CML或 HDC 模型。这些方法为构建异构机器学习系统提供了一个模板。

关键词——超维计算、认知地图、模块化、终身学习、状态表征学习、神经工程

I. 引言

尽管分类是人工神经网络 (ANN) 的一种常见应用，但大量研究表明，预测是生物神经网络认知过程中的关键组成部分 [1, 2]。例如，小牛出生后，会用最初几个小时学习协调腿部肌肉运动，以实现运动，即从一处有方向地移动到另一处。通过不断挥动四肢，它学会了自己可以处于哪些物理状态，以及可以采取哪些动作。传统的强化学习旨在最大化（或最小化）环境奖励（或惩罚） [3], , 而预测学习不受具体任务限制，它要最小化的是以下两者之间的误差：a) 实际观察到的下一个状态；b) 根据当前状态和所选动作预测出的下一个状态。

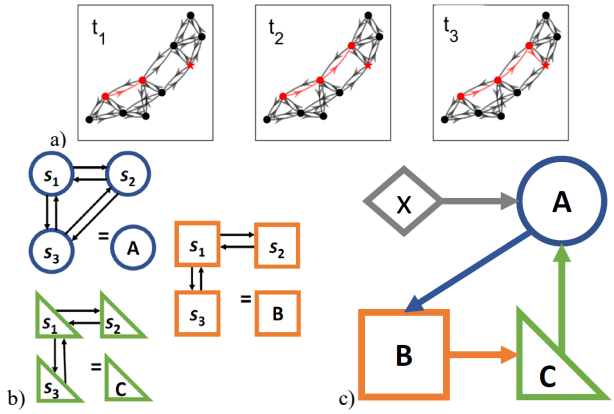


图 1. a）认知地图学习器（CML）迭代地从当前状态（红色节点）中选出通往用户指定目标状态（红色星形）的近似最优边集。b）通过将这些节点状态表示为高维向量，可对独立训练的 CML 进行模块化（字母形状），并且 c）依据超维计算规则任意集成，从而无需重新训练底层 CML 组件即可为机器学习任务提供工程解决方案。

学习双向图的拓扑结构，是对预测学习的一种抽象，其中每个节点代表一个可观测状态，每条边代表处于该状态时可采取的动作。近期提出了一种认知地图学习器 (CML)，它使用三个分别训练但相互协作的人工神经网络，构建以下内容的内部表征：a) 节点状态；b) 节点特定的边动作；以及 c) 边动作效用值 [4]。值得注意的是，尽管 CML 从未接受过路径规划方面的显式训练，用户仍可指定一个目标节点状态，CML 随后便会沿着一条近似最优路径（边数最少）从当前节点状态前往目标节点状态（图 1a）[4]。需要注意的是，CML 并未学会何时以及为何从一个节点状态转换到另一个节点状态；相反，必须由外部来源提供目标节点状态。

本研究探讨如何以有限状态机的形式（图 1b、c）整合并协调多个 CML 的数学方法；对于传统人工神经网络而言，这是一项众所周知的难题[5]。整合的一大挑战在于接口：确保输入和输出类型一致。例如，在数字逻辑 {0,1} 中，一致的信息表示可让经过独立优化的逻辑单元组装在一起，解决超出单个布尔运算范围的问题，例如将与、或和异或逻辑门组合成数字加法器。此外，接口一致还能让组件相互替换，而不必重新设计系统的其他部分。

已批准公开发布；不受分发限制：AFRL-2023-0658

本文提出使用长度为$d \geq 1e3$的高维向量作为信息表示，并依据超维计算（HDC）代数来组装和控制 CML；HDC 是向量符号架构（VSA）的一种形式[6, 7]。近年来，HDC 已成为最受欢迎的非人工神经网络机器学习方法之一。HDC 并非学习突触权重值，而是通过操纵一组由 {0, 1} [8], {-1, +1} [9], 或复数值[10]构成的超向量之间的相似度来编码学习。作为一种代数，这种学习可以明确表示为方程，便于编辑和逆向工程，从而支持人工理解和干预[11]。

本文的贡献如下：

- 经过训练的 CML 的节点状态矩阵生成详细的可行超向量，$d = 1e3, 1e4$。
- 表明 CML 可以围绕用户指定的节点状态矩阵进行训练，从而实现任意高层级 CML 的工程构建。
- 为单个 CML 实现了一个超维计算刺激-响应模型，用于学习并决定目标状态。
- 展示了将多个此类模型添加到一个单体式超维计算体验模型中，以并行控制多个 CML，而无需重新训练超维计算模型或底层 CML。
- 开发了一种将预训练的 CML-超维计算机器学习单元模块化的方法，该方法可接受特定于应用的超向量输入，而无需对超维计算模型或 CML 进行重新训练。

第二节介绍 CML 的构建与训练，并阐述超维计算的代数规则。第三节介绍如何根据 CML 节点状态表示生成超向量，从而为后续整合多个独立训练的 CML 奠定基础。第四节详述整合结果，随后第五节将讨论本研究及其未来应用。

II. 背景

A. 认知地图学习器（CML）

认知地图学习器是由三个彼此独立、但协同训练的单层人工神经网络组成的系统 [4]。尽管 CML 也可以学习其他任务，本研究聚焦于双向图（图 2）。每条边表示仅能在对应两个节点状态之间执行的动作。双向性要求每个动作都可逆。在本研究中，“节点”和“状态”将交替使用，“边”和“动作”也同样如此。

CML 学习三项基本内容：a) 每个节点的内部状态表示（W_q），b) 每个节点处可用动作的集合（W_k），以及 c) 给定节点处每个动作的效用（W_v）。给定一个包含 n 个节点和 e 条边、状态表示长度为 d 的图，矩阵 $W_q \in \mathrm{R}^{(d,n)}$ 与 $W_v \in \mathrm{R}^{(d,e)}$ 均初始化为随机高斯分布，$\mu = 0$，$\sigma = 1$，其分别为 . 和。WR初始化为零矩阵。d 在原论文中由用户定义，默认设为 $2n$。

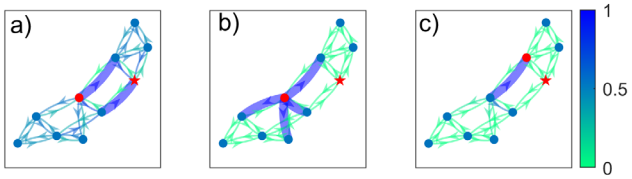


图 2. 给定起始节点状态 s_t（红色圆圈）和目标节点状态 s^*（红色星形），CML 的人工神经网络会针对每个边动作计算：a) 效用值 u_t，b) 在给定 s_t 的情况下的许可值 g_t，以及 c) 最终选择值 $g_t \odot u_t$。为便于视觉辨识，数值最高的边以更粗的线条显示。

在时间 t，CML 接收一个可观测状态向量 o_t，该向量来自单位矩阵 $I \in \mathrm{R}^{(n,n)}$，其中索引与当前节点索引相匹配。该观测的内部状态表示计算如下：

$$s_t = W_q\, o_t. \tag{1}$$

关键在于，由于o_t 来自单位矩阵，s_t 明确对应W_q的t^{th} 列；这一点将在本研究的后续部分加以利用。

动作向量 a_t 同样来自一个单位矩阵，$I \in \mathrm{R}^{(e,e)}$，其中图中的每条边都有唯一的索引。下一个可观测节点状态 $\hat{s}_{t+1}$根据当前状态和所选动作进行预测，

$$\hat{s}_{t+1} = s_t + W_v\, a_t. \tag{2}$$

每个训练轮次中，都会执行全部 e 个动作。为编码动作的可执行性，还会学习一个门控函数 g，

$$g_t = f_g(W_k\, s_t), \tag{3}$$

其中 f_g 是一个广义 sigmoid 函数

$$f_g(x) = \mathrm{clip}\left(\frac{\sigma(x) - \sigma(0)}{\sigma(\alpha) - \sigma(0)}, 0, 1\right), \tag{4}$$

其中 σ 是 sigmoid 函数，α 是定义该函数饱和点的常数。在学习阶段，$\alpha =$ ；而在测试阶段，.. 1 $\alpha = 0\,1$ [4]

为了训练 W_q、W_v 和 W_k，采用简单的 delta 学习规则 [12]；具体而言，变化量通过实际值与预测值之差乘以作用输入的转置来计算，

$$\Delta W_k(t) = l_k(a_t - g_t)s_t^{\mathsf{T}}, \tag{5}$$

$$\Delta W_v(t) = l_v(s_{t+1} - \hat{s}_{t+1})a_t^{\mathsf{T}}, \tag{6}$$

$$\Delta W_q(t) = l_q(\hat{s}_{t+1} - s_{t+1})o_{t+1}^{\mathsf{T}}, \tag{7}$$

其中 l $_k$ $_v$ $_q$ $_k = _v$、l 和 l $= _q = 0\,1$ 分别是相应的学习率。这里，lll。.为简化起见，权重更新量会累加，并在每个训练轮次结束时应用。随后进行正则化，以保持向量的单位长度，

$$W = \frac{w}{\|W\|^2}. \tag{8}$$

注意，W_v 沿e轴进行正则化，而W_k 沿d轴进行归一化。

要使用 CML 遍历其图，用户首先选择一个起始观测 o_t，设置当前状态 s_t，

(1). 接下来，用户选择一个目标观测 o^*，其目标状态为

$$s^* = W_q\, o^*. \tag{9}$$

其次，每个动作的效用通过将目标状态与当前状态之差乘以 W_v 矩阵的转置 [4] 来计算（图 2a），

$$u_t = W_v^\top (s^* - s_t). \tag{10}$$

第三，门控函数 g_t (3) 用于计算状态 s_t 下允许的动作索引（图 2b）。然后，将 g_t 与 u_t 逐元素相乘（图 2c），并应用胜者全得（WTA）函数生成独热向量，从而确定所选动作，

$$a_t = \mathrm{WTA}(g_t \odot u_t), \tag{11}$$

其中，⊙ 表示逐元素乘法。最后，计算下一个预测状态$\hat{s}$

$_{t+1}$ 2 10, （）；通过迭

11, 2

代执行（），CML 能够找到任意起始状态与目标状态之间（近似）最优的动作集合，尽管该网络从未针对这类遍历任务进行过显式训练（图 1a）。

B. 超维计算（超维计算）

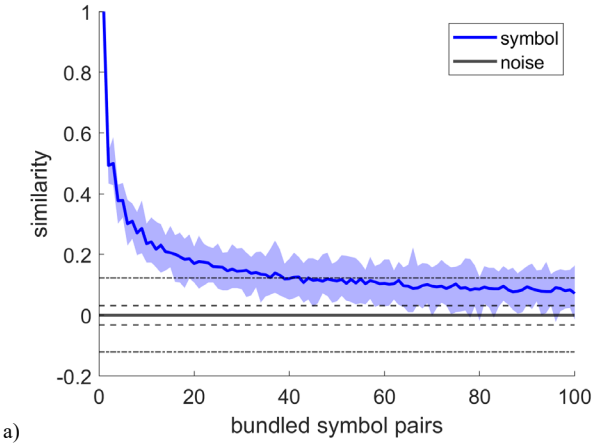
尽管超维计算所涉及的信息表示思想与大脑神经活动中的信息表示思想相同，但它是一种代数，而不是一类新的人工神经网络（ANN）。超维计算并不使用人工神经元和突触，而是利用超向量进行符号计算，超向量的长度为$d \geq$ 1e3。超维计算的基本度量是相似度，因此关注重点从不相似元素的位置（例如最高位/最低位以及纠错码）转移到了不匹配元素的数量。这样一来，每个元素对于定义特定符号都具有同等的（不）重要性。随着这些随机生成向量的长度增加，它们实际上必然会成为伪正交向量[6]。因此，如果两个符号不是伪正交的，那么它们之间必然存在某种相关性。在本研究中，根据乘法、加法和置换（MAP）方法[9]，使用了由均匀随机 {-1, +1} 元素组成的稠密超向量。

向量之间的相似度通过余弦相似度来衡量，即它们的点积除以各自模长的乘积，

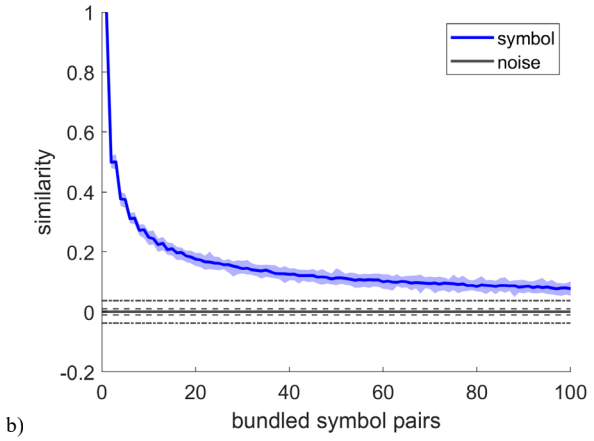
$$sim_{cosine} = \frac{x \cdot y}{\|x\| \|y\|}. \tag{12}$$

相同向量的余弦相似度为 1，而伪正交超向量的相似度接近 0。长度为 d = 1e3 的随机生成超向量，其平均相似度为 0.0 ± 0.032 （d = 1e4 时为 0.0 ± 0.009 ），但相似度范围最高可达 0.1（0.03）（图 3）。标准差或最大相似度值是否适合作为噪声底阈值 θ，取决于具体应用。

超维计算中的基本运算包括超向量的加法、乘法和置换。这些都是逐元素运算，因此无论多少个超向量



a)



b)

图 3. 对于 a) d = 1e3 和 b) d = 1e4，噪声符号相似度（蓝色）随捆绑在一起的绑定对数量变化（均值及其极值范围）。噪声底限（黑色）定义为随机符号之间的相似度（均值：实线；标准差：双虚线；极值：点划线）

相加或相乘，所得超向量的维度都保持为d。（置换只是对向量索引进行循环移位，记作 Π，本文未使用此操作。）

可以通过有符号加法将超向量相加（捆绑），并将数值截断回 {-1, +1}，

$$s = [x + y + z]. \tag{13}$$

捆绑偶数个超向量时，会加入一个随机超向量 η 以打破平局。由于信息编码在整个超向量中，捆绑操作类似于在 s 上叠加各个符号。给定复合超向量 s 和符号字典 D {w、x、y、z}，便可以识别（并重构）构成 s 的各个向量，

$$sim(s, x) \approx sim(s, y) \approx sim(s, z) >> sim(s, w). \tag{14}$$

超向量的乘法（或绑定），记作 ⊗，将两个符号绑定在一起，类似于键值配对。与加法不同，所得的超向量与其任一组成向量都不相似。在此，逐元素

乘法是绑定算子，并且它是一种自可逆运算。

令 $s = [w⊗x + y⊗z]$。要恢复与 w 绑定的超向量，

$$\begin{aligned} w⊗s &= w⊗[w⊗x + y⊗z] \\ &= \cancel{w⊗w}⊗x + w⊗y⊗z \\ &= x + \eta \end{aligned} \tag{15}$$

其中 η 是一个随机超向量。由于任意超向量几乎总是伪正交的，$w⊗y⊗z$可以视为等同于一个随机超向量 [8]。将$x+\eta$与字典D中的所有其他符号进行比较，可以发现其与符号x的相似度最高。然而，由于乘法具有分配性，随着绑定并捆绑在一起的符号数量增加，合并到 η 中的多余项数量会快速增加，从而降低 $sim(x,x+\eta)$（图 3）。只要噪声符号的相似度大于 θ，就可以通过字典搜索实现完整的纠错。

III. 方法

A. 训练超向量节点状态

为限制评估空间，本研究在随机生成的图上训练 CML，这些图包含 $n = \{10, 25\}$ 个节点和 $e \sim 2n$ 条边。成功训练的 CML 能够 a) 选择正确的边动作，在所有相邻节点对之间移动；并且 b) 在不调用当前状态下不可用的任何边动作的情况下，成功在 50 个随机选取的起始 s_t 和目标 s^* 节点状态之间转换。s_t 和 s^* 至少相隔两条边。状态向量长度 d 对 CML 性能的影响，通过随机初始化边、W_q、W_v 和 W_k时成功训练的 CML 所占比例来衡量。

对 50 个 CML 分别训练 500 个训练轮次，并在 (5-7) 中遍历 $d = 2n$，当 $n = 10$ 时，成功率达到 1（当 $n = 25$ 时为 0.92）；而当 n 和 $d = \{1e2, 1e3, 1e4\}$ 时，两者的成功率均达到 1。为便于开展超维计算，本文其余部分将超向量长度固定为 $d = \{1e3, 1e4\}$。

训练得到的 W_q 矩阵是稠密的，其值呈高斯分布，范围为 [-0.1、0.1]（图 4a）。每次尝试通过强制令 W_q 的值为 {-1, +1} 来训练可用的 CML，均以失败告终（未显示）。然而，应用符号函数，

$$sgn(x) := \begin{cases} -1 & if\ x < 0, \\ 0 & if\ x = 0, \\ 1 & if\ x > 0, \end{cases} \tag{16}$$

对 W_q 进行处理后，得到了预期的 a) 等概率 {-1, +1} 分布，以及 b) 节点状态的伪正交性，$sim(sgn(s_i), sgn(s_j))$，其中 $i \neq j$（图 4b）。这些基础结果使本文其余实验得以开展。

其他观察结果还包括明显的可分性：$sim(s_i, sign(s_i)) \approx 0.8$ 与 $sim(s_i, sign(s_j)) \approx 0$，其中 $i \neq j$（图 5）。这种高相似度意味着，在与 CML 交互时，可以直接将 $sign(s_t)$ 用作目标状态 s^*；不过，始终可以使用字典清理操作来恢复未经修改的 s_t。然而，为方便符号表示，除非另有说明，本文后续所有关于 W_q 和 s 的讨论均仅指其符号化版本。

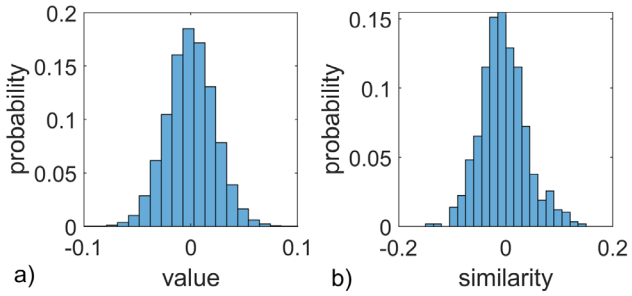


图 4. a)训练后的 W_q 值以及b) $n = 25$、$d = 1e3$ 时，状态超向量在 $sgn(W_q)$ 中的相似度

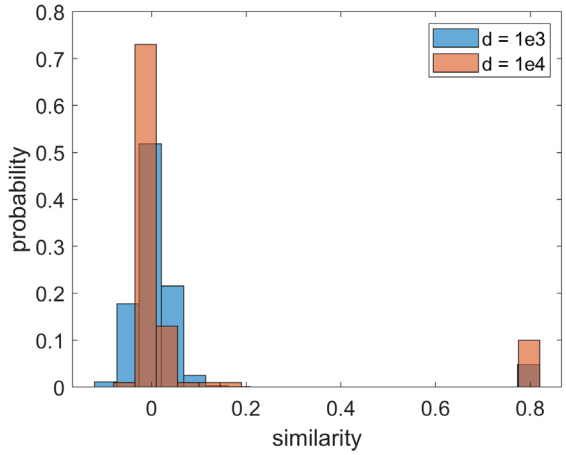


图 5. W_q 与$sign(W_q)$ 中节点状态之间相似度的直方图，其中$d = 1e3$（蓝色）和 1e4（橙色）。

最后，δ倾斜规则可以训练为适应预定义的 W_q。将 W_q 随机初始化为 {-1, +1} 值后再进行正则化（8），仍然达到了 CML 训练成功率 1。将“随机”稍作改为“任意”，便可采用多种方法将不同的 CML 集成在一起。

B. 训练超维计算经验模型

以下 HDC 刺激-反应经验模型在 [14] 中提出。HDC 用于将输入场景（任意 HDC 符号）与响应（CML 的一个节点状态）关联起来（图 6a）。每个m 个输入都从其自身包含k 个状态的字典中传递，记为 x_k^m。在此，输入是任意超向量，但在实践中，这些输入符号可能来自环境传感器或其他 CML 节点状态。

一个输入 场景 由一组m 个输入状态组成，

$$scene_i = [x_i^1 + x_i^2 + \ldots + x_i^m], \tag{17}$$

虽然为方便记 notation，在所有输入中都使用 i，但它指的是任意输入符号。在训练期间，每个 场景都会与一个目标节点状态配对，从而创建一个情境，

$$scenario_i = scene_i⊗s_i. \tag{18}$$

最后，通过将所有的 场景 捆绑在一起，创建了一个体验模型，

$$EXP = \sum scenario_i. \tag{19}$$

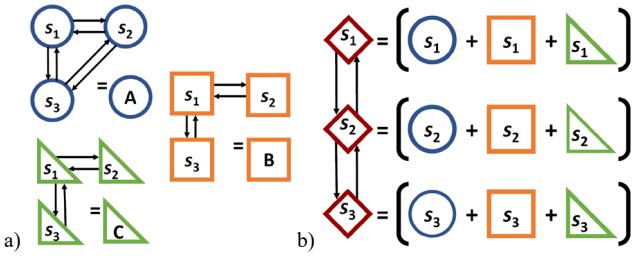


图 6. a) CML *A*（蓝色圆形）、*B*（橙色方形）和*C*（绿色三角形），其节点在 b) 中捆绑为层次化 CML 的节点状态

最佳实践是，通过使用训练场景对其进行查询来验证 *EXP* 模型，

$$response^{val} = scene_i \otimes EXP. \quad (20)$$

测量响应^{val} 与字典W_q 之间的相似度，并返回高于噪声底阈值 θ 且相似度最高的节点状态，作为指定的目标状态，

$$s_i = cleanup(response, W_q, \theta). \quad (21)$$

只要响应^{val} 的相似度值保持高于噪声底限，*EXP*就会返回正确的目标状态。

由于任何响应与其目标状态的相似度只需大于 θ，因此可以使用由额外输入符号构成的场景查询已部署的*EXP*模型，而无需重新计算*EXP*,

$$query = [x_i^1 + x_i^2 + \ldots + x_i^m + \eta], \quad (22)$$

$$response = query \otimes EXP, \quad (23)$$

按此前方式继续清理，(21)。对于足够相似的响应（>θ），CML 将接受新的目标状态，并开始遍历其图。如果响应不够相似，CML 会忽略它。

为衡量*EXP* 模型的性能，选择噪声相似度阈值，以最大限度地减少虚假的 CML状态变化（假阳性），同时接受每个场景 中遗漏的CML状态转换（假阴性）增加。特异度测量为真阴性（TN）占真阴性总数与假阳性（FP）总数之和的比例，

$$specificity = \frac{TN}{TN + FP}. \quad (24)$$

灵敏度的测量值为真阳性占真阳性与假阴性（FN）总数的 (TP) 比例

$$sensitivity = \frac{TP}{TP + FN}. \quad (25)$$

IV. 结果

对于以下每个实验，均按照实验标准对随机生成的 CML 进行训练和测试。报告的数值是每种参数设置下 5 次试验的平均值。每个 CML 的节点数为 n = 10, { }, 25 = 1 3, 1 4 超向量长度为 d { e e }。

A. CML 层级结构

通过超向量节点状态对 CML 进行最简单的扩展，是构建层级 CML。一个新的 W_q

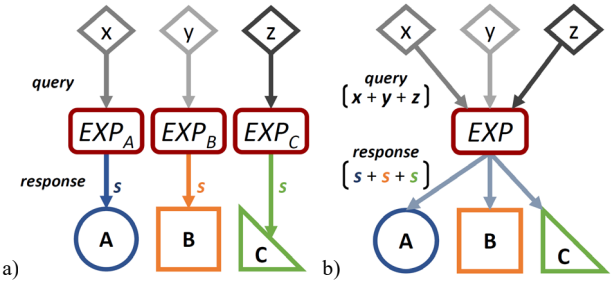


图 7. a) 三个独立训练的 CML（形状为*A*、*B*、*C*），以及各自的 *EXP*。b) 捆绑*EXP*使查询*query*和响应*response*均得以整合。

通过捆绑 n 个节点构建 n 个节点的节点状态矩阵，每个节点状态分别来自 n 个低层 CML（图 6）。高层 CML 围绕这个预定义的 W_q 进行训练。随着高层 CML 遍历其双向图，它的每个节点状态都会为 n 个低层 CML 中的每一个指定目标状态，由这些低层 CML 执行所需的清理操作（21）。

由于 CML 可以围绕随机 W_q 矩阵进行训练，而这些矩阵也会生成伪正交超向量，因此，只需使用随机 d e 超向量模拟 50 = 10，层层级 25 = 1 4 CML n { } 即可。从符号 s^1，（第 1 层的目标符号）开始，通过将 s^1 与 n-1 个随机生成的超向量捆绑，创建下一层的节点状态 s^2，以表示这样一种 CML：其中每个节点状态都是其他 n 个 CML 节点状态的总和（图 6b）。更一般地，每个 CML 层级的 s^{v+1} 节点状态计算如下

$$s^{v+1} = [s^v + \sum^{n-1} \eta]. \quad (26)$$

参见图 3，s^1 到s^3时，与更高层节点变为伪正交（无法从中恢复）。然而，相邻层节点s^v 和s^{v+1}之间的相似度仍保持在 0.24 ± 0.04（n = 10 ）和 0.16 ± 0.02（n = 25）的稳定水平，远高于噪声底限。由于每个v^{th} 层 CML 只需根据自身的节点状态字典清理接收到的v+1 状态，因此这一结果表明，可以从任意s^v 节点出发，始终逐步重建捆绑在一起并用于构成该节点的较低层节点。

B. 单体式超维计算体验模型

然而，层级 CML 只是将初始目标状态选择算法进一步推至 CML 层级的更高处。直接使用超维计算进行符号机器学习，使模型能够学习输入符号与 CML 目标状态之间的关联（图 7）。CML *A*、*B* 和 *C* 分别接收输入x、y和z，每种输入包含 =1-5 个符号。每个*EXP*模型学习了k种情境，将每个输入与一个目标状态相关联，

$$EXP_A = \sum_{i=1}^{k} x_i \otimes s_i^A, \quad (27)$$

$$EXP_B = \sum_{i=1}^{k} y_i \otimes s_i^B, \quad (28)$$

$$EXP_C = \sum_{i=1}^{k} z_i \otimes s_i^C. \quad (29)$$

验证每个经验模型（20）后，将它们捆绑在一起，无需重新训练，从而构建一个整体式*EXP*，

$$EXP = [EXP_A + EXP_B + EXP_C]. \quad (30)$$

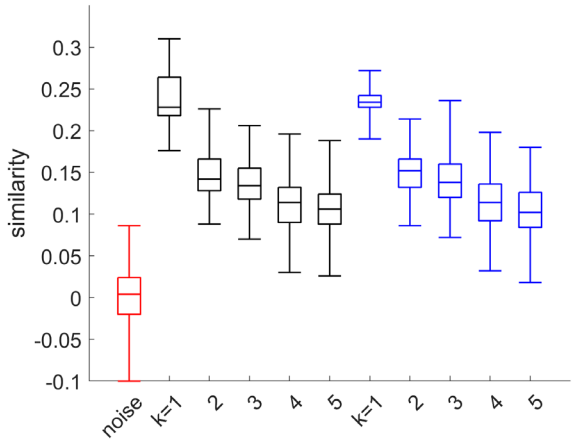


图 8. 在 3k 个学习输入状态下，d = 1e3 时噪声（红色）和响应与目标状态s_i 的相似度箱线图，n = 10 （黑色）和n = 25 （蓝色)

使用全部 3k 个训练场景对这个组合EXP进行验证，并在清理后对所得目标状态进行分类（20, 21）。对于所有n和d组合，组合后的EXP在训练场景上的灵敏度和特异度均为 1。

为测试这种捆绑式EXP方法的稳健性，每个输入都扩展为 2k 个可能的符号，包括 k 个已训练符号和 k 个新符号。每个测试周期都遍历 3k 个训练场景，并从另外两个输入中随机选取符号进行捆绑。也就是说，每个测试场景包含 3 个符号，其组合由 1–3 个已训练符号和 0–2 个新符号构成。查询EXP时，这个捆绑后的场景会产生一个捆绑后的响应（图 7b），

$$response = [x_i + y_i + z_i] \otimes EXP = [s_i^A + s_i^B + s_i^C + \eta]. \quad (31)$$

同一个响应向量被传递给每个 CML，以便根据各自的W_q进行清理。

每个这样的测试周期重复 10 次，每次随机选择另外的输入符号索引；对于 5 次试验，每次测试周期包含 30k 个场景（3 个输入、k 个已训练符号、10 个周期）。将组合后的响应的相似度与每个 CML 的W_q进行比较。图中绘出了响应与其目标节点状态的相似度随k变化的情况（图 8、9）；箱线图从上到下依次表示最大值、第三四分位数、均值、第一四分位数和最小值。

如前所述，噪声底限是已学习场景数量的限制因素。由于 3 个 CML 中的任何一个都可能在每个场景中没有收到新的目标状态 (31)，因此将噪声底阈值设为噪声的最大相似度，以避免“恢复”出虚假的目标状态。不出所料，增大k（对应于在EXP中捆绑的 3k 个场景）降低了总体响应相似度。对于d = 1e3，两种n下的响应相似度在k = 3 时均与噪声底限重叠（图 8）。阈值 θ = 0.08 使两种n下的特异度都接近完美，达到 ≥ 0.99，但k下的灵敏度有所降低（表 I）。对于d = 1e4，两种n下的响应相似度都保持在噪声底限之上（图 9），因此在 θ = 0.04 时，灵敏度和特异度均达到完美水平。也测试了 k = 10，灵敏度仅略微下降至 0.99 ± 0.01。

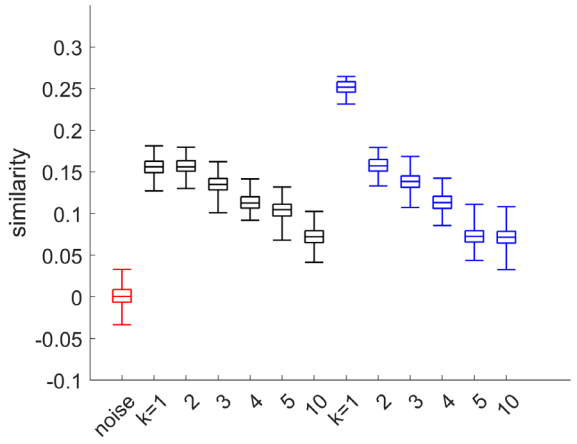


图 9. 噪声（红色）与响应对s_i 目标状态的相似度箱线图，覆盖 3k 个学习输入状态；对应d = 1e4、n = 10 （黑色）和n = 25 （蓝色）

表 I。 单体 EXP灵敏度，d = 1E3

		k = 1	2	3	4	5
n	10	1	1	0.97 ± 0.01	0.89 ± 0.03	0.86 ± 0.04
	25	1	0.99 ± 0.01	0.95 ± 0.03	0.88 ± 0.02	0.82 ± 0.02

C. 带代理符号的 CML 接口

尽管EXP能够抵御一定程度的额外传感器输入 [13]，但如果训练输入符号集发生变化，就需要使用新的符号集重新训练EXP（图 1c、10a）。无需每次输入变化时都重新学习特定的场景，而是可以使用超维计算通过代理输入符号对场景进行编码，并将应用特定输入映射到这些代理符号。例如，在图 10a 中，CMLA的输入为x和s^D；不过，超维计算可以创建一个包含两个输入的通用EXP（图 10b）。

模板场景 场景^0 由 m =1-8 个输入创建，每个输入包含 k =1-8 个代理状态，

$$scene_i^0 = [p_i^1 + p_i^2 + \ldots + p_i^m]. \quad (32)$$

这些 场景被绑定到k 个目标节点状态，

$$EXP = \sum_{i=1}^{k} scene_i^0 \otimes s_i. \quad (33)$$

要与 CML 交互，需要了解代理符号 p、场景^0 模板和 W_q 节点状态。随后，将一个新符号 x 分别与每个 km 已学习的代理输入符号绑定。然后将这些绑定对捆绑在一起，创建一个 映射，

$$map = \sum_{i=1}^{k} [x_i^1 \otimes p_i^1 + x_i^2 \otimes p_i^2 + \ldots + x_i^m \otimes p_i^m]. \quad (34)$$

在测试期间，将特定于应用的场景与映射相乘，得到一个近似版本的场景^0 模板，

$$\begin{aligned} query &= scene \otimes map \\ &= [x_i^1 + x_i^2 + \ldots + x_i^m] \otimes map \\ &= [\not{x}_i^1 \otimes \not{x}_i^1 \otimes p_i^1 + \not{x}_i^2 \otimes \not{x}_i^2 \otimes p_i^2 + \ldots + \not{x}_i^m \otimes \not{x}_i^m \otimes p_i^m + \eta] \\ &= [p_i^1 + p_i^2 + \ldots + p_i^m + \eta]. \end{aligned} \quad (35)$$

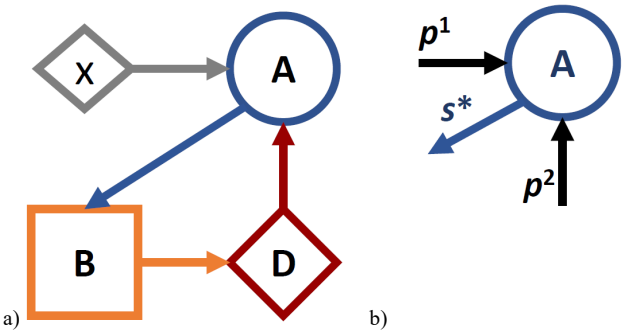


图 10. a) 集成超维计算的 CML 支持模块替换（见图 1c），无需对整个机器学习系统进行完整的重新训练。b) CML 的 *EXP* 模型可使用任务无关代理符号进行训练，随后将这些符号映射到特定应用输入 x 和 s^D。

请注意，由于乘法满足结合律，η 汇总了大量无意义项，例如 $x_i^2 \otimes x_i^1 \otimes p_i^1$ 出现在（35）中。这些噪声项导致符号严重退化（图 3），因此场景^0 模板对于在照常计算响应之前清理查询至关重要，

$$scene_i^0 = cleanup(query, scene^0, \theta). \quad (36)$$

由于用户可以访问代理符号p和W_q，因此可以根据需要创建其他场景^0 和情境^0 模板。最后，由于预期每个查询都能恢复目标状态，因此使用噪声底限的标准差来设定阈值，而非使用最大范围。

完整的映射返回了一个查询 = 映射⊗场景，其与正确场景^0 模板的相似度最高。通过随机生成的超向量（代表任意应用特定输入）生成此类映射的一致性，以生成训练好的*EXP*对应的 10 个完整映射所需试验的比例来衡量，其中d = 1e3（图 11a）和d = 1e4（图 11b）。（所有实验中n =25，但只使用了k 个节点状态。）如果 300 次尝试内未能生成 10 个映射，则将一致性标记为 0。对于d = 1e3，尽管成功生成映射的一致性迅速下降，但仍能为$mk \leq 35$ 个符号找到完整的映射。使用 d = 1e4 时，一致性下降得更慢，并且除mk = 64外，所有m和k组合都能找到映射。

虽然生成映射的成功率很高，尤其是对于d = 1e4，但将映射后的查询 (35) 直接应用于*EXP*，会导致响应相似度迅速下降，进而降低目标节点的灵敏度（超过简单的$m \leq 2$ or $k \leq 2$后）（图 12）。执行查询清理步骤 (36) 后，图 11 中每个成功的映射都表现出完美的响应灵敏度，值为 1。

V. 讨论

CML 的人工神经网络之间的知识分离，使得能够精确提取（或定义）节点状态表示。将 CML 节点状态表示为超向量后，便可使用超维计算来组装并并行运行独立训练的 CML，而无需后续重新训练。由于噪声底限是这些实验中的限制因素，采用长度为 $d \geq$ 1e3 的超向量，确保能够获得最大的相似度范围。

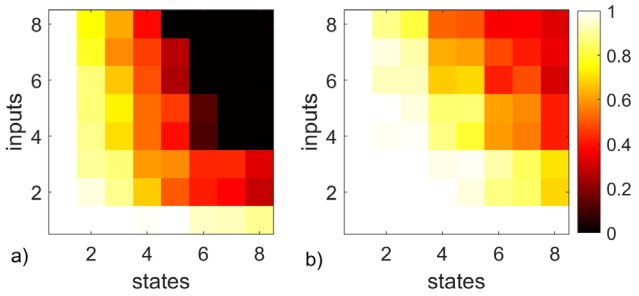


图 11. 在任意输入向量与代理向量之间生成成功映射的概率：a) d = 1e3；b) d = 1e4。

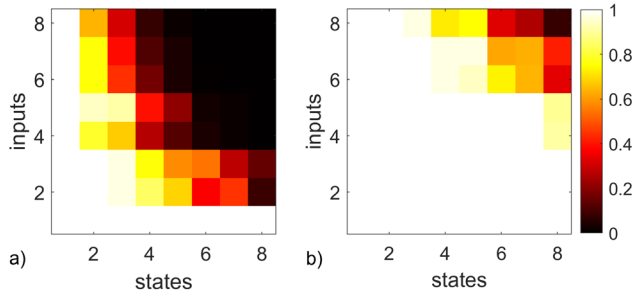


图 12. 使用 查询 = 映射⊗场景 且不进行清理来调用响应时的准确率：a) d = 1e3；b) d = 1e4。

终身学习是指在最大限度减少既有知识退化的情况下，按顺序适应新的学习内容或行为的能力 [15]。CML算法上实现了将图遍历与目标节点决策机制分离，从而能够构建更复杂的机器学习系统。超维计算允许输入<शैली id='2'>场景存在一定变化时产生相同的< शैली id='4'>响应，否则可以将新的< शैली id='6'>场景轻松添加到当前的< शैलीid='8'>EXP模型中。或者，还可以使用更高层级的机器学习算法，根据环境或历史背景，在多个可用模型中决定最合适的< शैली id='10'>EXP。

面向机器人技术的“即插即用”机器学习是这些 CML-HDC 模块的一种预期应用。图中的每个节点都可以表示机械臂的一个位置，而图本身则定义一种特定行为，例如抓取或行走。针对每个附肢训练一个 CML-HDC *EXP* 输入-响应模型后，可以将附肢添加到基准机器人上（或将其移除），并在无需额外重新训练的情况下运行它（前提是目标相似度仍高于噪声底限）。

然而，四足行走等任务需要（可能不对称的）附肢之间相互协作。将四个 CML 的节点状态捆绑起来，是创建层次化 CML 的一种简单方法，其节点定义了四个腿部 CML 各自的状态。例如，每个高层 CML 节点都可以表示埃德沃德·迈布里奇动物运动照片中的一帧 [16]。

此应用还展示了 CML 通过代理符号进行接口连接的优势。即使假设四条腿完全相同，输入符号也可能因前后和左右腿部构型而异。可以针对预期的输入和输出范围训练一个 CML-HDC 模型，然后将每条腿分别映射到同一个 *EXP* 模型。

超维计算要得到更广泛应用，一大挑战是需要开发算法，将实值传感器数据映射为超向量符号 [17, 18]。然而，人工神经网络在分类领域有着丰富的历史：将原始传感器数据映射到任意类别标签。近期已有研究将人工神经网络训练为与任务无关的特征提取器 [5, 19] ，然后将这些稀疏特征向量映射为任意超向量符号，以供后续的超维计算使用 [20, 21, 22]。实际上，这种方法将人工神经网络本身转变为模块化的机器学习组件，发挥着相当于模拟转数字（A2D）转换器的机器学习功能。

最后，CML算法运行于实值神经网络之上；但文中所述的示例性生物学实例基于脉冲神经网络（SNN）。未来的研究将重点放在实现基于共振-发放（RF）神经元的SNN版本CMLs上 [23]。这类SNN根据神经元在周期τ内发放脉冲的时间对信息进行编码，而不是采用速率编码，即测量时间窗口内的脉冲数量。重要的是，相对于周期为 τ 的局部振荡器，在时间 t发放的脉冲可以表示为复值相量（或相位向量）。因此，RF神经元还可通过基于复相量的全息约简表示（HRR）实现HDC接口 [10]。

VI. 结论

认知地图学习器（CML）由一组彼此独立但协同训练的人工神经网络组成，用于学习遍历双向图。本研究创建了 CML，其图节点状态以高维向量表示，并具备超维计算（HDC）这一符号机器学习形式所需的数学属性。将 CML 节点状态表示为超向量，使得能够利用 HDC 组装并并行运行独立训练的 CML，而无需后续重新训练。本研究构建了任意数量的层级 CML，其中每个图节点状态都指定了下一低层 CML 要遍历的目标节点状态。为每个 CML 创建了基于 HDC 的刺激－响应经验模型，随后将这些模型捆绑在一起，从而再次实现并行运行，且无需任何重新训练。最后，我们开发了一种映射算法，使 HDC 模型能够在代理符号上训练，再将这些符号映射到特定应用的输入符号，从而创建可组合的 CML-HDC 机器学习模块。

致谢

本材料中表达的任何意见、发现和结论或建议均属于作者观点，不一定反映美国政府、国防部或空军研究实验室的看法。

分发声明 A。批准公开发布；无限制分发：AFRL-2023-0658

参考文献

[1] A. Alink, C. Schwiedrzik, A. Kohler,W.Singer, andL. Muckli，“刺激可预测性降低初级视觉皮层中的响应”，《神经科学杂志》，第30卷，第 8 期，第2960-2966 页，2010[2] F. De Lange,M. Heilbron,and P. Kok，“预期如何塑造知觉？”《认知科学趋势》，第 22 卷，779，2018 9，764第。页 -

[3] Y. Duan、X. Chen、R. Houthooft、J. Schulman 和 P. Abbeel，“连续控制的深度强化学习基准测试”，机器学习国际会议，第 1329-1338 页，PMLR，2016。[4] C. Stoeckl 和 W. Maass，“局部预测学习使神经网络能够进行规划”，bioRxiv，2022[5] S.Duan、S.Yu 和J. Príncipe，“通过使用核的成对学习实现深度学习模块化”，IEEE神经网络与学习系统汇刊，第33卷，第 4 期，第1441-1451 页，2021[6] D.Kleyko、D. Rachkovskij、E.Osipov、A. Rahimi，“超维计算（又称向量符号架构）综述，第一部分：模型与数据变换”，ACM ComputingSurveys，第55卷，第 6 期，第 130 号，第 1–40 页，2023[7] D.Kleyko、D. Rachkovskij、E.Osipov、A. Rahimi，“超维计算（又称向量符号架构）综述，第二部分：应用、认知模型与挑战”，ACM ComputingSurveys，第55卷，第 9 期，第 175号，第1–52 页，2023[8] P.Kanerva，“超维计算：使用高维随机向量进行分布式表示计算的介绍”，认知计算，第 1 卷，第 139-159页，2009[9] R. Gayler，“乘法绑定、表示算子与类比（研讨会海报）”，（1998）。[10] T. Plate，“全息约简表示”，IEEE 神经网络汇刊，第6 卷，第 3 期，第 623-641 页，1995[11] P. Neubert、S.Schubert 和 P. Protzel，“面向机器人技术的超维计算介绍”，KI-Künstliche Intelligenz，第 33卷，第 319-330 页，2019[12] J. Hertz、A. Krogh 和 R. Palmer，神经计算理论导论。CRC Press，2018。[13] D. Kleyko、E. Osipov、R.Gayler、A. Khan 和A.Dyer，“使用向量符号架构模拟蜜蜂的概念学习过程”，生物启发认知架构，第 14 卷，第 57-72页，2015 [14] N.McDonald、R. Davis、L.Loomis和J.Kopra，“面向机器人技术的超维计算：迁移学习、克隆、额外传感器与网络拓扑结构”，SPIE 信息科学中的颠覆性技术V，第 11751 卷，第21-33 页，2021[15] D. Kudithipudi，等，“终身学习机器的生物学基础”，自然机器智能，第4 卷，第 3 期，第 196-210 页，2022[16] E. Muybridge，《运动中的动物》，Courier Corporation，2012[17] A. Mitrokhin、P. Sutor、C.Fermüller 和Y.Aloimonos，“使用神经形态传感器学习感知运动控制：迈向超维主动感知”，科学机器人，第 4 卷，第 30期，eaaw6736，2019[18] P. Neubert、S. Schubert、K. Schlegel 和 P. Protzel，“作为视觉地点识别描述符的向量语义表示”，机器人技术：科学与系统，第 1-11页，2021[19] M. Gummadi、D. Kent、J. Mendez 和 E.Eaton，“SHELS：用于新颖性检测和无类别边界持续学习的互斥特征集”，PMLR 终身学习智能体会议，第1065-1085 页，2022[20] M. Hersche、G. Karunaratne、G. Cherubini、L. Benini、A.Sebastian 和 A. Rahimi，“受约束的小样本类增量学习”，IEEE/CVF计算机视觉与模式识别会议，第 9057-9067 页，2022[21] G. Karunaratne、M. Schmuck、M.Le Gallo、G. Cherubini、L. Benini、A. Sebastian 和 A. Rahimi，“稳健的高维记忆增强神经网络”，自然通信，第 12 卷，第 1 期，第 2468 页，2021[22] A. Hernandez-Cane、N. Matsumoto、E. Ping 和 M.Imani，“Onlinehd：使用超维系统进行稳健、高效且单遍的在线学习”，IEEE 欧洲设计、自动化与测试会议及展览会（DATE），第 56-61 页，2021[23] Frady、E.Paxon 和 Friedrich T. Sommer。“使用节律性脉冲模式进行稳健计算。”美国国家科学院院刊，第 116 卷，第 36 期，第 18050-18059 页，2019。