arXiv:2304.01086v1 [cs.NE] 3 Apr 2023

自构建神经网络

Andrea
Ferigo Giovanni
Iacca 特伦托大学
意大利特伦托

摘要
在生命早期，大脑会通过一种称为突触发生的过程，在学习的同时发育。神经元在生长并彼此互动时，会形成突触。然而，大脑最终会剪除这些突触。以往的研究将学习和剪枝作为独立过程来探讨，而在本研究中，我们提出了一种具有生物学合理性的模型，通过结合赫布学习与剪枝，旨在模拟突触发生过程。如此一来，智能体在学习如何解决任务的同时，会将其经验转化为特定的网络结构。也就是说，在执行任务的过程中，网络结构会自行构建。我们将这种方法称为自构建神经网络（SBNN）。我们将提出的 SBNN 与传统神经网络（神经网络）在 OpenAI 的三项经典控制任务上进行比较。结果表明，我们的模型总体上优于传统神经网络。此外，我们观察到，随着剪枝率增加，我们的模型性能下降幅度小于传统神经网络。最后，我们进行了验证测试，使用学习阶段未见过的任务对模型进行测试。在这种情况下，结果表明，SBNN 能比传统神经网络更好地适应新任务，尤其是在剪除超过 80% 的权重时。

CCS 概念
• **计算方法** → **神经网络**；遗传算法；人工生命；• **计算理论** → **学习模型**。

关键词
神经网络、可塑性、剪枝、神经进化

1 引言

自然大脑是我们所知最复杂的系统之一。它能够执行复杂任务，并以当前任何现代人工智能（AI）系统都无法企及的效率适应新情况。这些性能源于一个漫长的进化过程，该过程协调了大量以不同尺度运作的各种元素。例如，在分子层面，一些细胞必须通过交换三磷酸腺苷（ATP）为其他细胞提供能量。而另一些分子则用于调节细胞间通信。在细胞层面，干细胞必须特化为胶质细胞或神经元细胞。最终，这些细胞生长并建立连接，组成我们的大脑。这一过程称为突触发生 [44]，，随后是第二个过程，该过程会剪除相关性较低的突触 [3, 41]。人们认为，这一剪枝阶段是大脑如此高效的主要原因之一 [8, 26, 38]。

在这项工作中，我们关注最后一个方面：大脑中神经元的增长与组织。具体而言，我们感兴趣的是将增长与组织同智能体在执行任务期间的经验联系起来，正如自然大脑中发生的那样。为了模拟这一过程，我们结合了文献中两种广为人知的机制。

首先，我们考虑一个神经网络（NN），并在其上应用一种突触可塑性机制。该机制会根据智能体在任务执行过程中的经验改变神经网络的突触。在这种情况下，我们采用赫布学习 [5]，这一任务无关的可塑性模型，其灵感源自天然神经元。

其次，我们在模型中加入一种基于全局幅值算法 [2, 17] 的剪枝机制。不过，我们对该算法进行了改进，使其不仅能决定剪枝多少（即剪除多少个突触），还能决定何时剪枝。

我们将由此得到的模型称为自构建神经网络（SBNN），因为它能够根据智能体在其生命周期中获得的经验来构建自身结构。

我们在 OpenAI 的三个经典控制任务中测试了所提出的 SBNN，以展示该模型在性能方面的能力，以及网络结构如何因任务不同而有所差异。在另一组实验中，我们还评估了 SBNN 的泛化能力。

本文其余部分安排如下。第 2 节总结相关工作。第 3 节介绍方法，尤其是提出的 SBNN。随后，第 4 节展示结果，最后在第 5 节给出结论。

2 相关工作

如上一节所述，我们的 SBNN 基于两种机制：可塑性和剪枝。

关于第一种机制，可塑性是一种局部学习机制，用于调节神经元的增长及其对刺激的适应，同时也负责记忆机制 [36, 43]。

一种在生物学上合理的可塑性模型是赫布模型 [5,20]，，其源于这样一种直觉：如果某个突触经常被使用，它就应该得到加强。换言之，如果突触前神经元更频繁地引发突触后神经元的激活，那么这种连接就会变得更加重要 [6]。基于这一理念，人们开发了不同的赫布模型 [9, 12, 40, 42]。在这里，我们特别考虑所谓的 ABCD模型，该模型已被证明能够有效地优化智能体在若干控制任务中的表现 [10, 29, 34, 35, 39] 。

另一方面，剪枝是一个更高层次的过程，但对认知发育至关重要。例如，据估计，人类在青春期会失去约 40% 的突触，而记忆或认知能力却不会因此退化 [3]。

近年来，人工智能领域对剪枝给予了广泛关注，旨在提升性能和鲁棒性，尤其是在深度神经网络 [7, 14–16, 22, 27, 31]的背景下。此外，其他一些研究收集了证据，表明可塑性

实际上可被视为一种隐式剪枝，尤其是在突触连接能够使其数值饱和，从而在权重上形成准二值掩码 [10, 25, 32, 33]的情况下。然而，这些研究并未采用显式的剪枝机制。

3 方法

如前所述，我们旨在构建一个能够模拟突触发生过程的网络。本节将介绍该网络的结构及其行为。随后，我们将简要描述优化过程，以及实验中使用的经典控制任务。我们使用这些环境来衡量所提出的 SBNN 的性能，并证明它们如何在执行任务时（以及根据当前任务）改变自身结构。

3.1 赫布学习

赫布学习是一种可塑性模型，允许神经网络在执行任务期间改变其权重。重要的是，这种变化与任务的奖励无关，因为它仅基于各个突触的局部信息，尤其是突触前神经元和突触后神经元的激活。本文使用的 ABCD 模型按照以下规则，在网络每次前向传播后更新权重：

$$w_{i,j} = w_{i,j} + \eta(Aa_i + Ba_j + Ca_i a_j + D)$$

其中，a_i是突触前激活值，a_j是突触后值，而$w_{i,\ j}$是两个神经元之间连接上的权重。A、B、C和 D 是需要优化的参数。

3.2 剪枝机制

剪枝机制旨在找到神经网络的一个子集，其性能与原始网络相当（或更优）。该过程通过根据给定策略移除连接来实现。在本工作中，我们采用全局幅值剪枝算法 [2, 17]，该算法仅包括移除所有权重绝对值小于某一阈值的连接，该阈值定义为第 pr 个百分位数，其中 pr 是期望的剪枝率（即要移除的连接百分比）。

3.3 自构建神经网络

在突触发生过程中，神经元会探索细胞外空间，并尽可能与其他神经元建立连接[24, 37]。

在此，我们旨在通过允许任意两个神经元直接连接来模拟这一机制。

我们的模型按如下方式运行。我们从一个由 I 个输入、H 个隐藏节点和 O 个输出组成的神经网络开始。在任务的第一个回合中，I 个输入与所有 H 个隐藏节点和O 个输出相连。反过来，H 个隐藏节点彼此完全连接（不包括自环），并且与所有 O 个输出相连。连接总数 C表示为 H 的函数（这是唯一的超参数，因为 I和 O取决于任务），为 $C(H) = H^2+H\times(I+O) +I\times O$。总体而言，网络的内部结构类似于玻尔兹曼机 [1, 11, 21, 30]，，其中隐藏节点彼此完全连接。不过，我们也将输入直接连接到输出，因为在某些情况下，隐藏节点的计算能力可能并非必需。

我们将所有这些连接的权重初始化为零。这样，我们试图模拟神经元之间不存在连接的初始状态。然后，在 每个回合中，赫布过程会根据第 3.1 节所述的 ABCD规则更新权重。请注意，在我们的模型中，网络中的每个连接都有自己的 ABCD规则及相应的参数。这样，网络便能根据智能体在任务过程中积累的经验自行调整。我们为每个连接使用一条赫布规则，因为若从所有权重均为 0 的状态开始，使用单一的赫布规则可能会导致所有权重朝同一方向变化，进而使学习失效。

突触发生的第二步是对突触进行剪枝的过程，如第 3 节第 3.2 节所述。正如我们稍后将描述的，与赫布学习不同，剪枝在跨回合间进行。请注意，一旦开始剪枝，赫布学习便会停止。图2概述了剪枝过程，展示了初始网络的状态及其发育过程。从形式上讲，我们可以分析剪枝前后的网络。具体而言，在剪枝之前，隐藏节点彼此完全连接，且所有输入都与所有隐藏节点相连。因此，无法定义固定的激活顺序。于是，我们保留整体激活顺序：首先是输入，然后是隐藏节点，最后是输出。然而，对于隐藏节点，我们随机选择其激活顺序。

剪枝后，剩余的连接定义了网络。与之前不同，在这个阶段，我们可以更容易地定义激活顺序，因为剪枝机制自然会消除大多数环，尤其是在移除的连接比例足够高时。因此，为了找到激活顺序，我们可以对下划线图 $G(V, E)$ 进行拓扑排序，其中 V 是节点（即神经元）的集合，E 是连接的集合。如果在拓扑排序过程中发现环，我们就应用以下过程。用 N_c 表示 V 的子集，该子集包含环中的所有节点；首先，我们从 V 中移除 N_c 中的所有节点，并用一个虚拟节点 f 替换它们。然后，用 $E(N_c)^{incoming} = \{(v, n)\ \ \forall v \in V \setminus N_c \ \wedge \ \forall n \in N_c\}$ 表示 E 的子集，该子集由终止于 N_c 且不以 N_c 为起点的连接组成；我们将连接集合 $\{(v, f)\ \ \forall (v, v') \in E(N_c)_{incoming} \wedge v' \in N_c\}$ 添加到 E 中。对于从 N_c 出发的连接，我们也执行相同的操作。我们对图中发现的每个环独立地迭代应用这一过程，从而得到新图 G'，其中环 N_c 被虚拟节点 f 替换。现在，所有与 N_c 相连的节点都与 f 相连，而 f 则与从 N_c 可到达的所有节点相连。我们在变量 $cycles_history$ 中存储节点 f 替换了 N_c 个节点这一信息，然后再次尝试寻找拓扑序。我们重复这一过程，直到所有环都被替换，并且能够定义拓扑序。算法1展示了这一简化过程。请注意，N_c 中的节点也可以是该过程前一次迭代生成的虚拟节点，如图1所示。

计算出网络的拓扑序后，我们可以按照该顺序激活隐藏节点。如果在激活过程中发现虚拟节点，我们就从 $cycles_history$ 中取出组成该环的 N_c 节点集合，并按照随机激活顺序继续处理。如果 N_c 中的一个节点是覆盖 $N_{c'}$的虚拟节点 f，

1 函数 removeCycle(G):2 $N_c \leftarrow$findCycle(G) 3 $i \leftarrow 0$4 $cycleHistory \leftarrow map()$ 当 N_c不为空时执行 5 $E_{incoming} \leftarrow G$.入边($N_c$) 6 $E_{outgoing} \leftarrow G$.出边($N_c$) 7 G.removeNodes(N_c)8 G.addNode($fakeNode_i$) 9 $cycleHistory$.addKeyValue($fakeNode_i, N_c$) 10 对于每个 (v, v')，执行 $\in E_{incoming}$ 11 G.移除边$((v, v'))$12 G.添加边$((v, f))$ 13 结束 14 对于每个 (v, v')，执行 $\in E_{outgoing}$ 15 G.移除边$((v, v'))$16 G.添加边$((f, v'))$ 17 结束 18 $i \leftarrow i + 1$19 结束 20 返回 G21 结束 算法1：循环移除过程的伪代码

过程 d程取代1代循环 les w以f开头ake nodes while maintai。该构造一个被替换节点的历史记录。

环，我们重复求解内层循环 $N_{c'}$ 的过程，然后继续处理 N_c 中的节点。

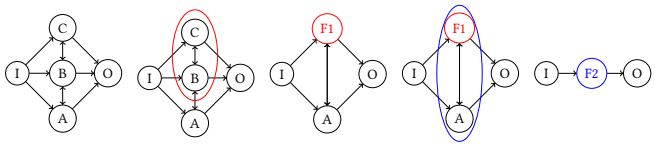


图1：算法1中所述简化过程的最小示例。基础图包含两个环：一个由节点 A 和 B组成，另一个由节点 B 和 C组成。该过程首先识别出环B，C（以红圈标出），并将其替换为红色虚拟节点F_1；该节点的输入连接来自 A 和 I，输出连接指向 A 和 O。接着，它找到环 F_1，A（以蓝圈标出），并将其替换为蓝色虚拟节点 F_2。最终得到一个有向无环图，其中包含单个虚拟隐藏节点 F_2，它隐藏了环 A，F_1；其中，F_1 是另一个虚拟节点，隐藏着节点 B 和 C。

得益于这一过程，剪枝后的网络可以具有不同的结构。我们确定了三种基础结构，具体如下。第一种情况下，输入与所有隐藏节点相连接，而这些隐藏节点又与输出相连接。这会形成一个只有单个隐藏层的神经网络，见图 4a。第二种情况下，剪枝过程会切断输入与隐藏节点之间的所有连接。因此，输入会直接连接到输出节点，从而形成一个零层神经网络，见图 4b。第三种情况下，剪枝过程会移除输入与部分隐藏节点之间的所有连接，但隐藏节点之间仍保持连接，从而形成一个具有多个层的神经网络。例如，给定隐藏节点 A、B 和 C，如果输入仅与 A 和 B 相连接，而不与 C 相连接，但 C 仍与 A 和 B 相连接，那么最终的神经网络将有两个隐藏层：第一层由神经元 A 和 B 组成，第二层仅由神经元 C 组成，见图 4c。

值得注意的是，从这三种基础结构出发，我们可以推导出更复杂的结构。例如，第三种情况中的每个节点都可以是一个虚拟节点，从而“隐藏”一组节点。

尽管这一模型原则上很有前景，但也并非没有弱点。首先，SBNN中的权重数量会随着隐藏节点数量的增加而呈二次增长，因为每个隐藏节点都与其他所有隐藏节点完全连接。此外，每条连接都对应一条ABCD规则，包含相应的4个参数。因此，与隐藏节点数量相同的前馈神经网络（FFNN）相比，需要优化的参数总数要高出几个数量级。图3比较了所提出的SBNN与另外两种基于神经网络的模型（即采用和未采用赫布学习的前馈神经网络）需要优化的参数数量。

其次，在剪枝之前，隐藏节点构成一个单一的全连接子网络，激活顺序可能会影响网络的输出。例如，假设激活顺序固定，第一个隐藏节点会将当前时间步的输入值作为输入，而对于其他隐藏节点，其接收到的输入值则是上一次激活时的值。另一方面，如上所述，剪枝之后，我们可以通过遍历底层图来确定网络的拓扑序。不过，如果存在一组彼此相连的隐藏节点，我们就无法确定唯一的激活顺序。

3.4 OpenAI 任务

为了衡量所提出的 SBNN 的性能，我们使用 OpenAI 的三种经典控制任务 [4],，即 倒立摆、山地车和月球着陆器。

在倒立摆任务中，智能体需要移动小车，使杆保持平衡状态。智能体可以向两个方向推动小车（向左/向右移动）；杆每保持平衡状态一个时间步，智能体就会获得正奖励。回合会在经过 500 个时间步后结束，或者在杆的角度超出 $\pm 12°$ 的范围时结束。

在山地车任务中，智能体需要驾驶汽车从山谷驶上山顶。智能体必须借助前方的另一座山坡积累动量，以提高速度。智能体可以执行三种动作（向左加速、向右加速或不加速）；在到达山顶之前，每个时间步都会收到负奖励。汽车到达山顶或经过 200 个时间步后，回合结束。

最后，在月球着陆器中，智能体必须让宇宙飞船着陆。智能体必须在抵消侧向风的同时降低宇宙飞船的终端速度。因此，智能体可以执行四种动作：两种用于控制侧向（左/右）发动机，一种用于启动主发动机，还有一种不执行任何动作。如果宇宙飞船在指定

区域以较低速度着陆且没有倾斜，智能体获得的奖励会增加。如果宇宙飞船着陆，或其 x 位置大于 1，则回合结束。

如果上述三个任务在 100 个回合内的平均奖励高于预定义阈值，则视为任务已解决。Cart Pole 的阈值为 475， −110Mountain Car 的阈值为 200，Lunar Lander 的阈值为 200。

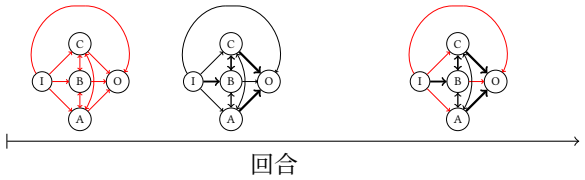


图2：SBNN 在各回合中的示意图。最初，我们将所有连接设为 0 （红色）；随后，在任务过程中，赫布可塑性会改变权重，从而形成第二个神经网络，其中不同的线条粗细表示不同的权重。在某个时刻，剪枝机制会切断最弱的连接，最终形成神经网络的结构。

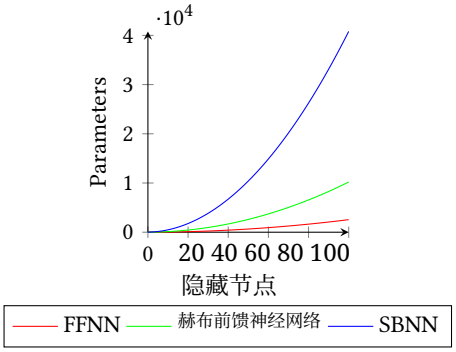
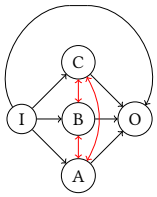


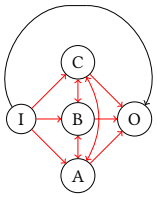
图3：需要优化的参数数量与隐藏节点数量的关系。对于（赫布）前馈神经网络，我们考虑一个具有两个隐藏层的神经网络，两个隐藏层中的神经元数量相同，如 x 轴所示。在所有情况下，均考虑单个输入和单个输出。采用赫布学习的前馈神经网络为每个连接使用不同的 ABCD规则。

4 结果

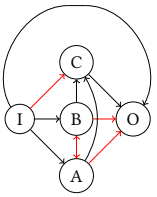
本节将分析 SBNN 与 FFNN 的性能和行为。对于 FFNN，我们在适应度评估之前应用与 SBNN 相同的剪枝机制。请注意，在我们的实现中，SBNN 和 FFNN 的隐藏节点以及输入/输出节点均使用 $tanh$ 函数作为激活函数。关于适应度评估，我们以智能体在训练期间经历的 100 个回合中的平均奖励来衡量其性能。对于 FFNN，由于剪枝过程发生在第一个回合之前，因此其适应度按定义是在剪枝后测量的。相反，对于 SBNN，剪枝过程发生在智能体的生命周期中（即跨越多个回合），因此其适应度



（a）剪枝会移除隐藏节点之间的内部连接。



（b）剪枝会移除与隐藏节点相连的所有连接。



（c）剪枝会移除 I 与 C 之间的连接，从而创建两个隐藏层。

图4：剪枝可以在 SBNN 中创建不同的结构。在图4a中，创建了一个具有一层的神经网络。在图4b中，剪枝切断了与内部节点的所有连接，使神经网络仅保留输入输出连接。最后，在图4c中，剪枝导致创建了两个隐藏层。

包含两个组成部分，一个在剪枝前，另一个在剪枝后（然后取两者的平均值）。我们将实验分为三部分，以回答三个不同的研究问题：

研究问题1 SBNN 的性能如何？影响该模型性能的主要超参数是什么？**研究问题2** SBNN 生成的网络与 FFNN 之间是否存在结构差异?

RQ3 SBNN 是否能够在不同任务之间实现泛化?

为回答这些问题，我们开展了一系列模拟实验，改变 SBNN 的三个主要超参数：隐藏节点数量 hn、剪枝率 pr和剪枝时间 pt，其中剪枝时间表示应用剪枝的时机。为计算剪枝时间，我们考虑任务中的回合数，即剪枝时间为 10 表示在智能体完成第10个回合后进行剪枝。

对于这些参数的每一种组合，我们都执行 30 次相互独立的进化过程。为了优化网络的参数（即 FFNN 的权重，或 SBNN 的 ABCD规则参数），我们采用广为人知的协方差矩阵自适应进化策略（CMA-ES） [13, 18, 19, 28]。当为月球着陆器和倒立摆生成固定数量的 2000 个个体、为山地车生成 4000 个个体后，我们停止进化。在所有情况下，我们设定 $\lambda = 4+\lfloor 3*ln(|\mathbf{p}|)\rfloor$ 和 $\mu = \frac{\lambda}{2}$，其中 $\mathbf{p}$ 是待优化的参数向量。表 1 总结了所测试的配置。请注意，我们省略了在 Cart Pole任务上获得的结果，因为 FFNN 与 SBNN 之间不存在显著差异：事实上，使用这两种模型的所有个体都完成了该任务，而该任务相较于另外两个任务更为简单。我们在 https://github.com/ndr09/SBM 公开了代码。

4.1 RQ1：性能

关于 SBNN 的性能，我们旨在评估其与 FFNN 的性能对比。由于在隐藏节点数量相同的情况下，SBNN 比 FFNN 具有更多连接，因此我们进行两项比较：一项是在隐藏节点数量相同的情况下比较两者的结果

参数	倒立摆	山地车	月球着陆器
适应度评估	2000	4000	2000
隐藏节点	3, 4	3, 4	5, 6, 7, 8, 9
剪枝时间	5、10	1、5、10	1、5、10、15、20
剪枝率	40, 60	40, 60	20, 40, 60, 80

表 1：RQ1 实验中所考虑的各项任务所使用的参数配置。对于 RQ2 和 RQ3，我们使用这些配置中的一个代表性子集。

模型，另一项是在剪枝后总连接数相同的情况下进行比较。下文将先从山地车任务开始，然后介绍月球着陆器任务的结果。

图 5 展示了山地车环境的结果。上排和下排分别对应具有 1 层、包含 3 个或 4 个隐藏节点的前馈神经网络。左列和右列分别展示剪枝率为 40% 和 60% 时的结果。在每个子图中，我们绘制了 30 次独立运行中最佳个体的平均结果。前三个箱线图分别展示剪枝时间为 10、5 和 1 时 SBNN 的结果，从左到右依次排列。最后一个箱线图展示前馈神经网络的基线结果。在山地车环境中，结果表明，与剪枝率相同的前馈神经网络相比，SBNN 的性能相近或更好。

有趣的是，我们发现剪枝时间呈现出明显的趋势：无论剪枝率和隐藏节点数量如何变化，减小 pt的值都会提升性能。因此，我们可以得出结论：在第一个回合之后，智能体已经获得了足够的经验（信息）来构建网络。

为了了解剪枝对性能的影响，我们进行了额外分析，比较剪枝前后的平均奖励。以 $pt = 10$为例，我们分别计算截至第10回合（即剪枝前）的平均奖励，以及第10回合之后（即剪枝后）的平均奖励。根据这一过程，我们发现，得益于剪枝，Mountain Car任务的性能提升了 7 – 13%。

图6展示了月球着陆器环境中的结果。对于该环境，鉴于任务的复杂度更高，我们考察了隐藏节点数量为5至9的情况。这里的结果展示方式与图5不同：具体而言，我们绘制了每次进化运行中最佳个体的奖励中位数，并在固定剪枝时间和隐藏节点数量的同时改变剪枝率。这样做突出了两点。第一，在大多数情况下，SBNN的性能等于或优于相同剪枝率下FFNN的性能。第二，随着剪枝率提高，SBNN的性能下降幅度小于FFNN基线。我们还可以观察到，在比较连接数量相近的解时，这一趋势依然成立。例如，拥有5个隐藏节点的SBNN与拥有9个隐藏节点的FFNN的连接数量相近（分别为117和108）。此外，我们可以看到与山地车中相同的趋势，即最佳结果

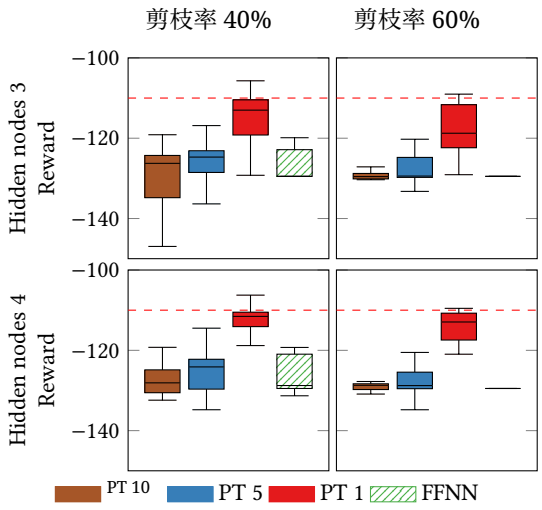


图5：山地车环境中的结果。y轴显示每次30 次运行中找到的最佳智能体在训练期间获得的平均奖励。虚线表示该环境的解决阈值。对于 SBNN，结果显示出明显趋势：随着 pt 降低，性能不断提升。

是在 $pt = 1$时取得的。因此，在这项任务中，单个回合似乎也包含了足够的信息，能够学习哪些连接可以剪枝。这些观察结果表明，至少在所测试的任务中，SBNN能够比FFNN更有效地利用网络结构（及其中的信息）。

在这种情况下，我们也采用与之前相同的分析方法，将结果分为剪枝前和剪枝后，并观察平均结果。与山地车的情况一样，在月球着陆器中，我们也能看到剪枝后性能有所提升，提升幅度在 6% 到 100% 之间。这一改进再次表明了剪枝的重要性，以及它与赫布学习的互补作用。

最后，我们根据剪枝后的连接总数比较各网络的结果。在图7中，我们比较剪枝率为 $pr = 20\%$ 的前馈神经网络与剪枝率为 $pr = 60\%$ 的 SBNN（实际上，在这些数值下，前馈神经网络和 SBNN 剪枝后的连接数量相近）。可以观察到，SBNN 的性能几乎总是与前馈神经网络相当或略好。不过，在前馈神经网络优于 SBNN 的情况下（即有 9 个节点时），差异在统计上并不显著（Wilcoxon秩和检验，$\alpha = 0.01$）。

4.2 研究问题2：SBNN 与前馈神经网络的差异

本节将分析通过 SBNN 找到的网络与前馈神经网络之间的结构差异。为此，我们根据有效连接的数量来描述这些网络，即剪枝后连接输入与输出的连接。为计算这些连接，我们会移除通向汇点或来自源节点的突触。汇点是指

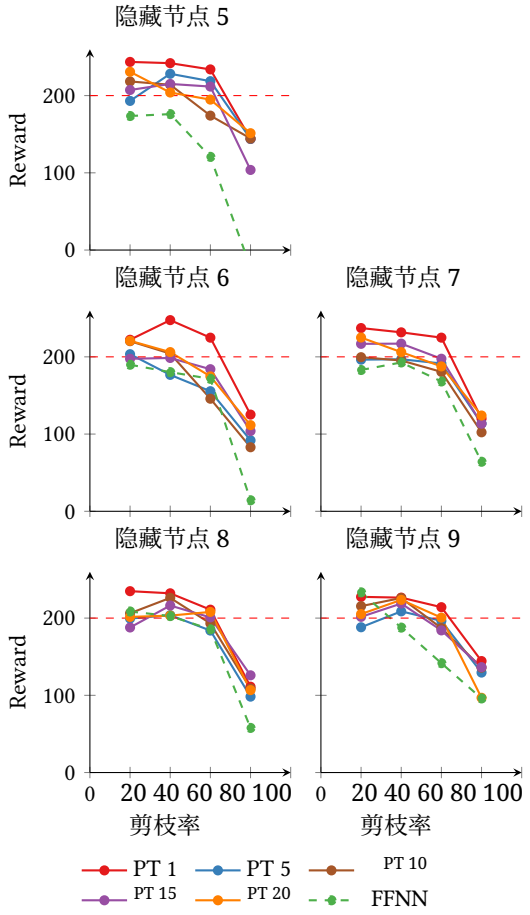


图6：在不同隐藏节点数量下，月球着陆任务的奖励中位数。x 轴表示剪枝率，不同曲线表示不同的剪枝时间。红色虚线表示求解阈值。结果表明，随着剪枝率提高，SBNN 的性能下降幅度小于前馈神经网络。

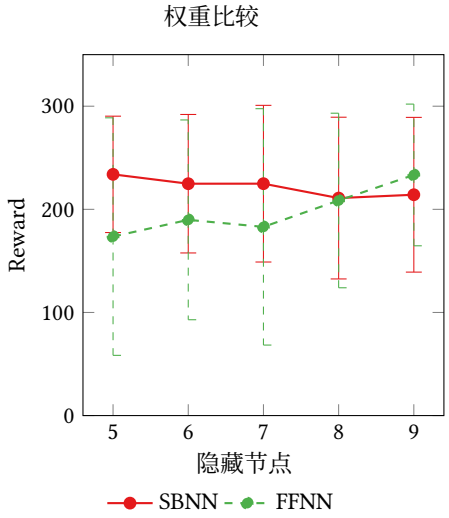


图7：剪枝率为60% 的 SBNN 与剪枝率为20%的前馈神经网络在月球着陆任务中的平均奖励。横轴表示隐藏节点的数量（请注意，在此设置下，经过剪枝过程后，两个模型具有相近的总连接数）。结果表明，SBNN 的性能与前馈神经网络相近，或略优于前馈神经网络。对于9 个隐藏节点，前馈神经网络的表现似乎略好，但差异在统计学上并不显著（Wilcoxon秩和检验，$\alpha = 0.01$）。

只有输入连接且不是输出节点的节点，而源节点是指只有输出连接且不是输入节点的节点。

图8和图9展示了上一节所述部分代表性配置的有效连接分布。对于每种配置，均考虑每次进化运行中获得的最佳解（每次运行一个）。x 轴表示剪枝后剩余有效连接占突触总数的百分比，并按每 10% 分组；y 轴表示具有相应连接数量的网络数（共 30 个，每次运行一个）。例如，若某点位于 $x = 40\%$ $y = 0.5$，表示在 50% 的运行中（即 30 次中的 15 次），剪枝后的网络有效连接数量处于 (30%，40%] 范围内。

从两幅图中，我们可以观察到一个相当清晰的规律：所有前馈神经网络配置都使用了大部分可用连接。另一方面，SBNN 的有效连接分布有两个峰值：Mountain Car任务的第一个峰值出现在 10% 到 20% 之间，月球着陆器任务的第一个峰值出现在 20% 到 40% 之间；两个任务的第二个峰值都在 90% 左右。我们对所有网络进行了目视分析，发现构成第一个峰值的 SBNN 具有如图 4b 所示的结构，其中所有隐藏节点都处于断开状态。有趣的是，当 pt 增大时，这种结构所占的比例也会增加，因为 pt 的值越高，第一个峰值就越高。这表明，剪枝发生得越晚，隐藏节点上的连接就越有可能被剪除，从而只留下输入输出连接。我们的直觉是，这种简化形式在某种程度上与解决任务所需的复杂度相关。

关于连接总数，我们观察到，在月球着陆器环境中，当剪枝率高于 20% 时，SBNN 保持着 10 到 30 个有效连接。这个范围似乎不受剪枝时间和可用隐藏节点数量的影响。相反，与山地车任务中的情况一样，FFNN 使用了大部分可用连接。因此，在这种情况下，有效连接数量很大程度上取决于隐藏节点数量和剪枝率，有效连接总数在 10 到 64 之间。有效连接数量的变化在比较 SBNN 和 FFNN 时尤其值得关注：例如，在月球着陆任务中，具有 9 个隐藏节点且剪枝率为 40% 的 FFNN 使用了全部 64 个可用连接，而 SBNN

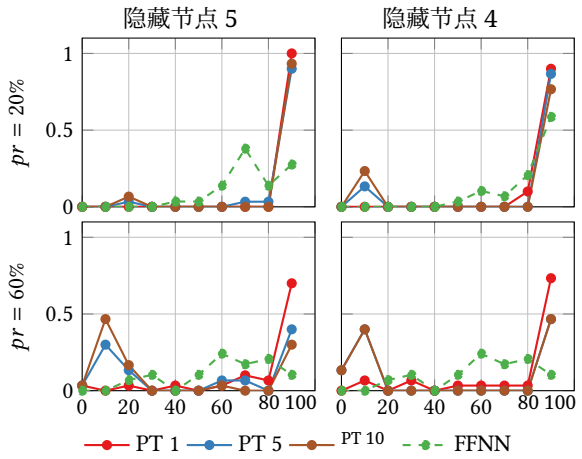


图8：在 Mountain Car 任务上对网络进行剪枝后连接数量的分布。x 轴显示有效连接的百分比（即不会指向汇节点，也不是从源节点发出的连接），每10%为一组。y 轴显示具有相应有效连接百分比的网络数量（共30个，每次运行一个）。结果表明，当剪枝率高于20%时，前馈神经网络倾向于使用所有可用连接，而 SBNN 则有两个峰值：一个约为 10 – 20%，另一个为 80 – 90%。

仅使用平均 20 个连接就解决了任务，并取得了更好的性能，见图6。

图10展示了在 Mountain Car 任务上训练的两个经过剪枝的 SBNN。这两种网络有所不同：一种移除了所有隐藏节点，仅使用两个动作（向左加速和向右加速）；另一种使用了 4 个可用隐藏节点中的 2 个，以及 3 个可用动作中的全部动作。

4.3 RQ3：泛化

在本节中，我们通过将最佳智能体（在给定任务上经过进化过程后找到的智能体）在另一个未见过的任务上进行测试，来评估 SBNN 的泛化能力。

具体而言，我们测试在月球着陆器上训练的智能体在倒立摆和山地车任务上的表现。我们将验证限制在这一情形下，因为月球着陆器环境是唯一具有足够输入节点和输出节点来执行另外两个任务的环境。事实上，不同任务具有不同的输入空间和动作空间。

为进行这项分析，我们将验证任务中的输入和输出重新映射到月球着陆器环境中的相应输入。例如，如果在月球着陆器环境中，第一个输入是 x 坐标，我们就将验证任务中对该位置的观测（即倒立摆中的小车或山地车中的汽车的 x 坐标）分别映射到第一个输入。我们将所有未使用的输入设为 0。对于输出，我们只考虑验证任务中存在的输出。例如，以倒立摆作为验证任务时，可用的动作只有向左移动 和向右移动。因此，我们只考虑控制月球着陆器左侧和右侧发动机的输出。

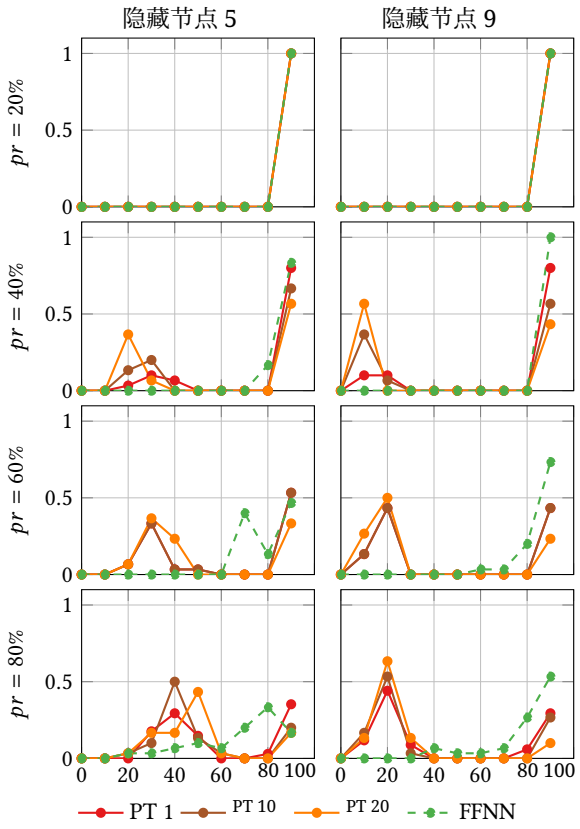


图9：在月球着陆任务中对网络进行剪枝后，连接数量的分布。横轴显示有效连接所占的百分比，这些连接不会通向汇节点，也不会从源节点出发，每10%为一组。纵轴显示具有相应有效连接百分比的网络数量（共30个，每次运行得到一个）。结果表明，当剪枝率高于20%时，前馈神经网络倾向于使用所有可用连接，而 SBNN 则有两个峰值：一个在 20 – 40%左右，另一个在 80 – 90%。

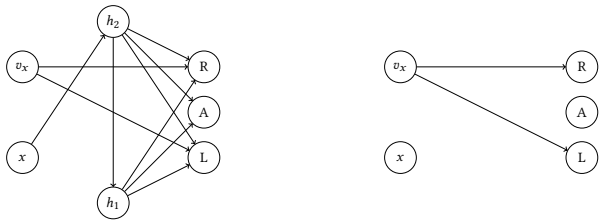


图10：在 Mountain Car任务中获得的两个精选 SBNN 的结构。左侧的 SBNN 使用2 个隐藏节点（共4个），结构为2层。右侧的 SBNN 仅使用两个连接。

图 12 展示了部分配置在验证任务中的性能（使用 Wilcoxon秩和检验评估统计显著性，$\alpha = 0.05$）。在 Cart Pole任务中，结果

表明，SBNN 的表现与前馈神经网络相近或更好。在山地车环境中，SBNN 在 $pr = 20\%$的情况下表现略逊于前馈神经网络，不过差异并非总是具有统计显著性；在 $pr = 80\%$的情况下，SBNN 表现更好，且第一四分位数的性能更高（再次表明，SBNN 模型能有效地学会利用网络结构）。

最后，通过这些验证实验，我们可以观察任务如何影响网络结构。图11展示了同一个智能体在剪枝过程后，分别解决月球着陆器和 Cart Pole任务时的网络结构。我们可以观察到，这些网络在所用连接的数量以及这些突触所连接的神经元方面存在差异。

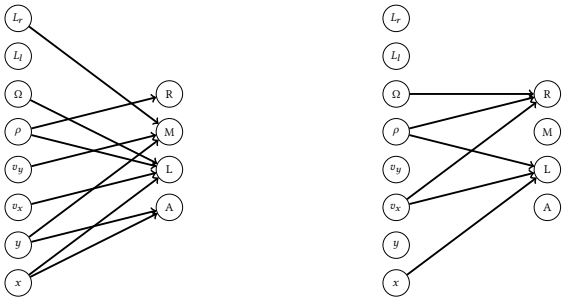


图11：同一个 SBNN 在月球着陆器（左）和倒立摆（右）环境中的结构。我们可以观察网络结构如何随任务而变化。

5 结论

在这项工作中，我们从自然大脑中发生的突触发生过程中汲取灵感，提出了一种结合可塑性和剪枝的新学习模型。在任务的第一个回合开始时，所有神经元彼此连接，并初始化为 0（即连接存在，但最初处于非激活状态）。随后，在每个回合中，基于赫布学习的可塑性模型会增强这些突触，从而激活相应的连接。最终，在智能体的生命周期中（即在预先设定的某个回合），这些连接会通过全局幅值剪枝算法进行剪枝，并停止赫布学习。我们将这一模型称为自构建神经网络（SBNN），因为它会根据智能体在任务各回合中的经验改变自身结构。

我们在 OpenAI 的三个经典控制任务上测试了模型，分别是倒立摆、山地车和月球着陆器。在实验中，我们改变了模型的三个主要参数，它们分别影响何时进行剪枝、剪枝多少以及隐藏节点的数量。我们表明，总体而言，SBNN 的性能优于前馈神经网络，并且能够更好地适应未见过的任务。此外，我们还评估了模型参数的重要性，尤其是何时剪枝以及剪枝多少。最后，我们展示了同一个智能体如何根据任务以不同方式重组其大脑，以及只要给予足够的时间，它如何从大脑中移除不必要的复杂度。

未来，我们计划根据网络中的信息流，开发一个自动决定剪枝量和剪枝时机的系统

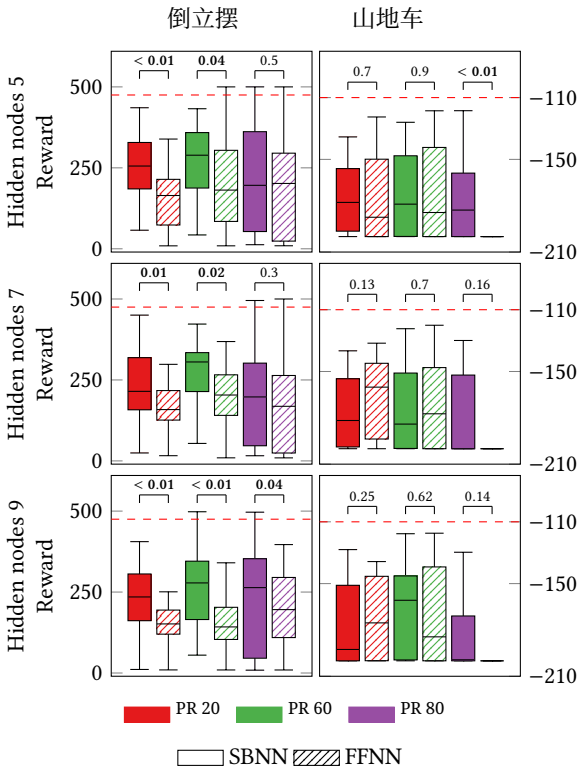


图 12：SBNN 与 $pt = 1$ 在月球着陆任务上训练的 FFNN 之间的比较。纵轴表示两个验证任务中的平均奖励：倒立摆（左）和山地车（右）。红色虚线表示两个任务的求解阈值。结果表明，在月球着陆任务上训练的 SBNN 往往能够取得优于或等同于 FFNN 的性能。对于每一组两两比较，我们注明了 Wilcoxon秩和检验的 p 值。

此外，我们还计划处理更复杂的任务，例如软体机器人的控制，在这类任务中，我们可以在更大的输入空间和动作空间上测试所提出的 SBNN。我们还计划解决第 3 节中指出的两个局限。对于需要优化的参数数量，我们将测试为每个神经元使用一个赫布规则，而不是为每个连接使用一个，同时修改更新规则，以避免每个连接都接收到相同的更新。对于激活顺序，我们计划采用基于距离的方法来解决环的问题，并将权重作为距离。

参考文献

[1] David H. Ackley、Geoffrey E. Hinton 和 Terrence J. Sejnowski。1985年。玻尔兹曼机的学习算法。*Cognitive Science* 9, 1 (1985), 147–169。https: //doi.org/10 1016 0364 0213 85 80012 4。/S - ( ) - Christopher M Bishop 等。《模式识别神经网络》。牛津大学出版社，英国牛津。[3] Jean-Pierre Bourgeois。2010。《新生儿突触大爆发》（第2版）。剑桥大学出版社，英国剑桥，71–84。[4] Greg Brockman、Vicki Cheung、Ludwig Pettersson、Jonas Schneider、John Schulman、Jie Tang 和 Wojciech Zaremba。2016。OpenAI Gym。arXiv:1606.01540。

[2] 1995

[5] Thomas H Brown、Edward W Kairiss 和 Claude L Keenan。1990。赫布突触：生物物理机制与算法。《神经科学年度评论》13, 1 (1990), 475–511。[6] Natalia Caporale 和 Yang Dan。2008。脉冲时序依赖可塑性：一条赫布学习规则。《神经科学年度评论》31 (2008), 25–46。[7] Yanqi Chen、Zhaofei Yu、Wei Fang、Tiejun Huang 和 Yonghong Tian。2021。通过梯度重连剪枝深度脉冲神经网络。arXiv:2105.04916。[8] Won-Suk Chung 和 Ben A Barres。2012。胶质细胞在突触消除中的作用。《神经生物学当前观点》22, 3 (2012), 438–445。突触结构与功能。[9] Oliver J Coleman 和 Alan D Blair。2012。为在线学习进化可塑性神经网络：综述与未来方向。载于《澳大拉西亚人工智能联合会议》。Springer，柏林、海德堡，326–337。[10] Andrea Ferigo、Giovanni Iacca、Eric Medvet 和 Federico Pigozzi。2022。在基于体素的软体机器人中进化赫布学习规则。《IEEE认知与发育系统汇刊》Early access (2022), 11。[11] Asja Fischer 和 Christian Igel。2012。受限玻尔兹曼机简介。载于《模式识别、图像分析、计算机视觉与应用进展》，Luis Alvarez、Marta Mejail、Luis Gomez 和 Julio Jacobo（编）。Springer，柏林、海德堡，14–36。[12] Dario Floreano 和 Joseba Urzelai。2000。具有在线自组织和行为适应度的进化机器人。《神经网络》13, 4-5 (2000), 431–443。[13] Matthew C Fontaine、Julian Togelius、Stefanos Nikolaidis 和 Amy K Hoover。2020。用于快速探索行为空间的协方差矩阵自适应。载于《遗传与进化计算会议》。ACM，纽约，美国纽约州，94–102。[14] Jonathan Frankle 和 Michael Carbin。2018。彩票假说：寻找稀疏且可训练的神经网络。arXiv:1803.03635。[15] Adam Gaier 和 David Ha。2019。权重无关神经网络。《神经信息处理系统进展》32 (2019), 15页。[16] Trevor Gale、Erich Elsen 和 Sara Hooker。2019。深度神经网络中的稀疏性现状。arXiv:1902.09574。[17] Song Han、Jeff Pool、John Tran 和 William J. Dally。2015。为高效神经网络同时学习权重和连接。arXiv:1506.02626。[18]Nikolaus Hansen。2016。CMA进化策略：教程。arXiv:1604.00772。[19] Nikolaus Hansen 和 Andreas Ostermeier。1996。在进化策略中自适应任意正态变异分布：协方差矩阵自适应。载于《IEEE进化计算国际会议》。IEEE，纽约，美国纽约州，312–317。[20] Donald Olding Hebb。2005。《行为的组织：一种神经心理学理论》。Psychology Press，英国伦敦。[21] Geoffrey E Hinton、Terrence J Sejnowski 等。1986。玻尔兹曼机中的学习与再学习。《并行分布式处理：认知微观结构探索》1, 282-317 (1986), 2。[22] Torsten Hoefler、Dan Alistarh、Tal Ben-Nun、Nikoli Dryden 和 Alexandra Peste。2021。深度学习中的稀疏性：通过剪枝与增长实现神经网络的高效推理和训练。《机器学习研究杂志》22, 241 (2021), 1–124。http://jmlr.org/papers/v22/21-0366.html [23] Peter R Huttenlocher 和 Arun S Dabholkar。1997。人类大脑皮层突触发生的区域差异。《比较神经学杂志》387, 2 1997 167 178 ( ), – 。[24] Yishi Jin。2005。突触发生。WormBook：《秀丽隐杆线虫生物学在线评论》。[25] Adrien Journé、Hector Garcia Rodriguez、Qinghai Guo 和 Timoleon Moraitis。2022。无反馈的赫布深度学习。arXiv:2209.11883。[26] Jeff W Lichtman 和 Howard Colman。2000。突触消除与不可磨灭的记忆。《神经元》25, 2 (2000), 269–278。[27] Shiwei Liu、Tianlong Chen、Xiaohan Chen、Zahra Atashgahi、Lu Yin、Huanyu Kou、Li Shen、Mykola Pechenizkiy、Zhangyang Wang 和 Decebal Constantin Mocanu。2021。通过增强剪枝可塑性和神经再生进行稀疏训练。《神经信息处理系统进展》34 (2021), 9908–9922。[28] Ilya Loshchilov 和 Frank Hutter。2016。用于深度神经网络超参数优化的CMA-ES。arXiv:1604.07269。[29] Claudio Mattiussi 和 Dario Floreano。2007。用于电路与网络进化的模拟遗传编码。《IEEE进化计算汇刊》11, 5 (2007), 596–607。[30] Decebal Constantin Mocanu、Elena Mocanu、Phuong H Nguyen、Madeleine Gibescu 和 Antonio Liotta。2016。受限玻尔兹曼机的拓扑学洞见。《机器学习》104 (2016), 243–270。[31] Decebal Constantin Mocanu、Elena Mocanu、Peter Stone、Phuong H Nguyen、Madeleine Gibescu 和 Antonio Liotta。2018。受网络科学启发，利用自适应稀疏连接实现人工神经网络的可扩展训练。《自然·通讯》9, 1 (2018), 2383。[32] Giorgia Nadizar、Eric Medvet、Felice Andrea Pellegrino、Marco Zullich 和 Ste- fano Nichele。2021。剪枝对软体机器人进化神经控制器的影响。载于《遗传与进化计算会议论文集附册》。ACM，纽约，美国纽约州，1744–1752。[33] Giorgia Nadizar、Eric Medvet、Hola Huse Ramstad、Stefano Nichele、Felice Andrea Pellegrino 和 Marco Zullich。2022 年。融合剪枝与神经进化：面向模块化软体机器人的稳健高效控制器。*The Knowledge Engineering Review* 37（2022），e3。[34] Elias Najarro 和 Sebastian Risi。2020 年。随机网络中的赫布可塑性元学习。《神经信息处理系统进展》33（2020），20719–20731。[35] Yael Niv、Daphna Joel、Isaac Meilijson 和 Eytan Ruppin。2001 年。不确定环境中的强化学习进化：风险规避与匹配的涌现。载于 欧洲人工生命会议。Springer，柏林、海德堡，252–261。[36] Anna R Patten、Suk Yu Yau、Christine J Fontaine、Alicia Meconi、Ryan C Wortman 和 Brian R Christie。2015 年。运动对不同疾病模型啮齿动物海马体结构和功能可塑性的益处。*Brain Plasticity* 1，1（2015），97–127。[37] Astrid G Petzoldt 和 Stephan J Sigrist。2014 年。突触发生。*Current biology* 24，22（2014），R1076–R1080。[38] Jessy Presumey、Allison R Bialas 和 Michael C Carroll。2017 年。补体系统在发育和疾病过程中的神经突触消除中的作用。*Advances in immunology* 135（2017），53–79。[39] Andrea Soltoggio、Peter Durr、Claudio Mattiussi 和 Dario Floreano。2007 年。为类强化学习问题进化神经调节拓扑结构。载于 *IEEE*进化计算大会。IEEE，纽约州纽约，美国，2471–2478。[40] Andrea Soltoggio、Kenneth O Stanley 和 Sebastian Risi。2018 年。生而学习：进化可塑性人工神经网络的灵感、进展与未来。神经网络 108（2018），48–67。[41] Gregory Z. Tau 和 Bradley S. Peterson。2010 年。大脑回路的正常发育。*Neuropsychopharmacology* 35（2010），147–168。[42] Anil Yaman、Giovanni Iacca、Decebal Constantin Mocanu、Matt Coler、George Fletcher 和 Mykola Pechenizkiy。2021 年。进化可塑性，以实现变化环境条件下的自主学习。*Evolutionarycomputation* 29，3（2021），391–414。[43] Karl Zilles。1992 年。作为中枢神经系统适应性特征的神经元可塑性。*Annals of Anatomy-Anatomischer Anzeiger* 174，5（1992），383–391。[44] Wendy Zukerman 和 Andrew Purcell。2011 年。大脑的突触修剪会持续到你二十多岁。*New Scientist* 211，2826（2011），9。