蒙特利尔大学

演化人工神经网络，以模仿《Shinobi III：忍者大师归来》中的人类行为

par

Maximilien Le Cleï

计算机科学与运筹学系，文理学院

为获得计算机科学理学硕士（M.Sc.）学位而提交的论文，人工智能方向

2021年8月

© Maximilien Le Cleï，2021

蒙特利尔大学

系　　计算机科学与运筹学系，文理学院　　S

这篇题为

演化人工神经网络以模仿《Shinobi III：Return of the Ninja Master》中的人类行为

由以下人员介绍

Maximilien Le Cleï

由以下人员组成的评审委员会进行了评审

Ioannis Mitliagkas主席兼评审员

Pierre-Louis Bellec研究主任

Devon Hjelm

联合导师

SarathChandar Anbil Parthipan评审委员会成员

摘要

我们的社会越来越依赖计算机工具。过去十年间，这一现象尤为显著，其中一个原因是人工智能新范式的兴起。具体而言，深度神经网络与随机梯度下降这两种算法技术相结合，在不断呈指数级增长的计算能力推动下，已成为并继续成为众多新兴技术的重要优势。然而，随着进步持续发展，仍有人在思考，其他方法是否也能同样受益于这些硬件进步，甚至获益更多。

为了推进这项研究，本论文探讨了进化算法及其在动态神经网络中的应用。这两种技术具有许多优势，却尚未在当代人工智能领域得到充分应用。我们发现，通过开发新方法并利用强大的计算能力，我们得以在基准上开发高性能智能体，也能开发出在电子游戏 *Shinobi III*：忍者大师归来中行为与人类受试者十分相似的其他智能体。该游戏是复杂任务的典型例子，迄今为止，只有梯度优化才能应对此类任务。

关键词

人工智能、进化算法、人工神经网络

摘要

我们的社会日益青睐计算工具。在过去十年中，受多种因素影响，这一现象愈发显著，其中包括一种全新人工智能范式的出现。具体而言，深度神经网络与随机梯度下降这两种算法技术相结合，并在计算能力呈指数级增长的推动下，已经并将继续成为众多现代技术的重要优势。然而，随着进步不断推进，仍有人在思考，其他方法能否同样受益于这些硬件进步，甚至从中获得更大的收益。

为了推进这项研究，我们在本文中深入探讨了进化算法及其在动态神经网络中的应用。这两种技术尽管具备诸多优势，却尚未在当代人工智能领域找到适合自己的位置。我们发现，通过开发新方法并利用强大的计算资源，不仅能够在各种基准测试中打造表现出色的智能体，还能让一些其他智能体在电子游戏 *Shinobi III*：忍者大师归来 上表现得与人类受试者极为相似。这类任务通常十分复杂，过去超出了非基于梯度的优化方法的能力范围。

关键词
人工智能、进化算法、人工神经网络

目录

摘要 3

摘要 4

表格目录 6

图目录 es 7

首字母缩略词与缩写词表 8

背景 9

1. 人工神经网络 1.1. 前馈神经网络 1.2. 循环神经网络 1.3. 卷积神经网络 9 9 10 10

2. 进化算法 2.1. 在人工神经网络中的应用 12 13

3. 优化 3.1. 度量优化 3.2. 生成式对抗优化 13 13 14

进化人工神经网络以模仿 Shinobi III：忍者大师归来中的人类行为 15

1. 引言 2. 背景 3. 方法 3.1. 进化对抗生成 3.2. 动态神经网络 3.2.1. 动态循环网络 3.2.1.1. 变异 #1：增长连接 3.2.1.2. 变异 #2：剪枝连接 3.2.1.3. 变异 #3：增长节点 3.2.1.4. 变异#4：剪枝节点 3.2.2. 动态卷积网络 3.2.2.1. 新增节点操作 3.2.2.2. 变异#1：生长分支 3.2.2.3. 3.2.2.3. 变异#2：剪枝分支 3.2.2.4. 变异#3：扩展节点 3.2.2.5. 变异#4：收缩节点 3.2.3. 连接动态网络 3.3. 进化算法的可扩展通信协议 4. 实验 4.1. 分数优化与深度强化学习智能体模仿 4.1.1. CartPole 和 LunarLander 4.1.2. Pong 4.2. Shinobi III：忍者大师归来中的人类行为模仿 5. 讨论 6. 结论 15 15 16 16 16 18 18 19 19 20 20 22 23 23 24 25 25 27 28 30 30 31 32 33 37 37

参考文献 38

表目录

表1. **Shinobi III 中可能的动作及动作组合的视频样本。35***表2.* **Shinobi III 中随机智能体和分数优化智能体的视频样本。35***表3.* **Shinobi III 中受试者行为的视频样本。35***表4.* **Shinobi III 中受试者行为的定性评估表。35***表5.* **Shinobi III 中智能体行为的定性评估表。36***表6.* **Shinobi III 模仿结果表。36**

图目录

图*1.* **人工神经元。**图*2.* **双层前馈神经网络。**图*3.* **三层前馈神经网络。**图*4.* **循环神经网络。** 9
图*5.* **卷积操作示例。**图*6.* **池化操作示例。**图*7.* **卷积神经网络示例。**图*8.* **进化算法示例。**图 9
9. **神经进化示例。**图*10.* **基于指标和对抗的数据模仿优化。**图*11.* **进化对抗生成：智能体匹配。**图*12.* **进化对抗生成：智能体评估。**图*13.* **动态循环网络节点。**图*14.* **DRN变异#1：增长连接。** 10
图*15.* **DRN变异#2：剪枝连接。**图*16.* **DRN变异#3：增长节点。**图 *17.* **DRN变异#4：剪枝节点。** 10
图*18.* **动态卷积网络节点。**图*20.* **DCN操作：增长节点。**图 *21.* **DCN变异#1：生长分支。** 11
图 *22.* **DCN变异#2：剪枝分支。**图 *23.* **DCN 变异 #3：扩展节点。**图*24.* **DCN 变异 #4：收缩节点。** 11
图*24.* **连接动态网络：生长分支变异。**图*25.* **连接动态网络：剪枝分支变异。**图*26.* **进化式进程间通信：分散/收集。**图 *27.* **进化式进程间通信：分散/收集 + 点对点。** 12
图*28.* **CartPole上的分数最大化和生成对抗模仿。**图*29.* **LunarLander上的分数最大化和生成对抗模仿。**图*30.* **Pong上的分数最大化和生成对抗模仿。**图*31.* **世嘉Genesis控制器示意图。** 12
13
14
17
17
18
19
20
21
21
22
23
24
25
26
26
27
28
29
30
31
32
33
34

首字母缩略词和缩写列表

DCN 动态卷积网络**DRN** 动态循环网络**遗传算法** 遗传算法**NEAT** 增广拓扑神经进化**强化学习** 强化学习

背景

1. 人工神经网络

人工神经网络（McCulloch & Pitts, 1943；Farley &Clark, 1954）是一种计算技术，灵感源自生物神经系统。与其生物对应系统一样，这些人工系统由相互连接的神经元组成，它们接收、处理并传输信息信号（图1）。

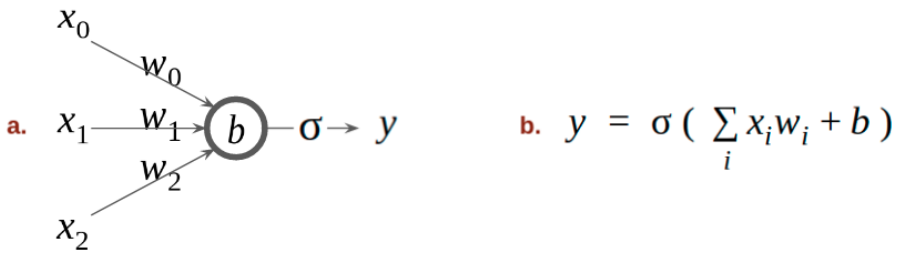

图*1.* **人工神经元。**(a) 该神经元接收输入值（x_0, x_1, x_2），并分别与权重（w_0, w_1, w_2）相乘，然后将乘积与偏置相加，**b** 得到一个值，再经非线性函数 **σ(x)** 变换后，最终输出**y**。例如，输入值分别为（**1**、**2**、**3**），权重分别为（**-1**、**1**、**0**），偏置为**-0.5** ，且 **σ(x) =max(x, 0)**，该神经元输出**y = 0.5**。(b) 广义数学公式。

1.1. 前馈神经网络

前馈神经网络（Rosenblatt，1958年；Widrow & Hoff，1960年）是最常见的人工神经网络类型之一。这些网络采用分层有向结构，以无环方式处理标量信息。第一层由所谓的**输入节点**组成，用于接收来自外部来源的信息信号，并仅将其传递到网络的下一层节点，不进行转换。最后一层由**输出节点**组成，用于接收、转换信息并将其传输出网络。此类网络最简单的实例是双层前馈神经网络，其中输出节点直接转换从输入节点接收的信息（图2）。

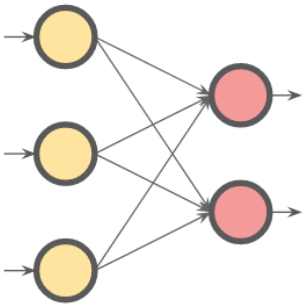

图*2*。 **双层前馈神经网络。** 输入节点 各自传递一个输入系统的值。 输出节点 接收并转换每个输入节点提供的信息，并将其输出到系统之外。

为了对更复杂的函数建模，应在**隐藏节点**的额外层置于**输入**与**输出**节点（图3）之间。这些节点可用作过渡计算单元，以便向网络更深层的节点发送更复杂的信息信号。

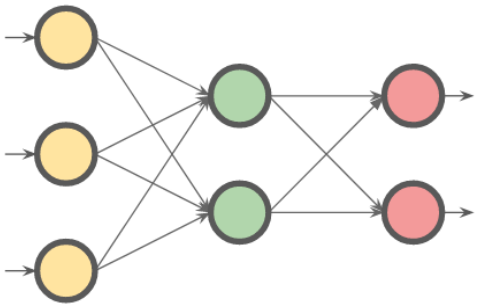

图*3*。 **三层前馈神经网络。** 在输入和输出节点之间可以插入隐藏节点，以使网络的计算图更加复杂，从而使其能够对更复杂的函数进行建模。

1.2. 循环神经网络

由于计算图完全沿前向流动，前馈神经网络不会保留其计算出的任何信息。循环神经网络（Hopfield，1982年；Rumelhart、Hinton 和 Williams，1985年）是对这类网络的扩展，其中某些节点或节点层会在计算图中向后连接，通常连接到自身，使先前计算出的部分信息能够反馈到系统中，供后续网络传递使用（图4）。

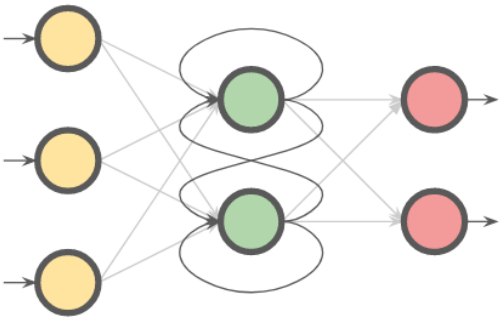

图*4*。 **循环神经网络。** 一种简单的循环神经网络，其中隐藏节点连接回自身，这意味着隐藏节点在时间**t** 计算出的信息会在时间**t+1**反馈给这些节点。

1.3. 卷积神经网络

迄今讨论过的不同类型人工神经网络都以标量值作为输入并处理标量值。然而，图像数据通常以 RGB[1] 像素值网格的形式组织，若不牺牲其结构信息，就无法将其恰当地转换为这种排列方式。

[1] 红绿蓝

受动物视觉研究的启发，福岛邦彦于1979年首次提出了“新认知机”，这是一种采用两种新型运算——卷积（图5）和池化（图6）运算的人工神经网络。这类网络如今通常被称为卷积神经网络（图7），与反向传播算法（LeCun等人，1989年）相结合后，已成为处理图像数据的标准神经网络技术。

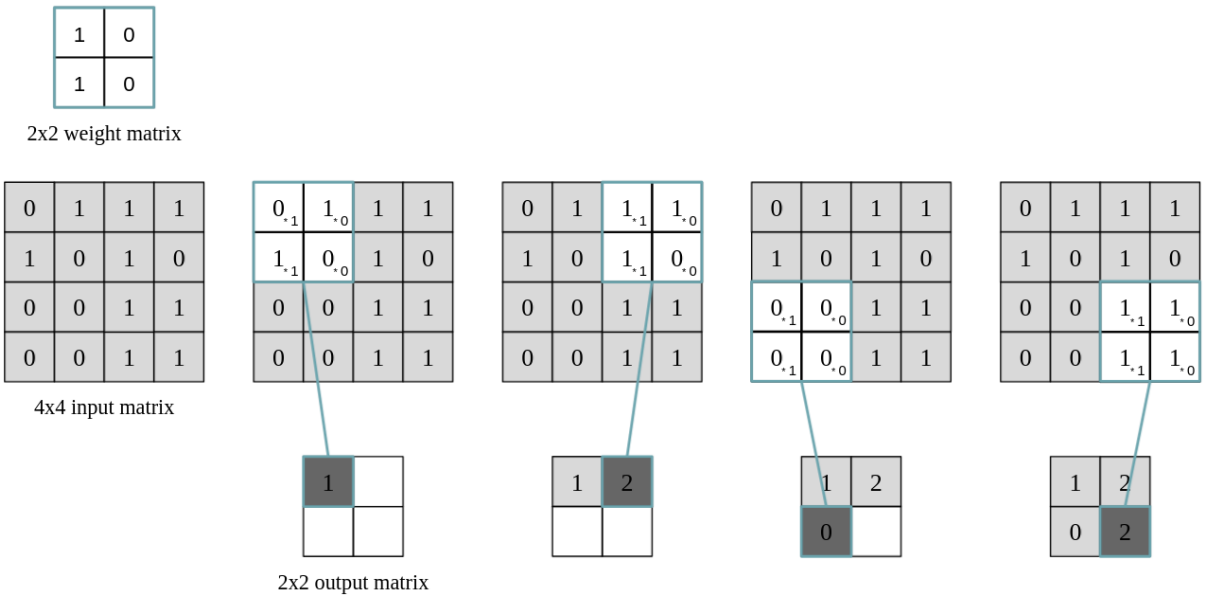

图 *5*。 **卷积操作示例。** 一个 2x2 权重核在输入图像上进行卷积，将对应值相乘后求和，从而生成相应的输出矩阵。

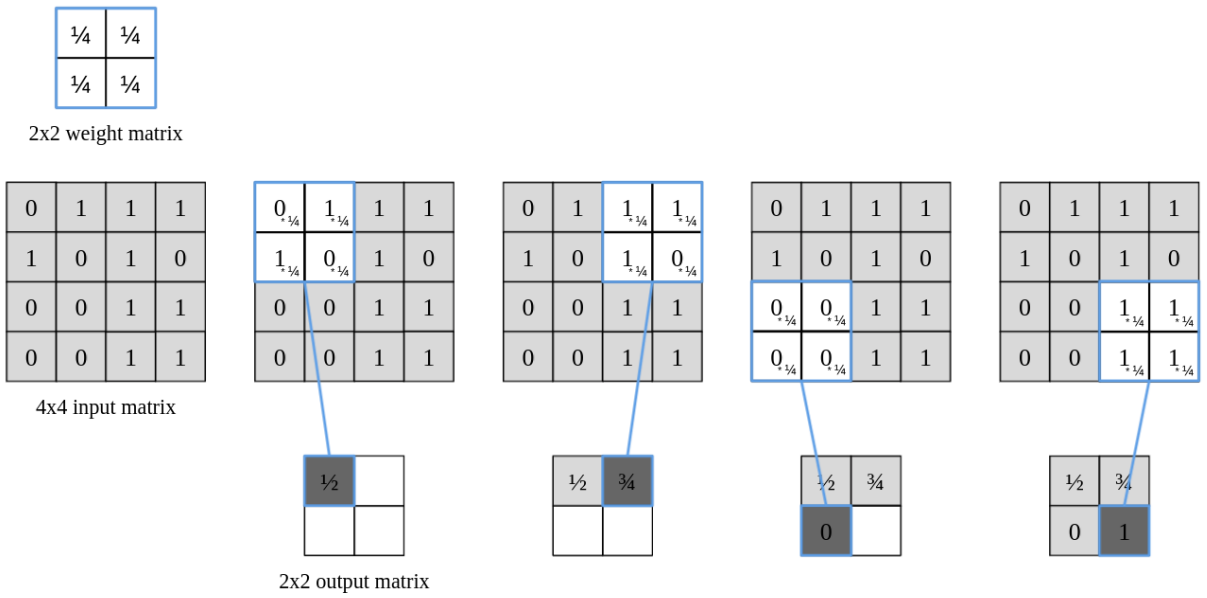

图 *6*。 **池化操作示例。** （平均）池化操作可以视为卷积操作的一种特殊情况，其中权重核的权重相等且总和为 1。

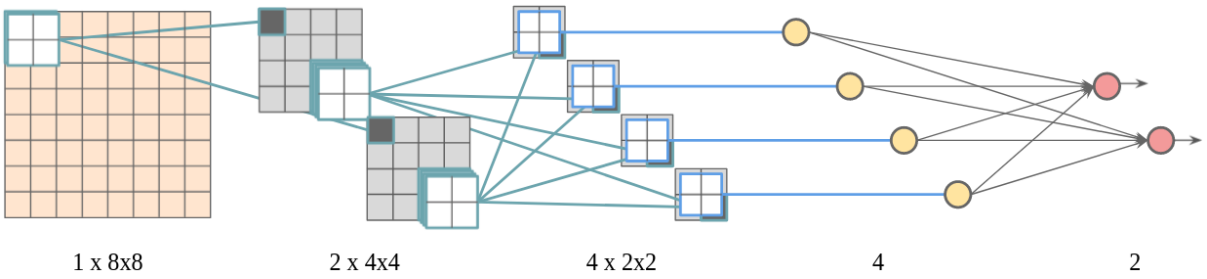

图7。**卷积神经网络示例。**该神经网络以一张单通道 8×8 图像作为输入。两个权重滤波器在其上进行卷积，生成两张 4×4 特征图。随后，四个新的权重核分别在这两张特征图上进行卷积，生成四张 2×2 特征图。接着应用 2×2 平均池化操作，得到四个标量值，并将其输入四个传统神经元。最后，通过两个输出节点，使用图1中所述的标准人工神经元运算对这四个标量值进行转换。

2. 进化算法

进化算法（Barricelli，1954年；Fraser，1957年；Fogel、Owens 和 Walsh，1966年；Rechemberg，1973年；Schwefel，1974年；Holland，1975年）是一种受自然选择这一生物过程启发的计算技术。这些算法有多种形式；不过，通常而言，由 P 个N个人工智能体组成的种群会循环经历三个阶段：**变异**、**评估**和**选择**。在**变异**阶段，种群中的智能体会通过两种操作发生改变：**交叉**，即组合现有智能体以形成新的智能体；以及**突变**，即对现有智能体进行随机扰动。随后，在**评估**阶段，会根据智能体在某项预先设定任务中的**性能**为其赋予适应度分数。最后，在**选择**阶段，会根据适应度分数选出一组智能体，组成种群的下一世代。

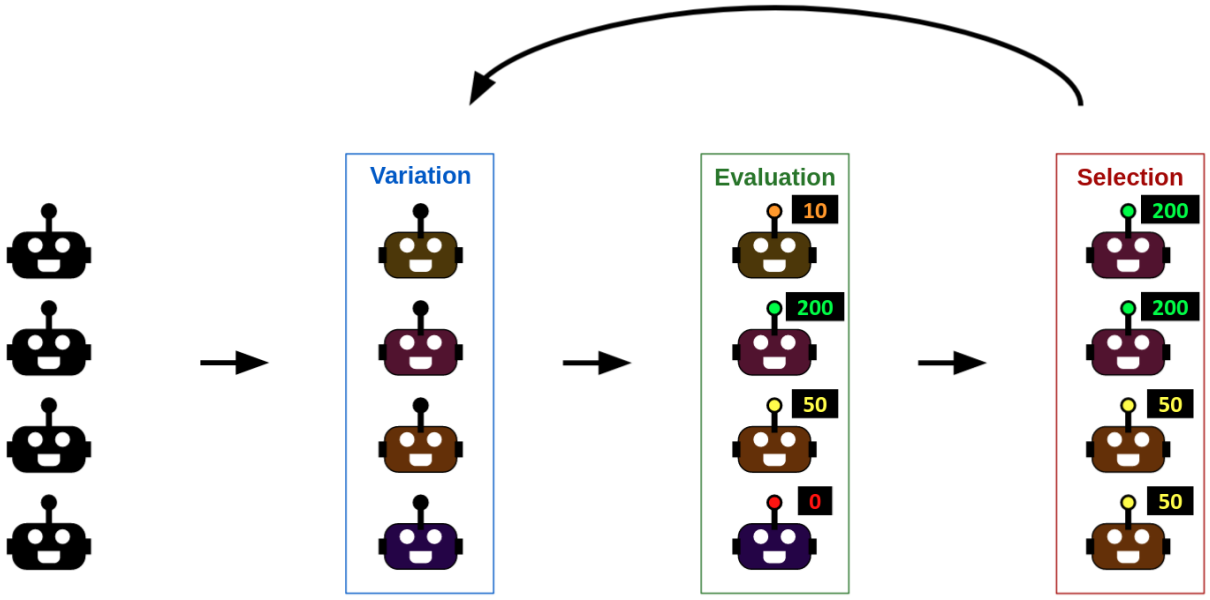

图8。**进化算法示例。**种群由四个完全相同的智能体初始化（最左侧）。首先，在变异阶段，这四个智能体会随机发生变异；随后，在评估阶段，它们会根据给定任务获得适应度得分；最后进入选择阶段，保留种群中得分最高的一半，并将其复制到得分较低的一半所占的位置。该过程应循环重复，除非设置了终止准则，否则将无限进行。

2.1. 在人工神经网络中的应用

由于定义较为宽泛，进化算法适用于各种不同场景。其中一种常见应用是人工神经网络（Montana 和 Davis，1989年；Miller、Todd 和 Hedge，1989年；Fogel、Fogel 和 Porto，1990年；Belew、McInerney 和Schraudolph，1990年；Whitley、Starkweather 和 Bogart，1990年），通常简称为神经进化。构成人工智能体的神经网络在变异阶段也应以类似方式受到扰动并进行重组。这些操作可以作用于从连接强度到网络架构，乃至元层级参数等各种对象。

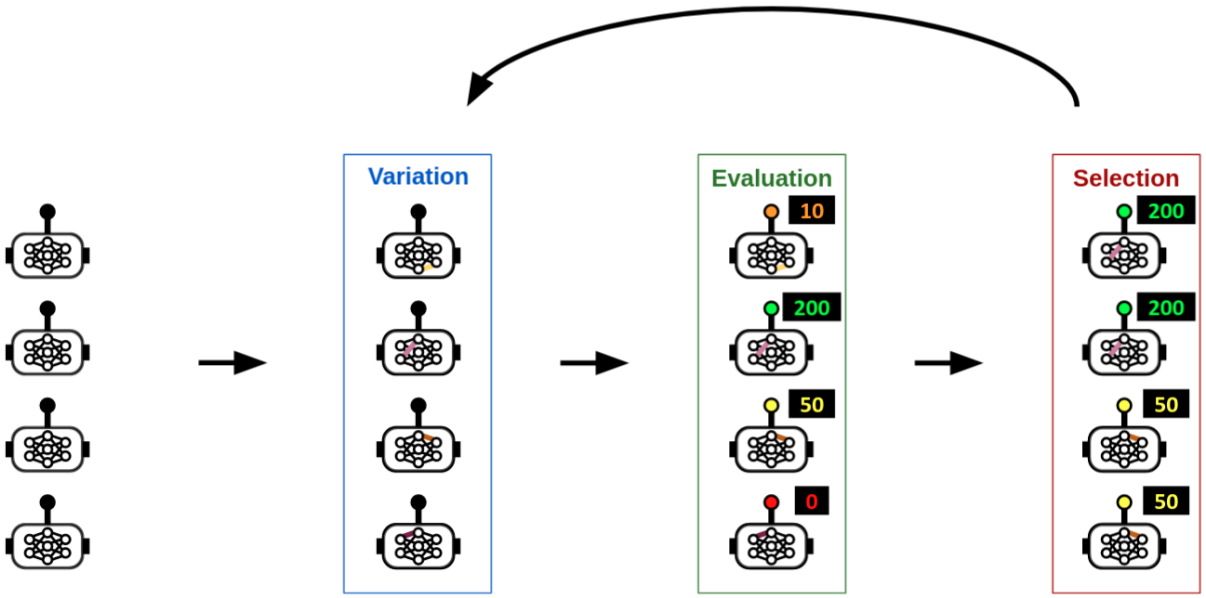

图 *9.* **神经进化示例。** 我们遵循 图 8 中介绍的相同原则，但加入了人工智能体现在配备了人工神经网络这一点。在此示例中，智能体首先使用相同的前馈神经网络进行初始化；在变异阶段，这些网络中的一个连接会受到随机扰动。随后，对所有这些略有差异的智能体进行评估和选择，如前文所示。

3. 优化

3.1. 度量优化

从历史上看，构建能够产生有用行为的“学习”系统[2] 最常见的方法，是设定一个定量性能标准，作为预期定性行为的替代指标，然后使用算法根据该指标优化系统。例如，模式识别系统通常以分类率、决定系数甚至困惑度进行评估和优化；而控制系统和游戏系统通常以得分和/或完成度为优化目标。

然而，许多理想的定性行为，例如模仿，一直难以通过定量指标恰当地衡量。

[2] 针对特定任务进行设置、能够自动且逐步优化（而非预编程）的系统。

3.2. 生成式对抗优化

近期在神经网络梯度优化领域广受欢迎的生成对抗优化框架，提出了一种定量衡量模仿效果的初步变通方案（Goodfellow 等人，2014）。从本质上说，这一框架不再手动量化性能，而是引入第二个智能体，使其通过优化来区分真实的可采集数据与人工生成的数据。与之相反，原先负责生成数据的智能体则经过优化，以生成令新智能体难以辨别的数据。当优化条件适宜时，生成智能体最终会产生与真实数据极为相似的合成数据，以至于判别器智能体再也无法分辨两者。

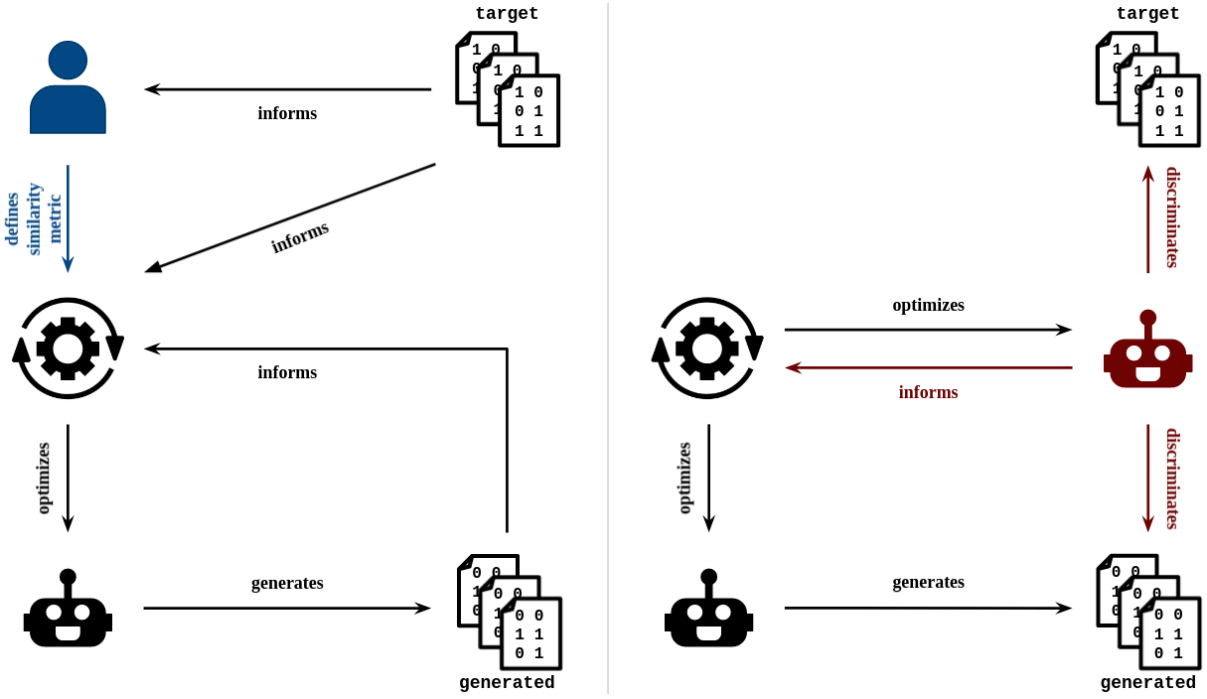

图*1*基于指标和对抗的数据模仿优化。（左）根据某个目标数据集手动定义相似度指标，供待优化的智能体作为优化依据。（右）不使用指标，而是引入第二个智能体并对其进行优化，使其能够区分生成数据和目标数据。随后优化生成智能体，以降低新智能体成功区分数据的概率。

演化人工神经网络以模仿《Shinobi III：Returnof the Ninja Master》中的人类行为

Maximilien Le Cleï [12]、François Paugam [12]、Olexa Bilaniuk [2]、Julie Boyle [1]、Yann Harel [1]、Paul-Henri Mignot [1]、Basile Pinsard [1]、Loïc Tetrel [1]、Devon Hjelm [2], Pierre Bellec [1]

[1] CRIUGM，蒙特利尔[2]
Mila，蒙特利尔大学

贡献声明：*Pierre Bellec* 和 *Devon Hjelm* 负责指导这项工作。*François Paugam* 除参与许多有价值的讨论外，还协助开发了软件库。*Loïc Tetrel* 和 *Olexa Bilaniuk* 提供了各种技术见解，后者提出的建议最终促成了新的进程间通信协议。*Paul-Henri Mignot* 和 *Basile Pinsard* 开发了受试者数据采集平台。*Julie Boyle* 协调了受试者行为数据采集。*Yann Harel* 开发了本工作所使用的受试者数据验证工具。我，*MaximilienLe Cleï*，开发了相关方法和实验。

我们尝试进一步展示进化算法利用现代计算资源的能力。通过 a) 运用对抗生成框架，b) 开发动态复杂度人工神经网络，以及 c) 探索更好地利用现代计算系统的方法，我们得以在电子游戏环境中，在较短的时间窗口内生成行为与人类受试者几乎完全相同的人工智能体。

1. 引言

基于梯度的深度神经网络优化（LeCun、Bengio 和 Hinton，2015年；Schmidhuber，2015年），凭借其利用现代计算资源的能力，已成为本十年来人工智能领域影响力最大的单一研究方向。然而，鉴于计算成本正呈指数级下降，其他具有类似强大扩展能力的技术似乎也很可能最终受到关注。越来越多的证据表明，进化计算可能就是其中之一。事实上，在计算资源充足的情况下，简单的进化算法与深度网络相结合，已开始与流行的现代方法相媲美（Salimans 等，2017；Such 等，2017；Risi 和 Stanley，2019年）。

在这项工作中，我们通过三项贡献进一步推进这一研究。首先，我们提出了一种专为进化算法设计的对抗生成框架改编方案，该框架最近由 Goodfellow 等人，2014 推广。接下来，我们重新审视神经网络的动态复杂化，这一组成部分是该领域此前取得成功的关键（Stanley 和 Miikkulainen，2002年）。最后，我们提出了一种高度可扩展的进程间通信协议，旨在帮助进化算法更好地利用现代计算系统。

我们发现，只要获得充足的资源，这些技术便能在短时间内，于电子游戏环境中生成几乎与人类受试者的行为无法区分的人工行为。

2. 背景

长期以来，人工智能研究人员一直利用电子游戏来评估其系统的能力。迄今为止，该领域最引人注目的研究大多出自受心理学启发的强化学习领域（Sutton &Barto，2018年）。这一学科最初在信息量较低、相对简单的游戏中取得成功，随后又与深度神经网络（Mnih 等人，2013）成功结合，使其能够应对需要更复杂信息处理的高维游戏。此后，深度强化学习取得了迄今最令人瞩目的一些现代人工智能成果，其智能体能够击败世界顶尖的围棋（Silver等人，2016）、Dota 2（Berner 等人，2019年）和 Starcraft 2（Vinyals 等人，2019年）玩家。

神经进化的故事（Stanley 等人，2019年）——这一名称源于将进化算法应用于人工神经网络——同样深深植根于电子游戏领域。时至今日，它或许最为人熟知的是 NEAT（Stanley & Miikkulainen，2002年），这是一种成功结合多种遗传算子、从而构建出日益复杂的神经网络的算法，能够应对维度相对较低的游戏。类似地，最近人们发现，只要拥有足够的计算资源，即使是非常简单的进化算法也能优化深度神经网络，使其进行复杂游戏（Salimans 等人，2017；Such 等人，2017；Risi & Stanley，2019年）。尽管该领域尚未产生能够击败世界冠军的智能体，但已有研究表明，进化组件在其中一个最先进的系统中发挥了关键作用（Arulkumaran、Cully 与 Togelius，2019年）。

模仿学习（Hussein 等人，2017）是创造击败世界冠军的系统所必不可少的另一项组成部分。此类研究中的技术被广泛用于帮助许多人工智能体通过学习人类行为来习得技能，并加快这一过程。该框架目前有多种形式，其中一种是将其与生成对抗网络（Goodfellow 等人，2014年）相结合（。神经进化已被广泛探索，并应用于对抗框架（Stanley 和 Miikkulainen，2004年；Costa等人，2020年）和人类行为模仿（Ki、Lyu 和 Oh，2006；Miranda、Sanchez-Ruiz 和Peinado，2016；Jalali 等人，2019年），但迄今为止，将两者结合起来的研究似乎仍相对有限。

3. 方法

3.1. 进化对抗生成

进化算法是经典的基于种群的优化算法，灵感源自生物界的自然选择过程。这些技术通常围绕一个迭代的三阶段循环来构建智能体种群。智能体首先经历“变异”阶段，在此阶段通过遗传算子对其进行改变。随后进入“评估”阶段，对智能体完成当前任务的能力进行评估。最后是“选择”阶段，此时决定哪些智能体将构成下一世代种群的基础。

定义这些阶段的方式有很多种；然而，评估智能体主要依赖于定义一种定量指标，作为定性目标的代理，这种指标称为适应度函数。然而，许多任务本质上很难进行恰当的量化（而另一些看似不难量化的任务，则容易出现意料之外的行为，Amodei & Clark，2016）。为绕过这一问题，我们提出构建一个对抗生成框架，让两个规模相同的智能体种群相互对抗。一个种群由生成数据点的智能体组成，称为生成器；另一个种群则包含负责区分生成数据点与从目标数据集中采样的数据点的智能体，称为判别器。每次迭代中，为准备评估阶段，会将每个判别器随机分配给一个不同的生成器和一个随机的"真实"数据点（图11）。在每场对抗中，生成器生成一个数据点后，根据判别器正确识别生成数据和真实数据的能力对其进行评估；与此同时，根据与生成器配对的判别器对生成数据的误判情况，对生成器进行评估（图12）。最后，选出每个种群中排名前50%的智能体，并将其复制到各自种群的后半部分。

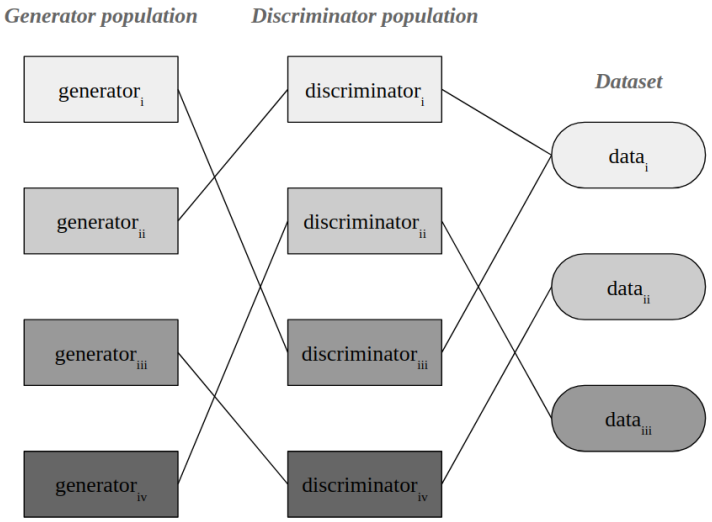

图*11.* **进化对抗生成：智能体匹配。** 在进行评估之前，会为每个判别器随机分配一个独特的生成器和一个随机数据点。

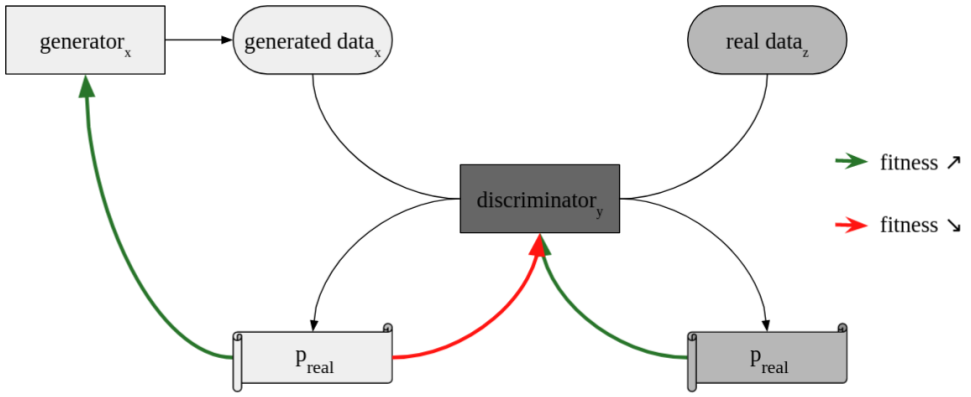

图*12.* **进化对抗生成：智能体评估。** 生成器生成数据点后，判别器会观察真实数据点和生成数据点，并分别输出一个得分，表示其对数据真实性的置信度。生成器的适应度等于判别器对生成数据真实性的（错误）置信度；判别器的适应度则等于其对真实数据真实性的（正确）置信度减去对生成数据真实性的（错误）置信度。需要注意的是，该框架不限制数据类型。

3.2. 动态神经网络

人工神经网络是一种受生物学启发的计算系统，已成为人工智能近期成就的核心。经过多轮创新，这些网络变得规模更大、效率更高，并且在图像分类（Tan 与 Le，2019年）、图像生成（Karras、Laine 与 Aila，2019年）和自然语言处理（Brown 等，2020年）等任务上表现得越来越出色。然而，由于随机梯度下降这一主要优化算法的特性，复杂度的提升一直需要人工实现。相比之下，神经进化能够同时修改权重和架构，正如 NEAT（Stanley 与 Miikkulainen，2002年）所成功展示的那样，为提升复杂度提供了一条更自然的途径。然而，过去十年间，高维数据不断涌现，而动态复杂度神经网络在处理这类数据时尚未取得同样出色的表现。

本节将通过利用卷积操作这一大多数现代人工智能系统中的关键组件，重新探讨动态网络这一主题；这些系统能够处理图像和视频数据。我们首先介绍动态循环网络（DRN），随后介绍动态卷积网络（DCN），最后在第三个小节中说明二者如何结合。

3.2.1. 动态循环网络

我们首先介绍动态循环网络（DRN），这是一种由有向分层图构成的网络，其中节点（图13）通过线性变换处理信息，然后再应用非线性激活函数。该网络通过四种结构变异不断演化，复杂度也随之提升：

1）增长连接（第3.2.1.1节）2）剪枝连接（第3.2.1.2节）3）增长节点（第3.2.1.3节）4）剪枝节点（第3.2.1.4节）

为了描述这四种突变，我们定义如下：**输入节点：** *(≠输入节点)* 某个节点接收信息的节点集合。**输出节点：** *(≠输出节点)* 某个节点向其发送信息的节点集合。**接收节点：** 由所有输入节点 + 拥有输入节点的节点组成的集合。**发射节点：** 拥有输出节点的节点列表。每个输出节点对应的节点仅出现一次。

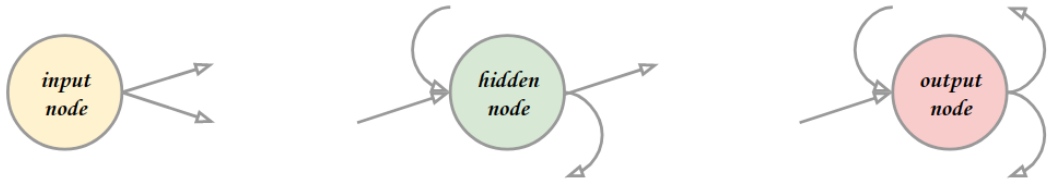

图*13.* **动态循环网络节点。** 在第一层，输入节点将输入信息传递给隐藏节点和输出节点。在最后一层，输出节点从任意潜在节点接收信息，并向其发射信息。在中间层，隐藏节点与输出节点一样，也从任意潜在节点接收信息，并向其发射信息。对于节点之间的前向连接，接收端节点会在同一次网络传递中获取并处理发射节点的输出信息；而对于反向/同层连接，接收端节点则会在下一次传递中获取发射节点的输出信息。

3.2.1.1. 变异 #1：增长连接

1. 从所有**接收节点**组成的集合中抽取第一个节点 [3]。***2.*** 从所有**隐藏节点** 和**输出节点**组成的集合中抽取第二个节点。***3.*** 在抽取的第一个节点与第二个节点之间建立一条新的加权 [4] 连接。

我们在图14中给出了这一变异的示例。

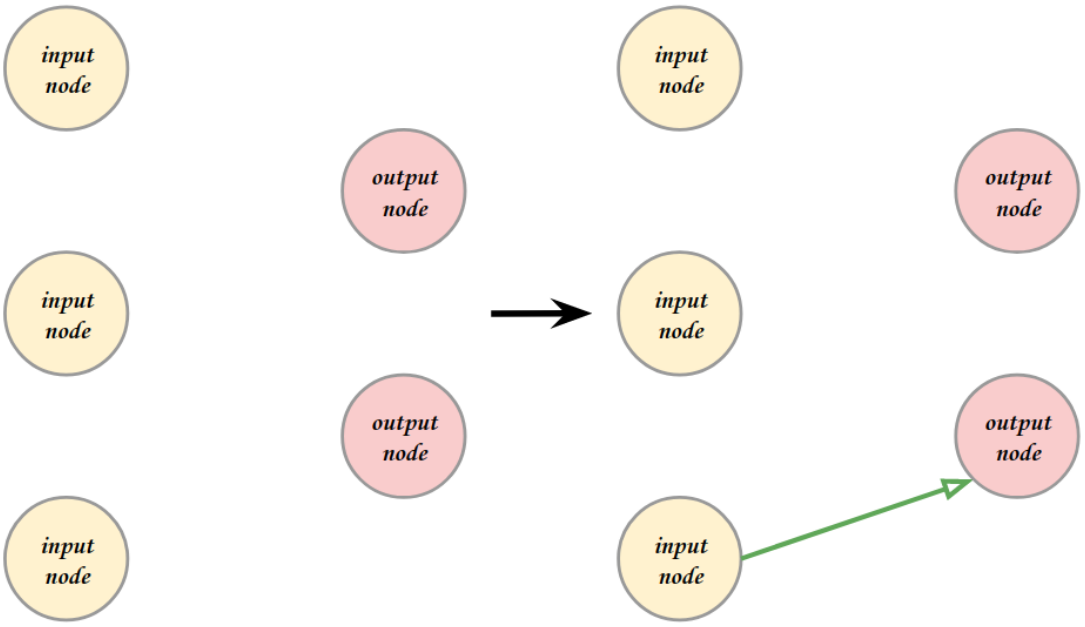

图*14*。 **DRN变异 #1：增长连接。** 为了有效展示四种 DRN 变异，我们将在接下来的四幅图中详细说明一个假设场景：我们希望神经网络处理三个标量输入并输出两个标量。在该场景中，DRN 初始化时包含三个输入节点和两个输出节点，且均不带有连接。对这样的 DRN 调用“增长连接”变异，需要从三个输入节点中采样第一个节点，再从两个输出节点中采样第二个节点。假设采样结果分别为第三个输入节点和第二个输出节点，那么这两个节点之间便会创建一条带权连接。

3.2.1.2. 变异 #2：剪枝连接

1. 首先从**发射节点**列表中采样一个节点。***2.*** 然后从第一个节点的**输出节点**集合中采样第二个节点。***3.*** 随后删除第一个节点与第二个节点之间现有的连接。

我们在图15中给出了这种变异的一个示例。

[3]这些动态网络中的采样操作始终采用均匀分布。[4] 其中，N(μ, σ2) 表示均值为 μ、方差为 σ2 的正态分布；所有 DRN 和 DCN 的权重和偏置均初始化为从 N(0, 1) 中采样的值，并在每次迭代时加上从 N(0, 0.01) 中采样的扰动值。

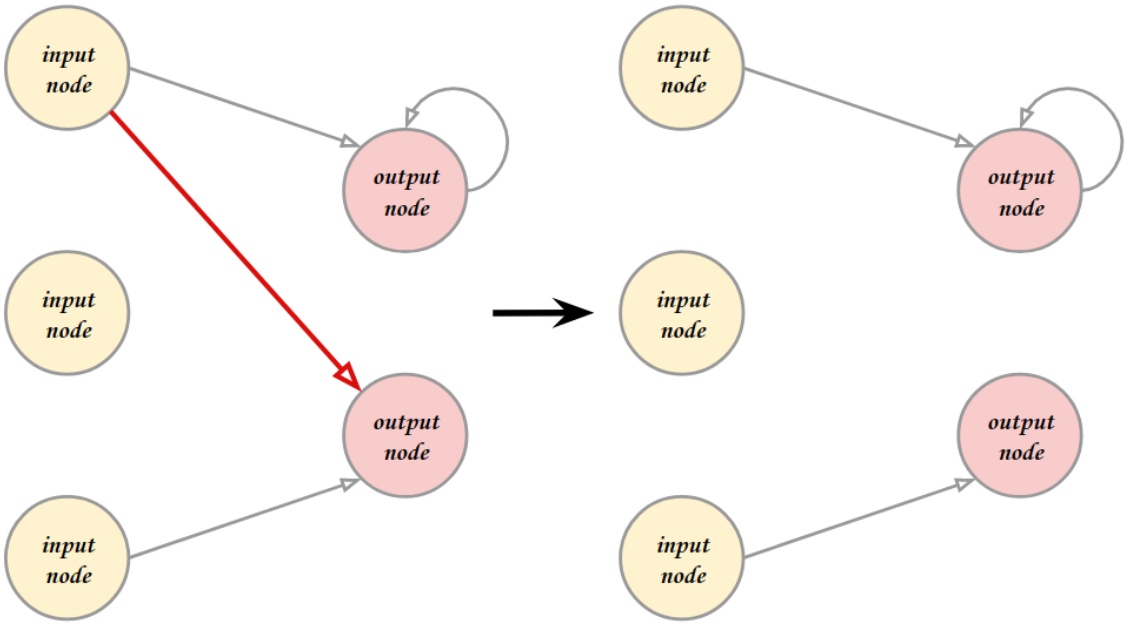

图*15*。 **DRN变异#2：剪枝连接。** 我们继续探讨 图14中引入的情景，并假设此后又应用了三次"增长连接"变异。对这个 DRN 执行"剪枝连接"变异时，需要从一个列表中抽取一个节点。该列表由第一个输入节点（出现两次）、第三个输入节点和第一个输出节点组成。假设抽中的是第一个输入节点。接着，从该节点的输出节点集合中再抽取一个节点，假设抽中的是第二个输出节点。因此，这两个抽中的节点之间的连接被删除。

3.2.1.3. 变异#3：增长节点

1. Th[ree no] d被抽取 led
：- 第一个节点，从所有**接收节点**组成的集合中抽取。- 第二个节点，从所有**接收节点** 组成的集合中抽取，但排除第一个抽中的节点。- 第三个节点，从所有**隐藏**节点和**输出节点**组成的集合中抽取。***2.*** 在中间初始化一个新的隐藏节点。***3.*** 新增三条连接：- 从第一个抽中的节点连接到新的隐藏节点。- 从第二个抽中的节点连接到新的隐藏节点。- 从新的隐藏节点连接到第三个抽中的节点。

我们在图 16中给出了这种变异的示例。

3.2.1.4. 变异#4：剪枝节点

1. 从所有**隐藏**节点中随机选取一个隐藏节点。***2.*** 删除该隐藏节点及其所有连接。由于 DRN 剪枝突变，任何其他因此不再接收或输出信息的隐藏节点也会被删除。

我们在图 17中给出了这种变异的示例。

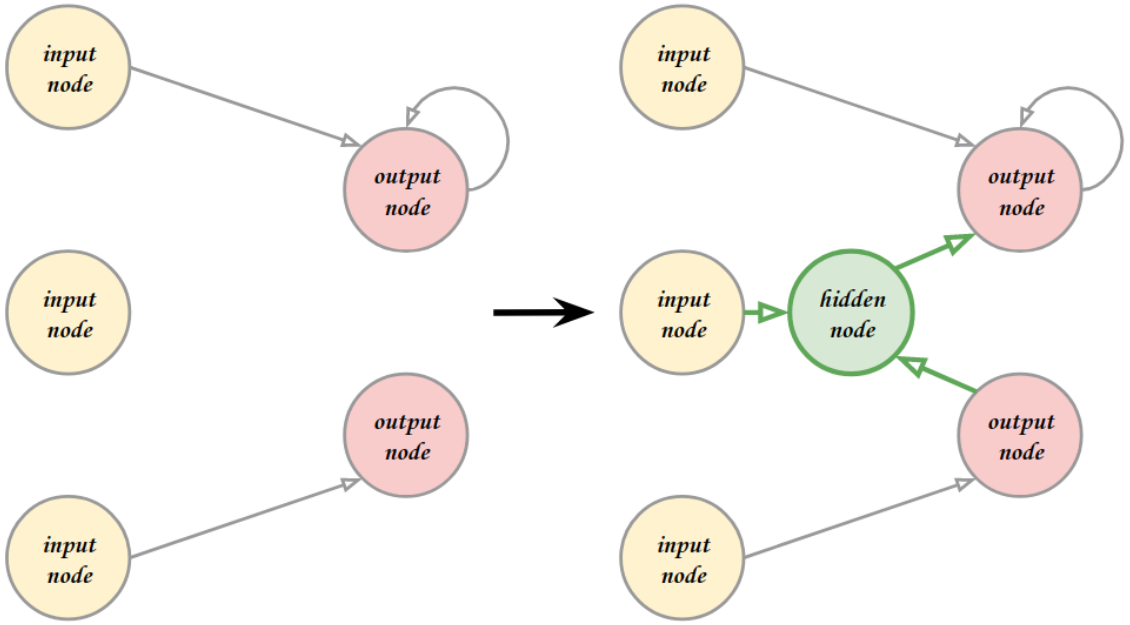

图 *16*。 **DRN变异#3：增长节点。** 我们从图15继续上次的内容。在此情形下调用"增长节点"变异，需要依次抽样：1）从所有可见节点中抽取第一个节点；2）从除第一个抽样节点外的所有可见节点中抽取第二个节点；3）从两个输出节点中抽取第三个节点。我们设想这三个节点依次为第二个输入节点、第二个输出节点和第一个输出节点。随后创建一个隐藏节点，并建立以下连接：1）从第二个输入节点到隐藏节点；2）从第二个输出节点到隐藏节点；3）从隐藏节点到第一个输出节点。

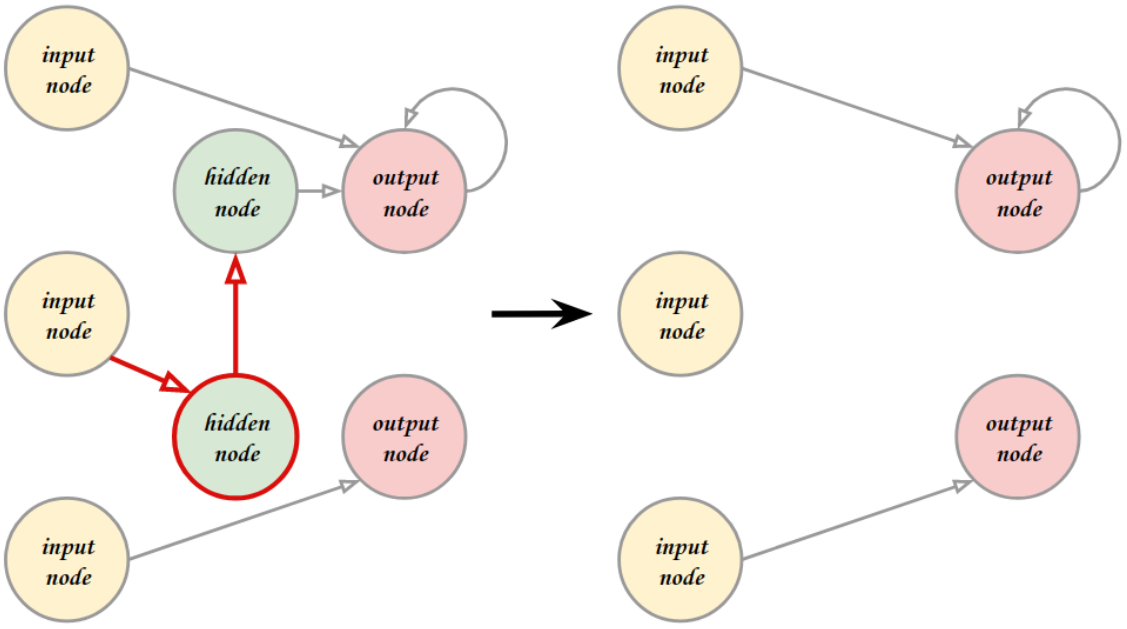

图 *17*。 **DRN变异#4：剪枝节点。** 我们现在设想，自 图 16以来，前文所述的许多变异都已发生。在这个更新后的 DRN 上调用"剪枝节点"变异，需要从两个隐藏节点中抽取一个节点。假设抽中的是第二个隐藏节点，则将其连同它的两条连接一起删除。由于这次删除，第一个隐藏节点因不再包含任何输入信息，也被删除。

3.2.2. 动态卷积网络

下面介绍动态卷积网络（DCN）。该网络中的节点通过树状结构连接，执行卷积和池化操作（图18）。卷积节点使用步长为 1 的标准 3×3 卷积核，输入/输出通道数可变；池化节点则使用尺寸可变的标准方形卷积核，并采用与其尺寸相同的步长。输出多值特征图的节点称为隐藏节点，而输出单个值的节点称为输出节点。除输入特征图小于 3×3 卷积核尺寸的卷积节点外，不使用填充操作。与 DRN 类似，DCN 的复杂度通过以下四种结构变异逐步演化：

1）GrowBranch（第 3.2.2.2 节）
2）剪枝分支（第 3.2.2.3 节）3）扩展节点（第 3.2.2.4 节）4）收缩节点（第 3.2.2.5 节）

我们定义以下术语：
分支节点：分支从中生长出来的节点列表。每个节点在此列表中按其产生的分支数出现一次。

扩展节点：输出多个特征图的卷积节点列表。每个节点在此列表中按其额外输出的特征图数出现一次。
池化因子：节点通过池化操作降低其输入维度的因子。

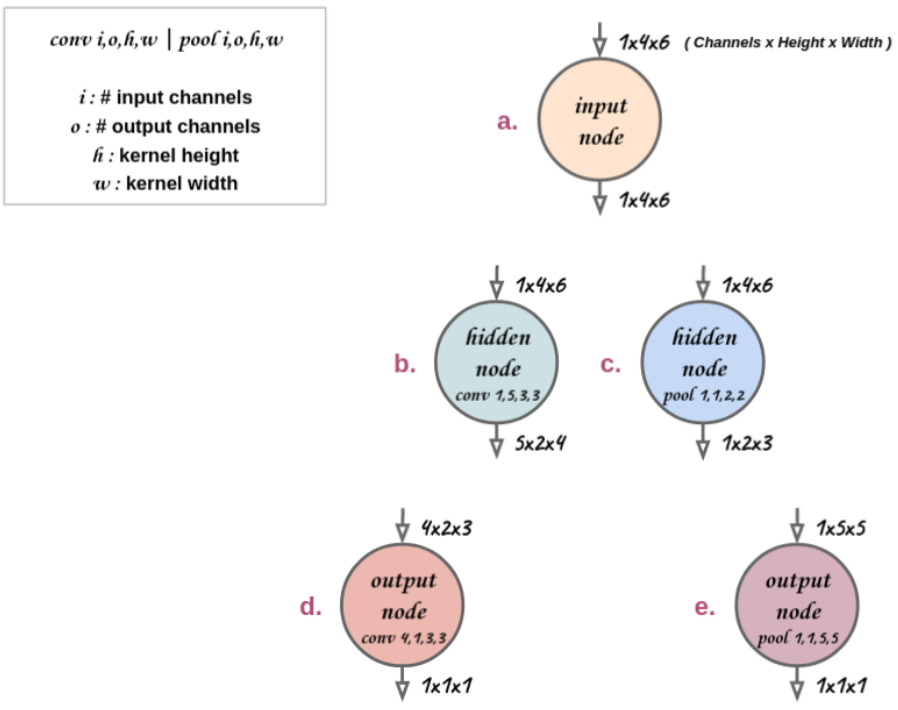

图*18.* **动态卷积网络节点。** DCN 中可能遇到的五种节点示例。DCN 树的根节点是输入节点（a）。内部/隐藏节点执行卷积（b）或（平均）池化操作（c）。叶/输出节点也执行卷积（d）或（平均）池化操作（e），但与隐藏节点不同的是，它们始终只输出一个值。在这些网络中，信息自上而下流动，不发生递归。

3.2.2.1. 新增节点操作

在 DCN 中，除根节点外，每个节点都有一个父节点，并且可能有多个子节点。新的 DCN 节点始终通过“增长节点”操作创建。此操作会创建一个新节点，并将其添加到现有节点的输出节点集合中。执行池化操作的节点总是会扩展出执行卷积操作的子节点；而对于输入节点和卷积节点，则会按照以下子节点扩展流程进行。为了确定新节点的功能（卷积或池化），需要根据父节点的输出维度确定池化因子。最初，第一个新子节点会获得能够整除父节点输出特征图的最大池化因子。然后，在一个子节点按此因子进行池化后，下一个最大的因子会分配给它的同级节点，依此类推，直到没有可用的池化因子；此时，该子节点会被确定为卷积节点。

我们在图19中给出了这一过程的示例。

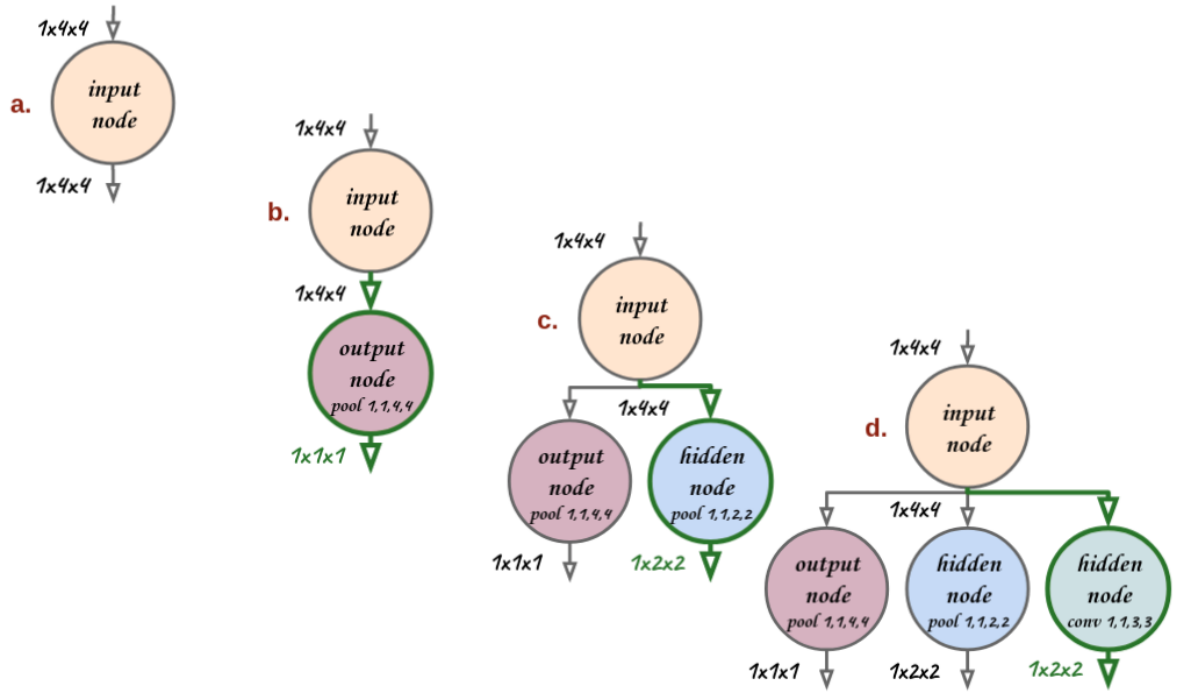

图*19*。 **DCN操作：增长节点。** 为了说明新增节点的过程，我们设置了一个 DCN 输入节点，用于处理大小为 1x4x4 的输入矩阵（a）。此时，对该输入节点调用“增长节点”操作，会创建一个池化因子最适合划分该输入节点输出的节点：池化因子为 4，从而创建一个具有该因子的输出池化节点（b）。再次对该输入节点调用“增长节点”操作，则会创建一个新的节点，其池化因子为第二大的可能值：2，从而创建一个隐藏池化节点（d）。最后，由于所有池化因子都已用尽，再次调用“增长节点”操作便会创建一个隐藏卷积节点。

3.2.2.2. 变异 #1：GrowBranch

1. 从由**输入节点**和所有**隐藏节点**组成的集合中抽取一个基节点。***2.*** 对这个基节点执行“增长节点”操作。

3. 不断对最近创建的节点执行“增长节点”操作，直到该操作生成一个**输出节点**。

我们给出了这一变异的一个示例，见图20。

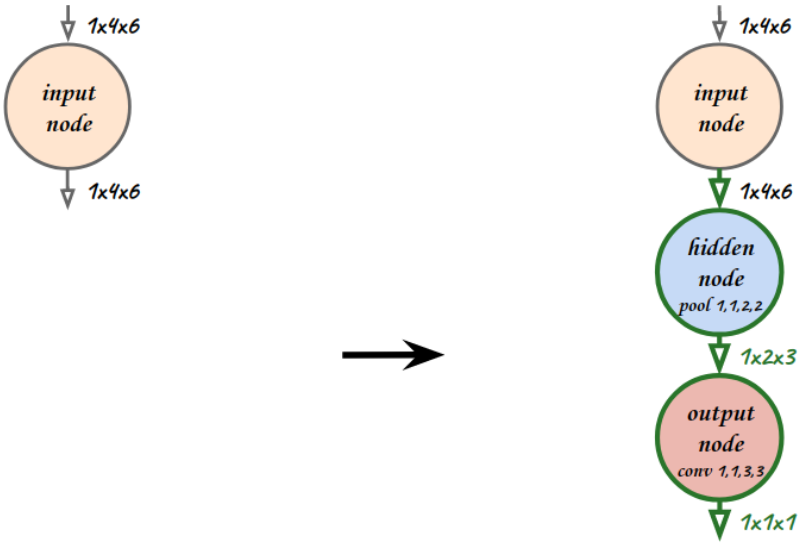

图*20*。 **DCN变异#1：生长分支。** 为了像展示 DRN 那样有效地展示 DCN 的四种变异，我们将在接下来的四幅图中依次说明一个场景：我们需要一个网络处理某个以 1x4x6 矩阵编码的图像信息。为了接收这些信息，DCN 首先初始化为一个对应的输入/根节点。此时，树仅由这一个节点构成，因此调用“生长分支”变异时会自动选中该节点。由于它没有输出节点，对输入节点调用“生长分支”变异会创建一个池化因子最大的新池化节点：2。采用这一池化因子后，新建节点将输出 1x2x3 矩阵，因此被标记为隐藏节点。随后再次调用“增长节点”操作，这次作用于新建的隐藏池化节点。由于该节点是池化节点，其子节点默认设为卷积节点。经过这一卷积操作，最新节点将输出 1x1x1 矩阵，因此被标记为输出节点，标志着“生长分支”变异结束。

3.2.2.3. 变异#2：剪枝分支

1. 从所有**分支节点**组成的列表中随机选取一个节点。***2.*** 从第一个节点的输出节点集合中随机选取第二个节点。***3.*** 随后剪除这个第二个节点及其整棵子树。

我们在图 21中给出了这种变异的示例。

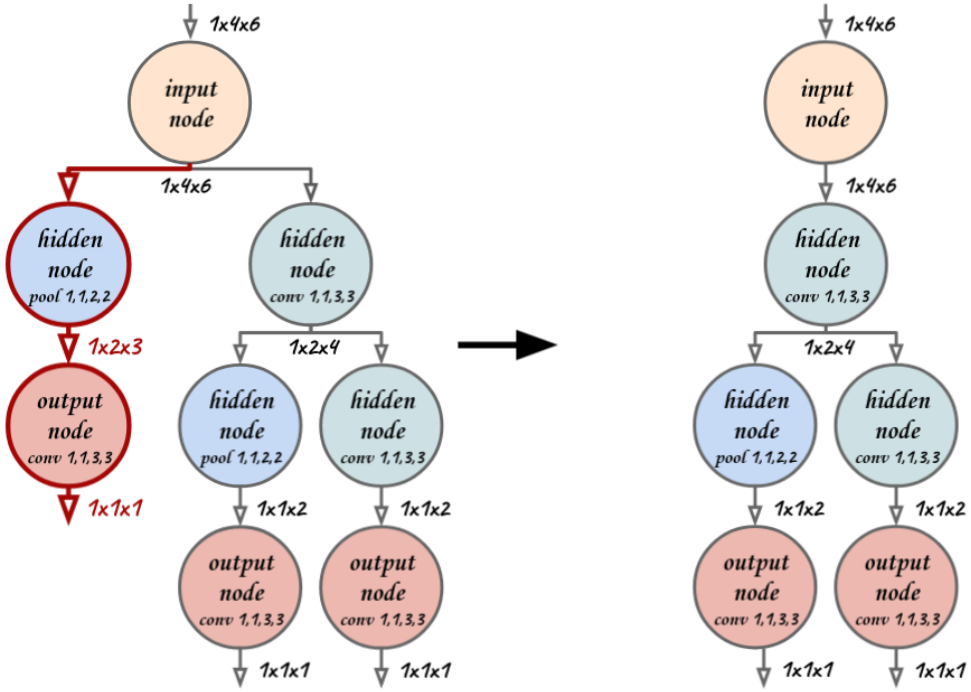

图 *21.* **DCN 变异#2：剪枝分支。** 我们从 图20所介绍的情形继续，并假设又进行了两次"生长分支"变异：一次作用于输入节点，另一次作用于由此产生的第二层隐藏卷积节点。对该 DCN 执行"剪枝分支"变异时，需要从以下节点组成的列表中抽取第一个节点：输入节点（出现两次）和第二层隐藏卷积节点。假设抽中的是第一个输入节点，随后便从该输入节点的输出节点集合中抽取第二个节点；我们假设抽中的是第二层隐藏池化节点。因此，该隐藏池化节点及其子树（由一个输出卷积节点组成）将被删除。

3.2.2.4. 变异#3：扩展节点

1. 从**卷积节点**集合中抽取一个节点，该节点既不是**输出节点**，其**输出节点**也不是**输出池化节点**。
2. 为该节点随机初始化一组新的权重，使其从输入中额外输出一个特征图。

3. 该节点的卷积子节点（以及每个池化子节点的孙节点）会进行必要的调整（创建一组新的权重），以便接收这个额外的特征图。

我们在图 22中给出了这种变异的示例。

3.2.2.5. 变异#4：收缩节点

1. 从**已扩展的**节点列表中抽取一个节点。
2. 抽取并删除该节点的一个输出权重通道。随后删除该节点的子节点（以及每个池化子节点的孙节点）中对应的输入权重通道。

我们在图 23中给出了这种变异的示例。

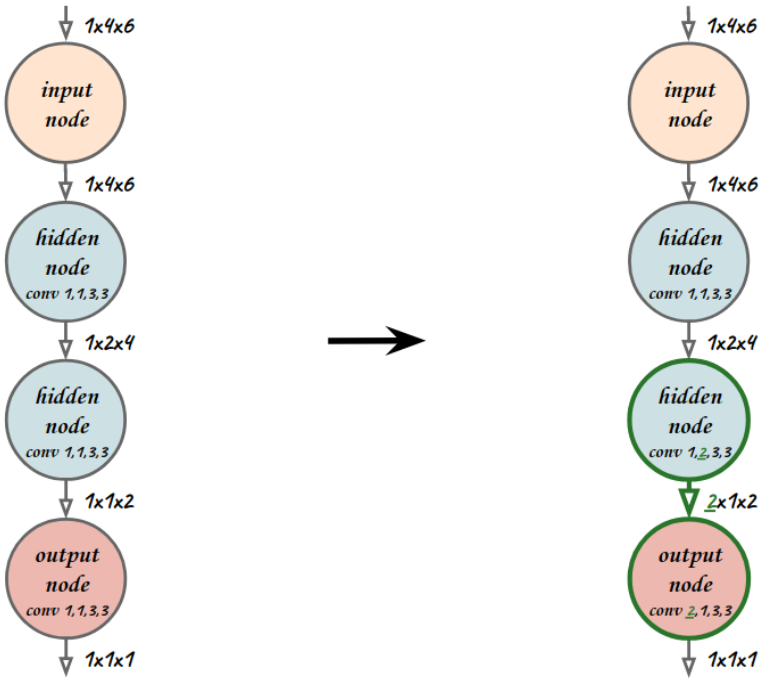

图 *22*。 **DCN 变异 #3：扩展节点。** 我们从 图 21 接着往下讲，并假设已对第二层隐藏节点执行了一次“剪枝分支”变异，从而删除了其左子树。对此 DCN 执行“扩展节点”变异时，首先要从由所有隐藏节点组成的集合中进行采样。假设选中的是第三层隐藏节点。该节点内会初始化一组新的权重，以输出第二个特征图。其输出节点（即输出卷积节点）会初始化新的权重，以便利用这个新的特征图。

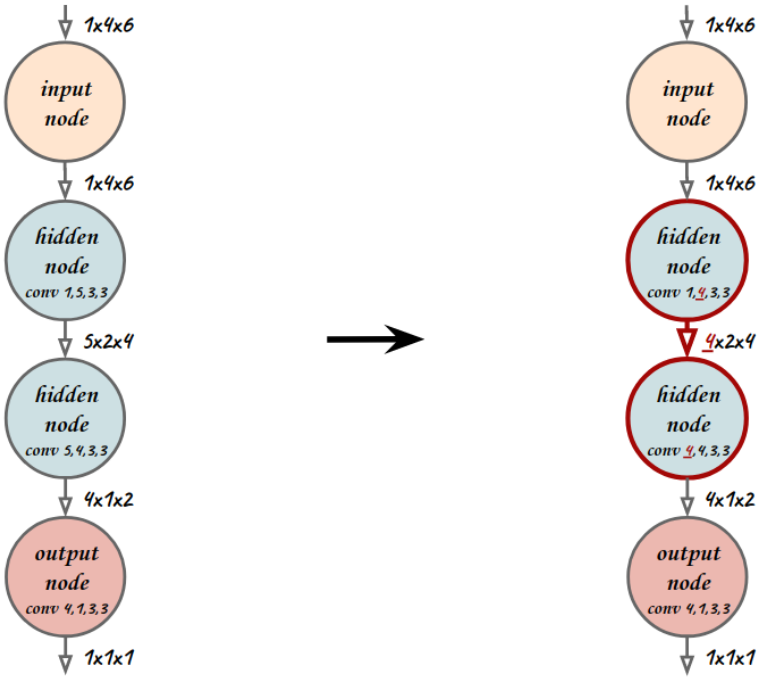

图 *23*。 **DCN 变异 #4：收缩节点。** 对于最后这项变异，我们假设，由于 图22，已执行了六次新的“扩展节点”变异：四次针对第二层隐藏节点，两次针对第三层隐藏节点。对此 DCN 执行“收缩节点”变异时，需要从由第二层隐藏节点（出现四次）和第三层隐藏节点（出现三次）组成的列表中采样一个节点。假设选中的是第二层隐藏节点。因此，其一个输出权重通道会被随机删除。其输出节点（即第三层卷积节点）也会删除相应的权重。

3.2.3. 连接动态网络

由于输出为单一值，动态卷积网络（DCN）可以连接到动态循环网络（DRN）。为了处理 DCN 因“生长分支”和“剪枝分支”变异而产生的动态数量叶节点，DRN 需要能够处理数量可变的输入节点。为此，将执行以下操作：

- 调用 DCN 的“生长分支”变异后，DRN 会生长出相应的输入节点，随后将其连接到一个随机的隐藏节点或输出节点（图24）。

- 调用 DCN 的“剪枝分支”变异后，DRN 会剪除相应的（可能有多个）输入节点，以及与之相连的连接和因不再包含输入信息而失去输入的隐藏节点（图25）。

连接这两个动态网络会产生八种结构变异。无论连接的网络数量是多少，每次迭代都会从所有潜在变异构成的集合中抽取一种变异，随后将其应用于相应的子网络。

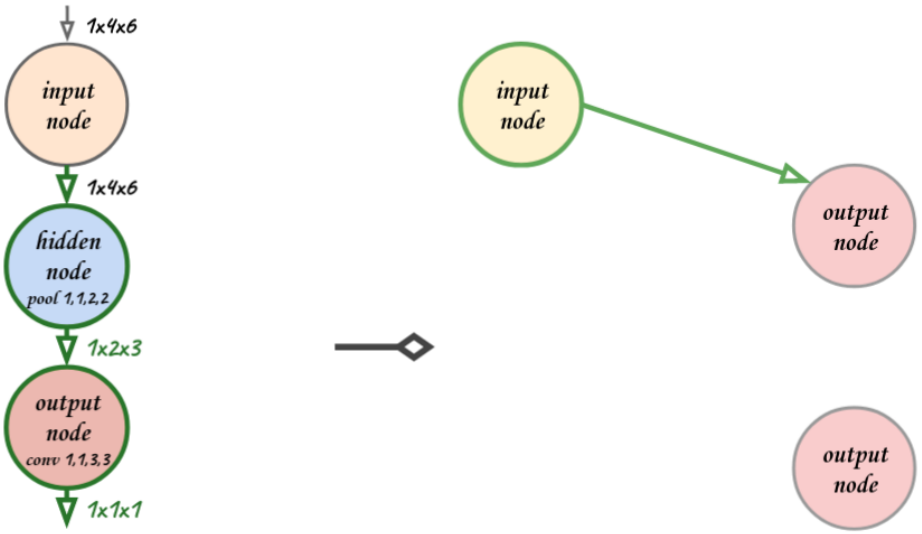

图*24*。**连接动态网络：GrowBranch 变异。**在接下来的两幅图中，我们将介绍一个修订后的假设场景：我们希望神经网络接收一幅 1x4x6 的图像，并输出两个标量值。为此，我们初始化一个 DCN 来接收该输入矩阵，并初始化一个 DRN 来输出这两个标量值。与图20类似，DCN 初始化时包含一个输入节点，而与图14不同，DRN 初始化时只包含两个输出节点。现在，再次对 DCN 调用“生长分支”变异，会产生一系列节点，直至创建一个输出/叶节点。由于这个新生成的 DCN 输出节点，DRN 会创建一个输入节点，以接收该叶节点输出的值。创建该输入节点后，从所有隐藏节点和输出节点组成的集合中随机选取一个节点；在此例中，我们假定选中第一个输出节点，最后将输入节点连接到该节点。

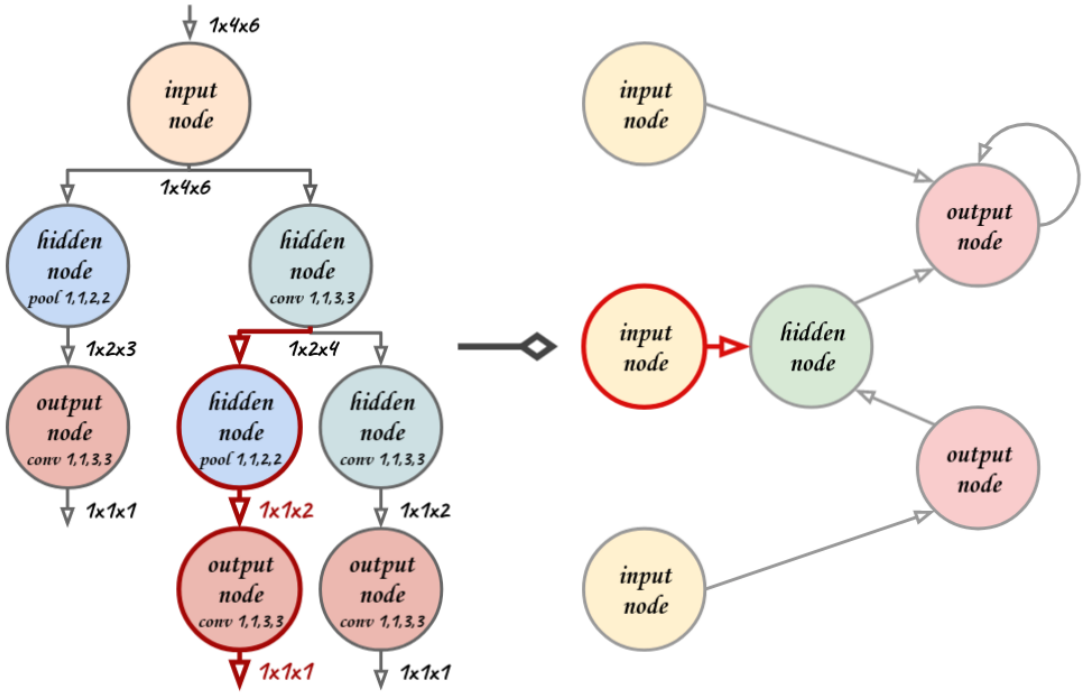

图*25.* **连接动态网络：剪枝分支突变。** 现在假设自 图24以来，两个子网络内部都发生了许多突变，从而产生了三条 DCN-DRN连接。在这种配置下，如果对 DCN 第三层的隐藏池化节点进行"剪枝分支"突变，不仅 DCN 分支会被删除，相应的第二个 DRN输入节点及其唯一连接也会被剪除。

3.3. 面向进化算法的可扩展通信协议

如前所述，进化算法通常需要通过一个由三个阶段组成的迭代流程来建立智能体种群；这三个阶段分别是变异、评估和选择。自然，种群中智能体数量的增加会提高每个阶段的计算成本。不过，大多数情况下，各阶段中执行的操作都可以并行化。如果智能体之间没有交互，甚至可以针对每个智能体完全独立地进行变异和评估。在这种情况下，应将工作负载分派到不同进程，以提高效率。

Such等，2017 最近提出了一种技术，其中变异应编码为"种子"。在该框架中，每个种子都会对施加于标准深度神经网络所有权重的扰动进行编码；智能体在进化过程中会不断追加代表新扰动的种子，并将它们层层叠加。

最简单的实现方式是将进程拆分为一个主进程和多个子进程。每次迭代时，主进程生成新的种子，并将其分发给子进程；子进程随后构建并评估分配给它们的智能体。完成后，主进程收集所有智能体的适应度，以便选出用于构建下一世代的种子（图26）。然而，在这种框架下，每一世代都需要重新构建智能体，导致变异阶段的成本至少会持续线性增长。变异操作成本较低时，

这种成本增长并不显著。然而，随着变异操作的计算成本不断增加，这种方法很快就会变得难以行得通。

因此，我们提出了一种替代实现，利用 50% 的重复截断选择。我们设置了一种通信方案：每一世代，每个持有入选智能体的进程都会将其发送给一个持有未入选智能体的进程（图27）。采用这种方法，在第 ***n*** 次迭代中，进程将 ***n*** 次变异操作替换为 ***1*** 次通信操作和 ***1*** 次变异操作。由于现代计算机集群中的进程间具有很高的通信带宽，这种通信操作几乎不产生额外成本，从而实现了可扩展性更强的进化过程。

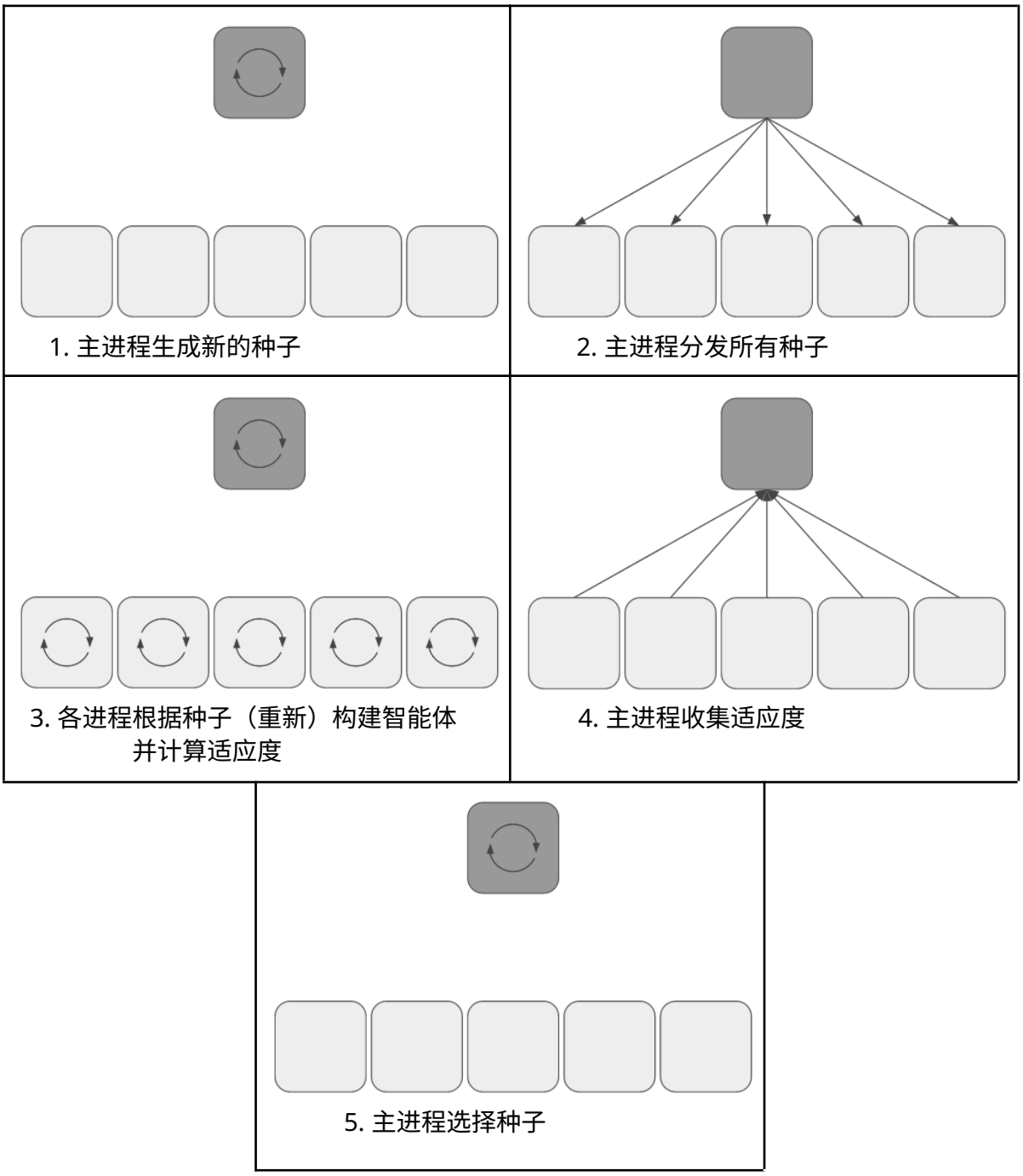

图*26*。**进化式进程间通信：分散/收集。**

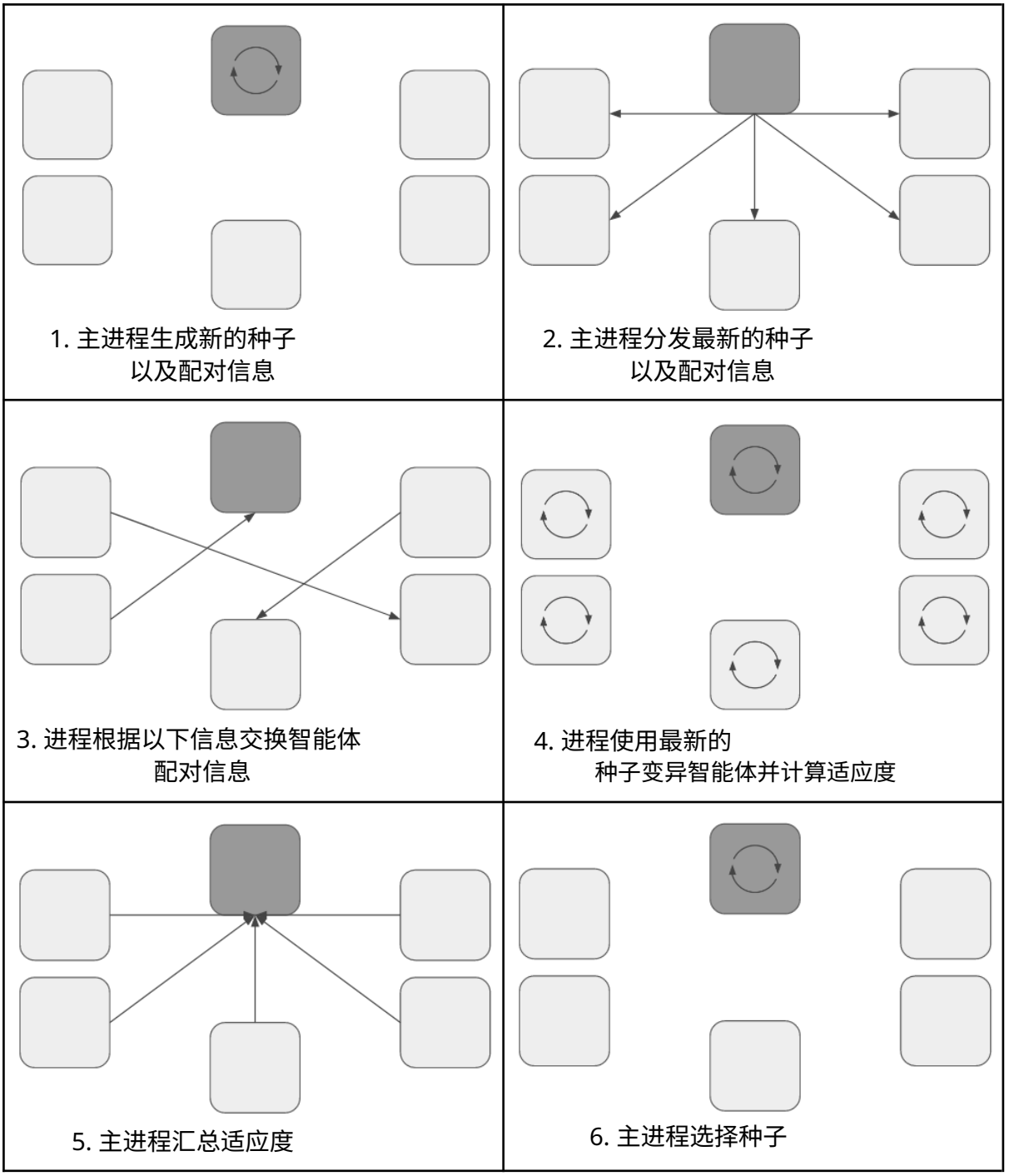

图*27.* **进化式进程间通信：分散/收集 + 点对点** **r.**

4. 实验

4.1. 分数优化与深度强化学习智能体模仿

我们在各种任务中进化了两类智能体：一类是得分最大化智能体，根据其在单个回合中获得的分数进行选择；另一类是模仿预训练深度强化学习智能体的智能体（Raffin 等人，2019），并在所提出的对抗生成框架中进行选择。我们通过考察不同规模种群在五次采用新随机种子的奔跑中获得的平均得分，评估各种动作执行智能体。

4.1.1. CartPole 和 LunarLander

我们首先在 CartPole 和 LunarLander 上评估我们的方法。这些任务中，输入神经网络的状态观测会被编码为一组标量值。在实验的对抗生成部分，我们分别尝试模仿近端策略优化（Schulman 等人，2017）和 TQC（Kuznetsov 等，2020年智能体。我们尝试了两种类型的网络：动态循环网络（DRN），以及深度强化学习智能体所用的三层前馈神经网络（判别器除外；对于判别器，我们将最后一个隐藏全连接层改为循环层）。我们发现，得分最大化的动态网络和静态网络都能够解决这些任务。此外，为模仿而优化的智能体表现出了相当强的能力，并且随着种群规模增大而不断进步。

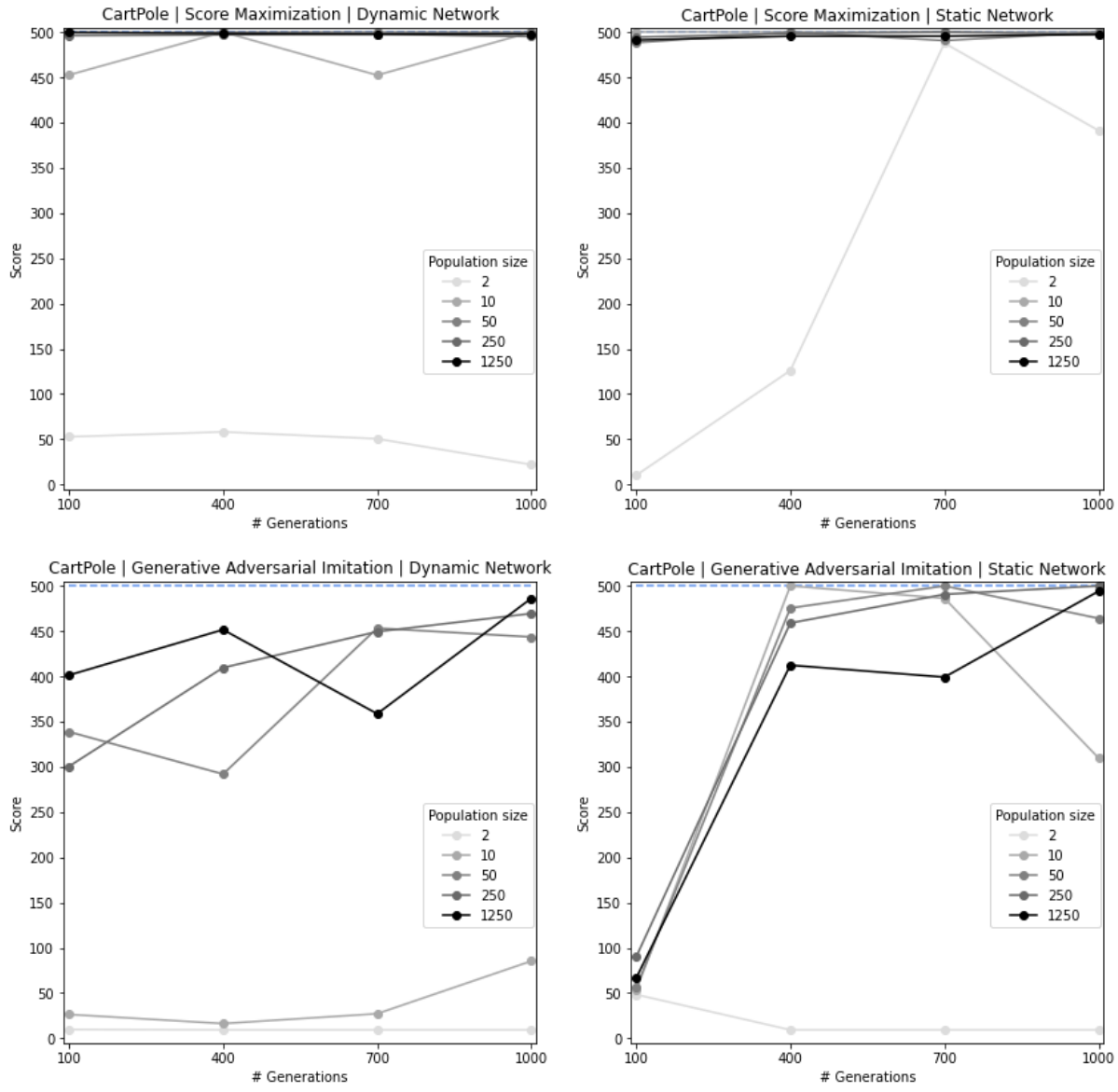

图*28.* **CartPole 上的分数最大化与生成对抗模仿。** 蓝色虚线表示近端策略优化取得的平均得分，这也是该任务可达到的最高得分。从种群规模为 10 开始，动态和静态分数优化智能体也都能持续达到最高得分。对于这两种网络类型，从种群规模为 50 开始进行模仿的智能体也逐渐达到最高得分。

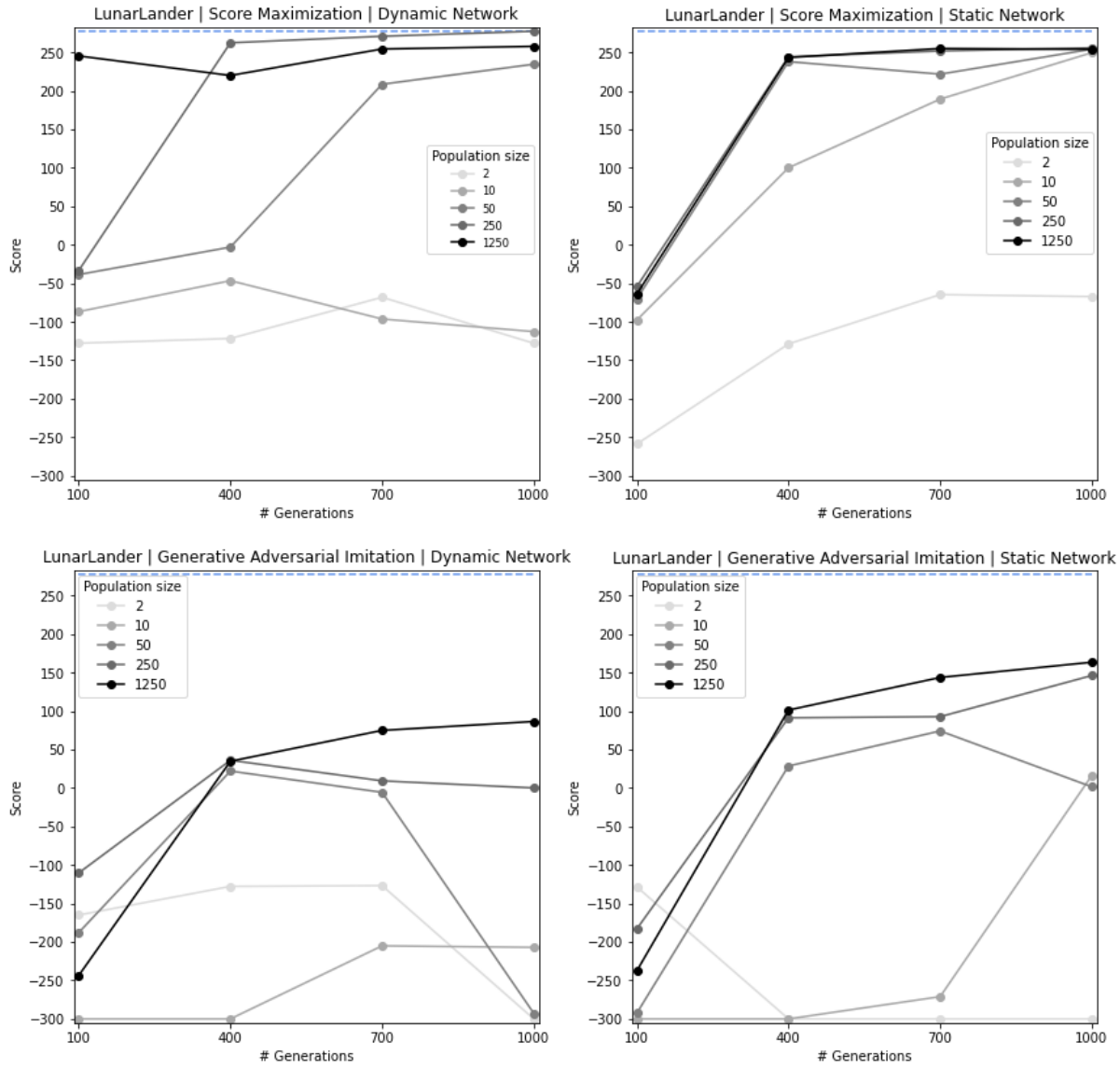

图*29*。 **LunarLander 上的分数最大化和生成对抗模仿。** 蓝色虚线表示 TQC 获得的平均得分。当种群规模分别达到 50 和 10 时，进行得分优化的动态网络和静态网络均达到与最先进基线相近的性能。对于动态网络和静态网络，通过对抗生成进行优化的动作执行智能体，其性能均随着迭代次数和种群规模的增加而显著提升。

4.1.2. Pong

接下来，我们在 Pong 上进行实验。这是一项深度强化学习智能体历来表现出色的任务（Mnih 等人，2013；Schulman 等，2017），但深度遗传算法智能体在这项任务上表现不佳（Such 等，2017）。随后，我们采用结合动态卷积网络（DCN）与动态循环网络（DRN）的动态网络，并将其与参数量为 10 万的深度卷积循环网络进行比较。与前两个任务中整个回合不超过几秒不同，一局 Pong 可能持续几分钟。因此，作为概念验证，我们提议只让智能体进行前 400 帧的游戏（100 个动作， ~7 秒）。最后，在实验的对抗生成部分，我们尝试模仿一个近端策略优化（Schulman 等，2017）智能体。我们发现，在得分优化任务中，动态网络的表现远胜静态网络；而在 1000 个世代后，所有为模仿而优化的智能体表现都不如得分最大化智能体

世代中，随着种群规模增大，动态网络呈现出明显的改进迹象。

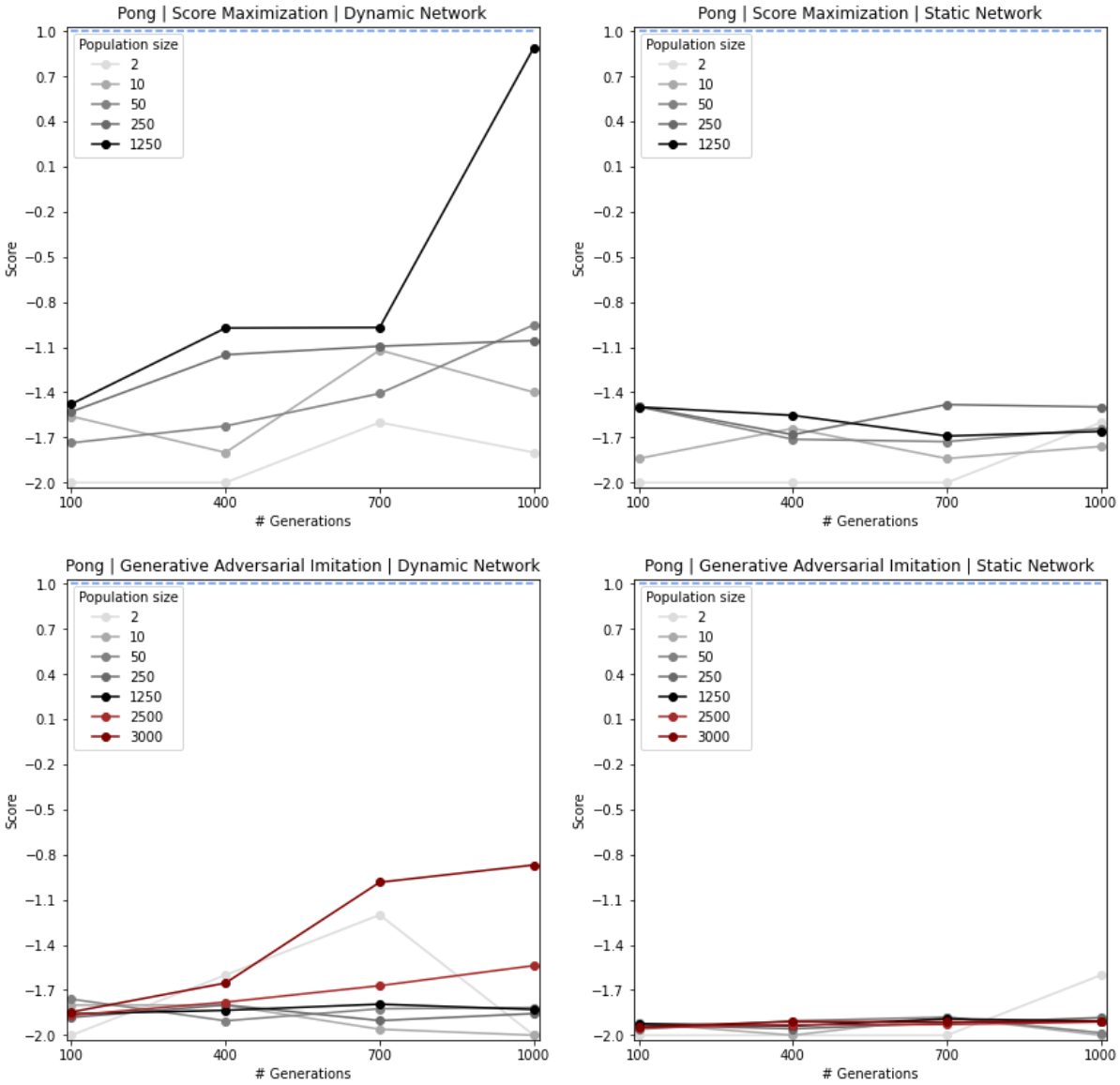

图*30*。 **Pong 中的分数最大化与生成对抗模仿。** 蓝色虚线表示通过近端策略优化（PPO）获得的平均得分。在纯粹的分数最大化任务中，经过 1000 代后，动态网络的表现远胜于静态网络；在规模为 1250 的种群中，其平均得分达到 0.89 分。然而，通过对抗生成进行优化的动作执行智能体，在最初 1000 代后的表现远逊于以得分为优化目标的同类智能体；不过，随着种群规模增大，动态智能体似乎呈现出上升趋势。

4.2. 在 Shinobi III：忍者大师归来中模仿人类行为

现在，我们转而尝试在电子游戏环境中创建具有类似人类行为的智能体。我们使用通过 gym-retro 收集的行为数据（Nichol 等，2018），这些数据来自 Courtois 神经调控项目（Boyle 等，2019），研究对象是电子游戏*Shinobi III: Return of the Ninja Master* [5]。该游戏使用控制器进行操作（图31）

[5] 实验方案已获当地机构审查委员会——“老化与神经影像学研究伦理委员会”批准，所有参与者均已提供知情同意并同意参与本研究。

玩家可以通过控制器使用各种动作及动作组合来移动并与非玩家角色互动（表1）。随机智能体和分数优化智能体所表现出的行为见表2。然而，我们的主体并非试图优化得分，因此其行为与前两个示例不同（表3）。

我们再次采用进化对抗生成框架，为生成器和判别器都配备一个馈入 DRN 的 DCN，同时也针对参数量相近、约十万参数的深度神经网络开展实验。

为了量化这些智能体的质量，我们首先为每个主体制定定性行为评估表（表4）。随后，我们对智能体所展现的模仿质量进行定性评估（表5）。据此，我们将结果展示在表6中。我们发现，只要计算资源充足，嵌入静态网络和动态网络的智能体所表现出的行为都与相应的人类主体相似。然而，使用动态复杂度网络的智能体似乎能更好地利用增加的计算资源；对于 sub-02，其行为几乎与人类主体无法区分。在这一特定情况下，两个子网络各由十层组成，包含约一百个节点，并通过四十个 DCN输出节点交换信息。

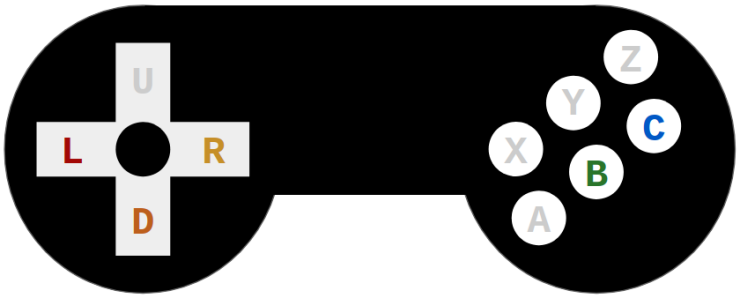

图*31*。 **世嘉Genesis控制器示意图。**

R	向右行走	视频
L	向左走	视频
D	蹲下	视频
C	jump	视频
B	（附近没有敌人）投掷一个投射物	视频
B	（附近有敌人）挥动刀刃	视频
R，暂停，R	run	视频

C，暂停，C	二段跳	视频
C，暂停，B	跳跃并投掷弹丸	视频
R，暂停，R+B	奔跑并挥刀	视频
C，暂停，D+B	跳跃下踢	视频
C，暂停，C+B	二段跳并投掷投射物	视频

表*1.* **Shinobi III 中可能的动作及动作组合的视频样本。**

随机智能体	分数优化智能体
视频	视频

表*2.* **Shinobi III 中随机智能体和分数优化智能体的视频样本。**

sub-01					sub-02				
视频	视频	视频	视频	视频	视频	视频	视频	视频	视频

表*3.* **Shinobi III 中主体的行为视频样本。**

	受试者 01	受试者 02
elements of behaviour	- 向右移动 - 不投掷投射物 - 跳跃 - 跳跃后下踢 - 奔跑 - 奔跑后挥刀	- 向右移动 - 不投掷投射物 - 不跳跃 - 奔跑 - 奔跑后挥刀
behaviour	1) 通过奔跑或行走 向右，1 次跳跃下踢，向右行走 2）用1 次 挥击 或 1 次跳跃下踢 3）通过以下方式向右移动：奔跑或 行走 向右，1 次跳跃下踢，向右行走 4）忽略第二个角色在 tree 5）用1次挥击击倒第三个角色 挥击或2次跳跃下踢	1）通过奔跑向右移动 2）用1次挥击击倒第一个角色 挥击 3）通过奔跑向右移动 4) 忽略了该场景中的第二个角色 tree 5) 一次击倒第三个角色，使用1x 挥击

表*4.* **Shinobi III 中受试者行为的定性评估表。**

	智能体行为与主体行为无法区分
	智能体行为与主体行为几乎无法区分
	智能体行为与主体行为非常相似，但肯定可以区分
	智能体行为大体上与主体行为相似
	智能体行为正逐渐变得与主体行为相似
	智能体明显表现出主体行为的特征
	智能体似乎表现出了主体行为的多个要素
	智能体似乎至少表现出了主体行为的两个要素
	智能体的行为几乎与主体完全不同
	智能体的行为与主体完全不同

表5。**Shinobi III 中智能体行为的定性评估表。**为了评估智能体在游戏过程最初几秒内模仿主体的能力，我们根据表4中的观察结果制定了一项性能定性指标。

	时间 ↓	sub-01				sub-02			
pop 尺寸 →		10	40	160	640	10	40	160	640
static	12h	视频	视频	视频	视频	视频	视频	视频	视频
	24h	视频	视频	视频	视频	视频	视频	视频	视频
	36h	视频	视频	视频	视频	视频	视频	视频	视频
	48h	视频	视频	视频	视频	视频	视频	视频	视频
dynamic	12h	视频	视频	视频	视频	视频	视频	视频	视频
	24h	视频	视频	视频	视频	视频	视频	视频	视频
	36h	视频	视频	视频	视频	视频	视频	视频	视频
	48h	视频	视频	视频	视频	视频	视频	视频	视频

表6。 **Shinobi III 的模仿结果表。** 我们针对四种种群规模（每个进程一个智能体）分别进行了为期48小时的实验。为了评估动态架构，我们还使用标准静态深度神经网络进行了实验。此处展示的每个视频样本，都是通过查看种群中的前三个生成器，并选出我们认为其行为最接近相应主体的生成器而收集的。我们采用颜色代码，对照文中所示的定性评估表进行标注：表5。

5. 讨论

根据这些实验，我们认为，只要有足够的计算资源，确实有可能成功进化出人工智能体，使其在电子游戏环境中的行为与人类受试者相似。对抗生成框架似乎非常适合进化人工智能体，使其能够编码以下复杂功能：像人类玩家一样进行游戏，并区分人类玩家的行为与其他类型的行为。同样重要的是，这两种复杂功能似乎都可以编码在所提出的动态复杂度神经网络中。最后，我们想指出点对点通信协议的价值；正是该协议使这些智能体能够进化数万代，而不只是数百代。

尽管本研究仅展示了对短时行为序列的建模，但这些初步结果增强了我们对这些技术能够建模更长时间跨度的信心。最后，这项工作进一步表明，随着资源日益强大，进化计算未来或许能够应对更复杂的问题。

6. 结论

本文介绍了三种独立的进化计算技术：进化对抗生成框架、用于开发动态复杂度人工神经网络的流程，以及一种可扩展的进程间通信协议。在计算资源充足的情况下，结合这些技术可生成电子游戏行为，在较短的时间窗口内与人类受试者的行为几乎相同。这些结果进一步表明，进化优化可用于解决复杂的高维问题。

参考文献

Amodei, Dario 和 Jack Clark。《野外的错误奖励函数》。（2016）。Arulkumaran, Kai、Antoine Cully 和 Julian Togelius。《Alphastar：进化计算视角》。（2019年）。

Baricelli, Nils A.。《进化过程的数值示例》。（1954年）。

Belew, Richard K.、John McInerney 和 Nicol N. Schraudolph。《进化网络：使用遗传算法与连接主义学习》。（1990年）。

Berner, Christopher 等。《通过大规模深度强化学习进行 Dota 2 对战》。（2019年）。

Boyle, Julie A. 等。《Courtois项目的神经元建模——首次数据发布》。（2020年）。

Brown, Tom B. 等。《语言模型是少样本学习器》。（2020年）。

Costa, Victor 等人。《生成对抗网络的神经进化》。（2020年）。

Farley, Belmont G. 和 Wesley A. Clark。《通过数字计算机仿真自组织系统》。（1954

oreano, Dario、Peter Dürr 和 Claudio Mattiussi。《神经进化：从架构到学习》。（2008年）。

Fogel, David B.、Lawrence J. Fogel 和 Vincent W. Porto。《进化神经网络》。（1990年）。

Fogel, Lawrence J., Alvin J. Owens 和 Michael J. Walsh。《通过模拟进化实现人工智能》（1966).

Fraser, Alex S.《使用自动数字计算机模拟遗传系统 I：引言》（1957年）。

Goodfellow, Ian 等。《生成对抗网络》（2014）。

Ho, Jonathan 和 Stefano Ermon。《生成对抗模仿学习》（2016）。

Holland, John H.《自然与人工系统中的适应：生物学、控制与人工智能应用导论》（1975年）。

Hopfield, John J.《具有涌现集体计算能力的神经网络与物理系统》（1982年）。

Hussein, Ahmed 等人。《模仿学习：学习方法综述》。（2017）。

Jalali, Seyed Mohammad Jafar 等人。《通过深度模仿学习和神经进化实现最优自动驾驶》。（2019年）。

Karras, Tero、Samuli Laine 和 Timo Aila。《生成对抗网络的基于风格的生成器架构》。（2019年）。

Ki, Hyun-woo、Ji-hye Lyu 和 Kyoung-su Oh。《用于模仿游戏玩家的实时神经进化》。（2006).

Kuznetsov, Arsenii 等人。《通过截断连续分布分位数评论家混合控制高估偏差》。（2020年）。

LeCun, Yann 等人。《将反向传播应用于手写邮政编码识别》。（1989年）。

LeCun、Yann、Yoshua Bengio 和 Geoffrey Hinton。“深度学习。”（2015年）。

McCulloch, Warren S. 和 Walter Pitts。《神经活动中固有观念的逻辑演算》。（1943年）。

Miller, Geoffrey F.、Peter M. Todd 和 Shailesh U. Hegde。《使用遗传算法设计神经网络》。（1989年）。

Miranda, Maximiliano, Antonio A. Sánchez-Ruiz 和 Federico Peinado。《一种在吃豆小姐电子游戏中模仿类人游戏行为的神经进化方法》（2016）。

Mnih, Volodymyr 等。《使用深度强化学习玩雅达利游戏》（2013）。

Montana, David J. 和 Lawrence Davis。《使用遗传算法训练前馈神经网络》（1989年）。

Nichol, Alex 等。《快速学习：强化学习泛化的新基准》（2018年）。

Raffin, Antonin 等。《Stable Baselines3》（2019年）。

Rechenberg, Ingo。《进化策略：根据生物进化原理优化技术系统》（1973年）。

Risi，Sebastian 和 Kenneth O. Stanley。《循环神经网络与离散世界模型的深度神经进化》。（2019年）。

Rosenblatt，Frank。《感知机：一种用于大脑中信息存储与组织的概率模型》。（1958年）

Rumelhart, David E.、Geoffrey E. Hinton 和 Ronald J. Williams。《通过误差传播学习内部表征》。（1985年）。

Salimans、Tim 等。《将进化策略作为强化学习的可扩展替代方案》。（2017）。

Schmidhuber, Jürgen。《神经网络中的深度学习：综述》。（2015年）。

Schulman, John 等。《近端策略优化算法》。（2017）。

Schwefel, Hans-Paul。《计算机模型的数值优化》。（1974年）。

Silver, David 等。《利用深度神经网络和树搜索掌握围棋》。（2016）。

Stanley, Kenneth O. 和 Risto Miikkulainen。《通过进化复杂化实现竞争性协同进化》。（2004年）。

Stanley, Kenneth O. 和 Risto Miikkulainen。《通过拓扑增广实现神经网络进化》。（2002年）。

Stanley, Kenneth O. 等。《通过神经进化设计神经网络》。（2019年）。

Such, Felipe P. 等。《深度神经进化：遗传算法是训练用于强化学习的深度神经网络的一种有竞争力的替代方案》。（2017）。

Sutton, Richard S. 和 Andrew G. Barto。《强化学习：导论》。（2018年）。

Tan, Mingxing 和 Quoc V. Le。《EfficientNet：重新思考卷积神经网络的模型缩放》。（2019).

Vinyals, Oriol 等。《使用多智能体强化学习达到《星际争霸II》宗师级水平》。（2019年）。

Whitley, Darrell、Timothy Starkweather 和 Christopher Bogart。《遗传算法与神经网络：优化连接与连通性》。（1990年）。

Widrow、Bernard 和 Marcian E. Hoff。《自适应开关电路》。（1960年）。

福岛邦彦。《不受位置偏移影响的模式识别机制的神经网络模型——新认知机》。（1979年）。