arXiv:2301.01452v3 [cs.LG] 2 Apr 2023

预测式前向-前向算法

Alexander Ororbia罗切斯
特理工学院 ago@cs.rit.edu

Ankur Mali
南佛罗里达大学
ankurarjunmali@usf.edu

摘要

我们提出预测式前向-前向（PFF）算法，用于在神经系统中进行信用分配。具体而言，我们设计了一种新颖的动态循环神经系统，使其能够与表征电路联合、同步地学习有向生成电路。值得注意的是，该系统将可学习的侧向竞争、噪声注入以及预测编码（这一新兴且可行的皮层功能神经生物学过程理论）的要素，与前向-前向（FF）适应方案相结合。此外，PFF 只需前向传播即可高效地学习传播学习信号并更新突触，从而消除基于反向传播的方案所施加的关键结构和计算限制。除了计算优势之外，PFF 过程还有望帮助我们理解生物神经元背后的学习机制：尽管缺少反馈连接，它们仍能利用局部信号。我们在图像数据上开展实验，证明 PFF 过程的效果与反向传播相当，是一种前景可期的脑启发算法，可用于分类、重构和合成数据模式。

关键词 脑启发计算 · 自监督学习 · 神经形态 · 前向学习

1 引言

被称为误差反向传播 [59, 32], （简称“反向传播”）的算法，长期以来因其神经生物学合理性而受到质疑 [10, 14, 56, 35, 15]。尽管它推动了深度学习的巨大进步与成功，并催生了日益丰富、前景广阔的应用 [57, 12], , 但反向传播不太可能是大脑（例如皮层区域）中的一种可行学习模型。值得注意的是，这既存在实际问题，也存在生物物理问题 [15, 35], ；其中包括缺乏证据证明：**1)** 神经活动会被明确存储起来，以供之后进行突触调节；**2)** 误差导数会沿全局反馈通路反向传播，以生成教学信号；**3)** 误差信号会沿着用于前向传播信息的同一神经通路反向传递；以及**4)** 推理与学习在很大程度上是彼此锁定、按顺序进行的（而非大规模并行进行）。此外，在处理时序数据时，大脑的神经回路显然不会沿时间反向展开以调整突触 [42] （如时间反向传播所示）。

近年来，脑启发计算这一研究领域受到越来越多的关注，其重点是开发算法和计算模型，以尝试规避或解决上述关键问题。其中最强大、最有前景的方法之一是预测编码（PC） [18, 48, 13, 4, 51, 41],，而最新的方法之一是前向-前向（FF）算法 [19]。这些替代方法提供了不同的信用分配方式，其性能与反向传播相近；但与反向传播相反，它们更有可能符合并类似于真实的生物神经元学习（有关各自信用分配设置的图示和比较，请参见图 1）。本文将提出一种新颖的模型和学习过程，即**预测式前向-前向（PFF）** 过程。该过程将 FF 和 PC 泛化并结合起来，构成一个稳健的随机神经系统，能够以生物学上合理的方式同时学习表征和生成模型。与 FF 算法一样，PFF 过程为生物神经回路提供了一种颇具前景、可能有助益的模型，也可能成为低功耗模拟硬件和神经形态电路的候选系统，并且有望成为值得进一步研究和探讨的反向传播替代方案。[1]

[1]PFF 算法的代码可 在此处找到：https://github.com/ago109/predictive-forward-forward

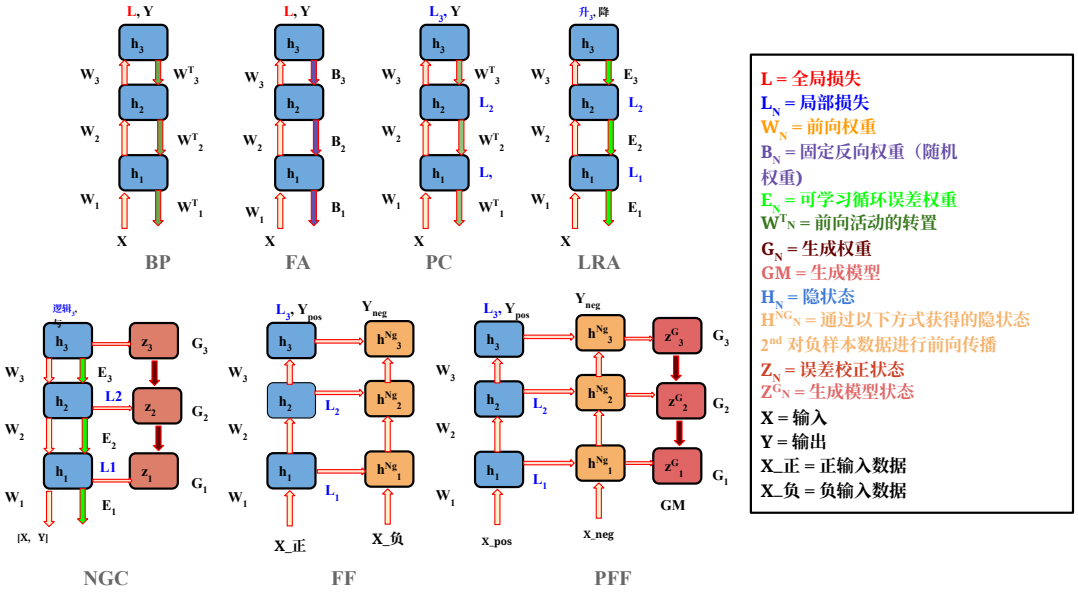


图 1：放宽误差反向传播（BP）所施加约束的信用分配算法比较。图中展示的算法包括反馈对齐（FA）[31], 预测编码（PC） [48, 52], 局部表征对齐（LRA） [45],神经生成编码（NGC） [43, 41], 前向-前向过程（FF） [19], 以及预测前向-前向算法（PFF）。

2 预测性前向-前向学习

我们将设计和研究的脑启发式信用分配过程称为预测式前向-前向（PFF）算法，它是 FF 算法 [19]的一种泛化。在较高层次上，PFF 过程由两个神经结构或电路组成，即表征电路（由 Θ_r 参数化），专注于获取数据样本的分布式表征；以及自上而下生成电路（由 Θ_g 参数化），专注于学习如何根据表征电路的活动值合成数据。因此，PFF 过程可以被描述为一种互补系统，能够联合学习分类器和生成模型。我们将首先定义本文所用的符号，然后介绍表征电路和生成电路的推理与学习机制。

记号： 我们用 $\odot$ 表示哈达玛积，用 $\cdot$ 表示矩阵/向量乘法。$(\mathbf{v})^T$ 是$\mathbf{v}$的转置。矩阵/向量以粗体表示，例如矩阵$\mathbf{M}$ 或向量$\mathbf{v}$ （标量以斜体表示）。$\mathbf{z}^j$ 表示从向量$\mathbf{z}$中提取第 j 个标量。 $<\mathbf{a},\mathbf{b}>$ 表示沿列维度进行向量拼接。最后， $||\mathbf{v}||_2$ 表示向量$\mathbf{v}$的欧几里得范数。感觉输入的形状为$\mathbf{x} \in \mathcal{R}^{J_0 \times 1}$ （J_0 为输入特征的数量），标签的形状为$\mathbf{y} \in \mathcal{R}^{C \times 1}$ （C 为类别数量），神经层的形状为$\mathbf{z}^\ell \in \mathcal{R}^{J_\ell \times 1}$ （J_ℓ 为 ℓ的神经元数量）。

2.1 前向-前向学习规则

PFF 过程与前向-前向算法应用于循环神经网络时的情况非常相似：它通过测量一组神经元的“良好度”，即其活动表明输入信号来自目标训练数据分布（或“正类”）的概率，来调整这组神经元的突触效能。形式上，对于一个包含 L 层的神经系统中的第 ℓ 层，给定神经活动向量$\mathbf{z}^\ell$，我们将良好度计算为活动值的平方和，并通过以下两种方式之一将其与阈值 θ_z 进行比较：

$$p(c=1)_\ell = \frac{1}{1+\exp\left(-(\sum_{j=1}^{J_\ell}(\mathbf{z}_j^\ell)^2 - \theta_z)\right)}, \text{ or,} \tag{1}$$

$$p(c=1)_\ell = \frac{1}{1+\exp\left(-(\theta_z - \sum_{j=1}^{J_\ell}(\mathbf{z}_j^\ell)^2)\right)} \tag{2}$$

其中， $p(c=1)_\ell$ 表示数据来自该数据分布（即正数据，正类的标签为 $c=1$）的概率，而数据不来自训练数据

分布的概率为 $p(c=0)_\ell = 1 - p(c=1)_\ell$。请注意， $p(c)_\ell$ 表示系统/网络中某一层 ℓ 神经元所赋予的概率。这意味着，任何一层试图求解/优化的代价函数，都类似于一个二分类逻辑回归问题，其形式如下：

$$\mathcal{L}(\Theta^\ell) = -\frac{1}{N}\sum_{i=1}^{N} c_i \log p(c_i=1)_\ell + (1-c_i)\log p(c_i=0)_\ell \tag{3}$$

其中，若设计一个生成负样本数据的生成过程，就能正确且自动地生成二元标签 c_i （第 i 个数据点 $\mathbf{x}_i$ 的标签）。从训练集$\mathbf{x}_j \sim \mathcal{D}_{train}$ 中采样的数据模式可标记为 $c_j = 1$，而在 $\mathcal{D}_{train}$ 之外采样的模式（来自负数据生成过程）则可标记为 $c_j = 0$。FF 前向-前向算法流程能否成功，关键在于设计出有用的负数据分布，这与噪声对比估计中的做法大致相同 [16]。

需要注意的是，FF学习规则是局部的——这意味着任何特定神经元层的突触都可以独立于其他层进行调节。该规则的形式不同于经典的赫布更新[17] （后者通过输入和输出神经活动的乘积产生权重变化），因为这种突触调节需要掌握一组神经元的整体信息（良好度取决于一组神经元活动的平方和，而非单个单元的活动），并将对比学习融入动力学中。

突触更新通过计算方程 3 的梯度得出，即 $^{\partial \mathcal{L}}{}_{(\Theta^\ell)\,\partial\Theta^\ell}$。实际上，神经层通过最大化该层的平方活动（使其高于 θ_z，即方程 1）或最小化平方活动（使其低于 θ_z，即方程 2）来优化方程 3。

2.2 表征电路

为了利用 FF学习规则并对上下文预测进行建模，我们提出了一种结构类似于循环神经网络（RNN）的神经电路，其理念与 [19], 中的电路相似：在每一层，自上而下和自下而上的影响相结合，以计算层 ℓ 的活动，这与深度玻尔兹曼机的推断过程 [50]颇为相似。该表征电路的核心参数包含在构造 $\Theta_r = \{\mathbf{W}^1, \mathbf{W}^2, ..., \mathbf{W}^L\}$ 中（称为表征参数）。请注意，我们的模型不包含额外的分类专用参数（不同于 [19] 的模型），不过加入这些参数也很简单。[2]

要计算该表征电路中任意一层的活动状态，需要将自上而下、横向和自下而上的消息与该层在前一时间步的活动状态插值结合起来。具体而言，在 PFF 中，计算方式如下：

$$\begin{aligned}\mathbf{z}^\ell(t) = \beta\Big(\phi^\ell\big(\mathbf{W}^\ell \cdot \text{LN}(\mathbf{z}^{\ell-1}(t-1)) + \mathbf{V}^\ell \cdot \text{LN}(\mathbf{z}^{\ell+1}(t-1)) \\ - \mathbf{L}^\ell \cdot \text{LN}(\mathbf{z}^\ell(t-1)) + \epsilon_r^\ell\big)\Big) + (1-\beta)\mathbf{z}^\ell(t-1)\end{aligned} \tag{4}$$

其中， $\epsilon_r^\ell \sim \mathcal{N}(0, \sigma)$ 是注入的中心化高斯噪声，且 $\mathbf{z}^0(t-1) = \mathbf{x}$。值得注意的是，我们电路的动力学通过矩阵$\mathbf{L}^\ell$直接整合了可学习的横向／交叉抑制和自激；该矩阵还可进一步分解如下：

$$\mathbf{L}^\ell = \text{ReLU}(\hat{\mathbf{L}}^\ell) \odot \mathbf{M}^\ell \odot (1 - \mathbf{I}) + \text{ReLU}(\hat{\mathbf{L}}^\ell) \odot \mathbf{I} \tag{5}$$

其中 $\hat{}\mathbf{L}^\ell \in \mathcal{R}^{J_\ell \times J_\ell}$ 是一个可学习参数矩阵，$\mathbf{M}^\ell \in \{0,1\}^{J_\ell \times J_\ell}$ 是一个二值掩码矩阵，用于强制形成特定的侧向神经竞争模式（详情参见附录），而 $\mathbf{I}^\ell \in \{0,1\}^{J_\ell \times J_\ell}$ 是一个单位矩阵。我们将每一层 ℓ 的激活函数 $\phi^\ell()$ 设为线性整流函数，即 $\phi^\ell(\mathbf{v}) = \max(0, \mathbf{v})$。插值系数 β 控制状态 $\mathbf{z}^\ell$ 随时间的整合（即，时刻 t 的新活动状态是新提出的状态与时刻 $t-1$时状态先前值的凸组合）。请注意，这种插值类似于循环算法 [22], 中引入的“回归”因子。该算法是一种经典的局部学习过程，利用精心设计的自编码器生成计算突触变化所需的信号。$\text{LN}(\mathbf{z})$ 是作用于活动向量 $\mathbf{z}^\ell$的层归一化函数，即，

[2]如果需要分类专用参数，可以加入一组额外的突触权重 $\Theta^d = \{\mathbf{W}, \mathbf{b}\}$，将循环表征电路最顶层的（归一化）活动状态 $\text{LN}(\mathbf{z}^L)$ 作为输入，从而对$\mathbf{y}$上的标签分布进行粗略预测，即 $p(y=i|\text{LN}(\mathbf{z}^L)) = \exp(\mathbf{W} \cdot \text{LN}(\mathbf{z}^L) + \mathbf{b})_i / \left(\sum_c \exp(\mathbf{W} \cdot \text{LN}(\mathbf{z}^L) + \mathbf{b})_c\right)$。这样一来，本文的循环模型就会与 [19] 的模型更加相似。随后，可通过计算目标函数 $\mathcal{L}^y(\mathbf{W}, \mathbf{b}) = -\log\, p(y=i|\text{LN}(\mathbf{z}^L))$ 的相关梯度来调整 Softmax 参数 $\mathbf{W}$ 和 $\mathbf{b}$。

$\text{LN}(\mathbf{z}^\ell) = \mathbf{z}^\ell/(||\mathbf{z}^\ell||_2 + \epsilon)$（其中 ϵ 是一个用于防止除以零的小数值稳定因子）。请注意，表征电路的最顶层被钳制为一个上下文向量$\mathbf{y}$ （它可以是另一个电路的输出，或被钳制为某个数据点的标签），即 $\mathbf{z}^{L+1} = \mathbf{y}$[3], ；而底层被钳制为感觉输入，即 $\mathbf{z}^0(t) = \mathbf{x}(t)$（$\mathbf{x}(t)$ 可以是一个视频帧，也可以是静态图像 $\mathbf{x}$ 的重复副本）。公式 4 描述了所有逐层活动的同步更新，但循环神经网络也可以异步实现 [19],，即首先根据奇数层的活动更新所有偶数层，然后根据偶数层的新值更新奇数层的值，如 [6]的生成式随机网络所采用的方式。

为了创建训练该系统所需的负样本数据，我们忽略正样本数据的标签$\mathbf{y}^{\,p}$ 所指示的当前类别$\mathbf{x}$，并通过随机（均匀）抽取一个不正确的类别索引（排除正确类别），创建一个错误的“负标签”$\mathbf{y}^n$ 。最终输入表征电路的内容，是将正负样本拼接而成，即$\mathbf{x} =< \mathbf{x}, \mathbf{x} >$（请注意，标签可用时，正样本的图像像素会被重复用作负样本数据），$\mathbf{y} =< \mathbf{y}, \mathbf{y}^n >$，以及$\mathbf{c} =< 1, 0 >$。
创建（$\mathbf{x}$,$\mathbf{y}$,$\mathbf{c}$）后，公式 4 会运行多次（$T = 8$ 至 $\mathbf{10}$），类似于预测编码系统中模拟的刺激处理窗口 [48, 41]。每次运行公式 4 时，都会根据以下局部更新规则计算第ℓ层突触的更新量：

$$\Delta \mathbf{W}^\ell = \Big(2\frac{\partial \mathcal{L}(\Theta^\ell)}{\partial \sum_j^{J_\ell} (\mathbf{z}_j^\ell)^2} \odot \mathbf{z}^\ell\Big) \cdot \big(\text{LN}(\mathbf{z}^{\ell-1})\big)^T, \text{ and,} \tag{6}$$

$$\Delta \mathbf{V}^\ell = \Big(2\frac{\partial \mathcal{L}(\Theta^\ell)}{\partial \sum_j^{J_\ell} (\mathbf{z}_j^\ell)^2} \odot \mathbf{z}^\ell\Big) \cdot \big(\text{LN}(\mathbf{z}^{\ell+1})\big)^T \tag{7}$$

$$\Delta \hat{\mathbf{L}}^\ell = \Big(\Big(2\frac{\partial \mathcal{L}(\Theta_r^\ell)}{\partial \sum_j^{J_\ell} (\mathbf{z}_j^\ell)^2} \odot \mathbf{z}^\ell\Big) \cdot \big(\text{LN}(\mathbf{z}^{\ell})\big)^T\Big) \odot \frac{\partial \mathbf{L}^\ell}{\partial \hat{\mathbf{L}}^\ell} \tag{8}$$

随后可通过随机梯度下降（随机梯度下降）并结合动量或 Adam 等方法应用这些更新，以调整相关参数，即$\mathbf{W}^\ell$ 和 $\mathbf{V}^\ell$ [25]。原则上，表征电路的神经层会全局优化目标函数 $\mathcal{L}(\Theta_r) = \sum_{\ell=1}^L \mathcal{L}(\Theta^\ell = \mathbf{W}^\ell)$（即局部优良度函数之和）。

关于感官模式的分类：可以注意到，我们的表征电路并不包含用于直接分类输入的判别性参数。然而，鉴于监督目标$\mathbf{y}$ （针对正数据样本）被用作上下文，以调节上述循环电路最顶层的潜在表征，该表征系统应当习得能够隐式编码标签信息的分布式表征。为了利用 PFF 的表征中编码的判别信息，正如 FF 算法中所采用的方法，我们可以通过执行类似于混合玻尔兹曼机模型 [28, 46]的推断过程来进行分类。

为了对输入$\mathbf{x}$进行分类，我们遍历$\mathbf{y}$ 可能取的所有（独热）值，从第一个类别索引开始。具体而言，对于任意选定的$\mathbf{y}$ （例如类别索引 i 的独热编码），我们在表征电路上运行公式 4 T步，然后记录中间三次迭代（从 $T/2-1$到 $T/2+1$）中各层的全局优度，即 $\mathcal{G}_{y=i} = \frac{1}{3}\sum_{T/2-1}^{T/2+1} \frac{1}{L}\sum_{\ell=1}^L \theta_z - \sum_j^{J_\ell}(\mathbf{z}_\mathbf{j}^{\ell 2})$。对于所有类别索引 $y = 1$、2、.....、C，均进行这一优度计算，从而得到 $\{\mathcal{G}_{y=1}, \mathcal{G}_{y=2}, ..., \mathcal{G}_{y=C}\}$，并对其应用最大值索引，以获得平均优度值最高的类别索引。注意，如 [19], 中所述，如果 PFF 的表征电路中包含分类专用参数，则可以使用前馈过程来获取初始类别概率。该过程可以简化分类流程，只需为概率最高的前 M 个类别计算优度 $\mathcal{G}_{y=i}$ （从而避开在海量类别上进行代价高昂的搜索）。为了估计表征电路下的标签概率分布，正如本文所做的那样，我们将优度（logits）输入 Softmax，即 $p(y = i|\mathbf{x}) \sim \exp(\mathcal{G}_i)/(\sum_c \exp(\mathcal{G}_c))$。

2.3 生成电路

如前所述，PFF 算法纳入了一个自上而下定向的生成模型的联合适应。PFF 过程的这一方面，受到预测处理（PP）模型的生成性质的启发 [48, 13],，尤其是那些采用神经生成

[3]将标签／上下文向量缩放约 5 倍非常重要，即最顶层的活动应为 $\mathbf{z}^{L+1} = \mathbf{y} * 5$（Geoffrey Hinton，个人通信，2022年12月12日）。[4]这不同于 FF 算法中的负标签生成， [19],后者根据分类专用权重前向传播所产生的概率，按比例选择一个错误的类别索引。部分前向传播过程不需要这样做。

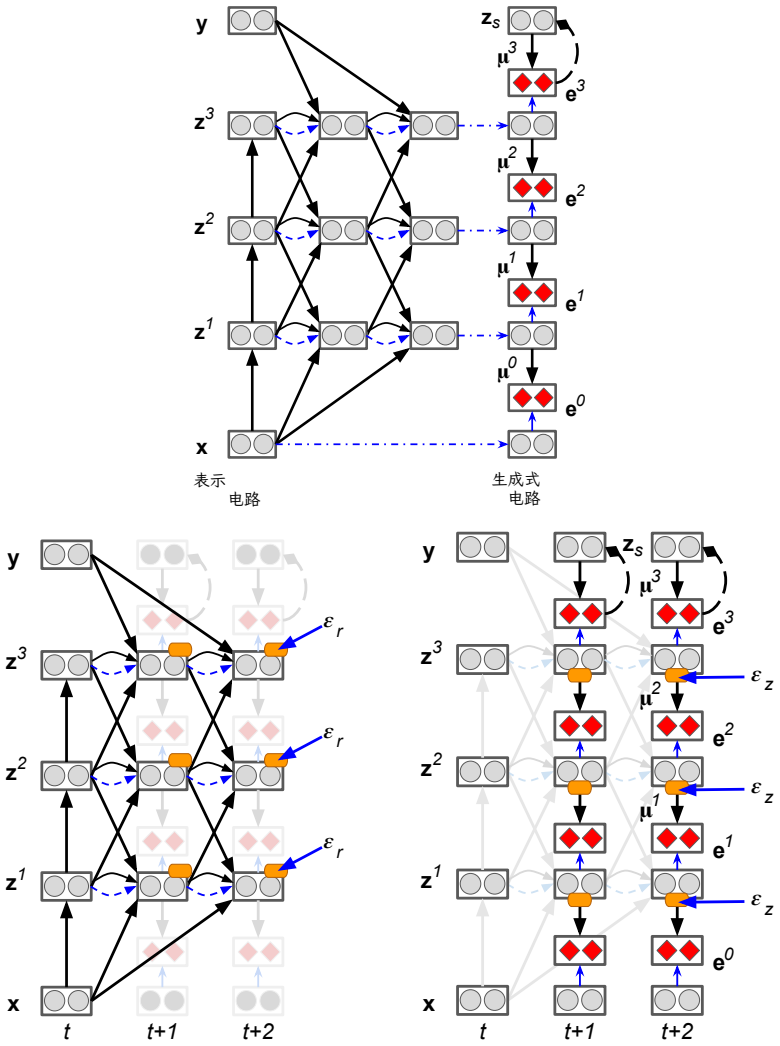


图2：（上）展示了一个包含三个隐藏层的表征电路与一个四层生成电路耦合时，PFF 算法在三个时间步内的运行过程（$\mathbf{z}_s$ 是其潜变量）。实线箭头表示突触，点划线箭头表示状态延续（直接复制）。虚线菱形曲线表示反馈通路，灰色圆圈表示神经元，红色菱形表示误差神经元。注意：由于系统的所有元素都会动态调整，因此生成电路会与表征电路同时运行／更新。（左下）展示了表征电路如何随时间迭代形成潜在表征（$\mathbf{z}^\ell$）。（右下）展示了生成电路如何迭代地对表征电路的逐层活动进行局部预测（μ^ℓ）。请注意，这两个过程同时发生，并且每个时间步都会向每一层注入噪声（蓝色箭头表示注入表征噪声 ϵ_r^ℓ 或生成噪声 ϵ_z^ℓ）。

编码框架、专注于学习自上而下生成模型的模型 [41]。至关重要的是，我们强调，以生物学上合理的方式联合学习一个生成式反馈系统，或许能够为检查 FF 核心过程所获取表征的内容提供一种有效手段，并为（内部）合成负样本数据提供一种可行的替代方案。

生成电路由一组突触参数 $\Theta_g = \{\mathbf{G}^1, \mathbf{G}^2, ..., \mathbf{G}^L, \mathbf{G}^{L+1}\}$ 构成，旨在学习如何在每一层预测局部区域的神经活动。正如我们将看到的，这种设计有助于进行简单的误差赫布更新（与预测处理系统中计算出的更新非常相似）。形式上，对于单个数据点，生成电路将尝试优化的目标函数为：

$$\mathcal{L}_g(\Theta_g) = \sum_{\ell=0}^{L} \mathcal{L}_g^\ell(\mathbf{G}^{\ell+1}) = \sum_{\ell=0}^{L} \sum_{j=1}^{J_\ell} (\bar{\mathbf{z}}_j^\ell - \mathbf{z}_j^\ell(t))^2 \tag{9}$$

其中 $\mathbf{z}^0 = \mathbf{x}$ （底层目标被钳制为正在处理的数据点）。生成电路的每一层都执行以下计算：

$$\bar{\mathbf{z}}^\ell = g^\ell(\mathbf{G}^{\ell+1} \cdot \text{LN}(\widehat{\mathbf{z}}^{\ell+1})), \tag{10}$$

$$\text{where, } \widehat{\mathbf{z}}^{\ell+1} = \phi^{\ell+1}(\mathbf{z}^{\ell+1}(t) + \epsilon_z^{\ell+1}) \text{ and, } \mathbf{e}^\ell = \bar{\mathbf{z}}^\ell - \mathbf{z}^\ell(t) \tag{11}$$

$$\bar{\mathbf{z}}^L = g^L(\mathbf{G}^{L+1} \cdot \text{LN}(\mathbf{z}_s)), \quad \text{// Topmost layer } \mathbf{z}_s \tag{12}$$

$$\text{where, } \mathbf{z}_s \leftarrow \mathbf{z}_s - \gamma \frac{\partial \mathcal{L}_g^L(\mathbf{G}^{L+1})}{\partial \mathbf{z}_s} \tag{13}$$

其中， $\epsilon_z^\ell \sim \mathcal{N}(0, \sigma_z)$ 是注入层 ℓ 的受控（加性）活动噪声（尺度较小，即 $\sigma_z = 0.025$）。$g^\ell()$ 是应用于生成层预测的逐元素激活函数。在本研究中，除最底层设为截断恒等函数外，层 $\ell >= 1$ 均采用线性整流函数；截断恒等函数即 $g^0(\mathbf{v}) = \text{HardClip}(\mathbf{v}, 0, 1)$。在“表征电路”一节所述的每一步推理中，生成模型的突触（各层）都通过以下赫布规则进行调整：

$$\Delta \mathbf{G}^\ell = \mathbf{e}^{\ell-1} \cdot \big(\text{LN}(\mathbf{z}^\ell(t))\big)^T, \text{ and, } \Delta \mathbf{G}^{L+1} = \mathbf{e}^L \cdot \big(\text{LN}(\mathbf{z}_s)\big)^T. \tag{14}$$

请注意，生成电路的最顶层（即层 $L+1$）与其他层的处理略有不同：最高层 $\mathbf{z}_s$ 预测表征电路的最顶层神经活动 $\mathbf{z}^L$，随后通过迭代推理反馈方案进行调整，这与稀疏编码/预测编码 [39, 48, 41] 非常相似。训练完成后，通过祖先采样即可从生成电路合成数据：

$$\bar{\mathbf{z}}^{L+1} = \mathbf{z}_s \sim P(\mathbf{z}_s), \text{ and, } \bar{\mathbf{z}}^\ell = g^\ell(\mathbf{G}^{\ell+1} \cdot \text{LN}(\bar{\mathbf{z}}^{\ell+1})), \ \ell = L, (L-1), ..., 0 \tag{15}$$

其中，我们选择先验 P（$\mathbf{z}_s$）为包含 10 个成分的高斯混合模型（GMM）。在本研究中，该模型根据训练数据集$\mathcal{D}_{train}$ 通过模型运行所获得的训练系统最顶层活动样本进行回溯拟合，这与 [41]的自上而下定向生成式预测处理模型所采用的方法相同。请注意，对于 PFF 中的所有电路（包括表示电路和生成电路），我们将线性整流函数激活函数的导数视为与层活动 $\mathbf{z}^\ell$ 形状相同的全 1 向量（如 [19]中所采用的方法）。刻画 PFF 的两个关键神经回路如图2所示，PFF 过程的详细伪代码（算法1）见附录。

与对比赫布学习的关系： 在设计网络时（如上文所述），人们可能会注意到，其推断过程与通过对比赫布学习（CHL） [36], 学习得到的神经系统的推断过程非常相似，尽管二者之间存在若干显著差异。基于 CHL 的神经系统中的逐层活动根据以下动力学方程进行更新：

$$\mathbf{z}^\ell(t) = \mathbf{z}^\ell(t-1) + \beta(-\mathbf{z}^\ell(t-1) + \mathbf{m}^\ell) \tag{16}$$

$$\mathbf{m}^\ell = \phi^\ell\Big(\mathbf{W}^\ell \cdot \mathbf{z}^{\ell-1}(t-1) + (\mathbf{W}^{\ell+1})^T \cdot \mathbf{z}^{\ell+1}(t-1)\Big) \tag{17}$$

此处可以注意到，动力学不涉及归一化，并且 layer ℓ 的数值与公式 4 中的积分方式略有不同；也就是说，神经元数值通过泄漏欧拉积分发生变化，其中自上而下和自下而上的传递相结合，对该层产生扰动，而不是为状态本身提出一个新值。

与 CHL 类似，前向-前向算法和 PFF 需要两个阶段（或两种计算模式），在这两个阶段中，沿神经系统传播的信号将彼此进行对比。给定样本（$\mathbf{x}$，$\mathbf{y}$），CHL 首先在未夹紧阶段（负相）运行系统，此时只有输入图像 $\mathbf{x}$ 被夹紧到感觉输入 / 底层；随后进入夹紧阶段，此时 $\mathbf{x}$ 及其目标 $\mathbf{y}$ 均被夹紧，即将 $\mathbf{y}$ 夹紧到输出层（正相）。在每个阶段（或推理周期）结束时，记录逐层活动，并将其用于减法式赫布规则，以计算每个突触矩阵的更新。需要注意的是，CHL 的正相依赖于先运行负相。相比之下，前向-前向算法和 PFF 相当于并行运行正相与负相（每个阶段由不同的数据驱动），从而使整体处理时间更短（不再是一个推理周期的状态取决于另一个推理周期的统计特性，而是同时针对正样本数据或负样本数据运行相同的周期，并采用相反的目标 [19], ）。

与预测编码的关系： PFF 算法通过利用循环表征电路迭代处理窗口中获得的表征，将预测编码（PC）中的局部假设生成思想融入推断过程。具体而言，在每个时间步，表征电路的每一层都会成为生成电路各层的预测目标。相比之下，PC 模型必须利用一组反馈突触，逐步修改各层的活动，最后才调整突触值。此外，PFF 会在每个处理时间步内动态修改突触，而；

学习 算法	BP	FA	PC	LRA	NGC	FF	PFF
前向锁定	全局	全局	局部	局部	局部	*None*	*None*
前向误差	⛔	⛔	⛔	⛔	⛔	✅	✅
前向目标	⛔	⛔	⛔	⛔	⛔	✅	✅
后向锁定	全局	全局	*None*	*None*	*None*	*None*	*None*
反向误差	⛔	⛔	✅	✅	✅	⛔	⛔
反向目标	⛔	⛔	✅	✅	✅	⛔	⛔
局部损失	⛔	⛔	✅	✅	✅	✅	✅
误差突触	⛔	固定	⛔	学习得到的	学习得到的	⛔	⛔
全局信号	✅	✅	⛔	✅	⛔	⛔	⛔
局部信号	⛔	⛔	✅	✅	✅	✅	✅
生成式 能力	⛔	⛔	⛔	⛔	✅	⛔	✅
生成式 权重	⛔	⛔	⛔	⛔	✅	⛔	✅

图 3：不同学习算法的性质，即反向传播（BP）、反馈对齐（FA）、预测编码（PC）、局部表征对齐（LRA）、神经生成编码（NGC）、前向-前向算法（FF）和预测前向-前向过程（PFF）。

通常，PC 电路会实现某种形式的期望最大化，因此需要更长的刺激处理窗口，才能学习有效的生成模型 [41]（在本研究中，PFF 生成电路仅用 8-10 步便学会了高质量的生成模型，而 [41] 的模型至少需要 50 步）。

与局部学习的关系： 长期以来，人们一直认为，大脑中的突触可能会按照一种局部方案进行调整，也就是说，只有在空间和时间上最接近目标突触的信息才会参与计算其效能变化。遵循这一生物学约束的方法被称为局部学习过程 [45, 30, 37, 27, 38, 5, 11, 26, 24], ，它们有望取代反向传播来训练深度网络，并放宽反向传播的一项或多项核心约束（有关各类算法中主要约束的比较，请参见图 3）。更值得期待的是，已有实证表明，局部方案的更新能够带来更好的泛化 [30, 45], ，即使处理时序数据 [33] 和离散信号 [34]时也是如此。人们为设计生物学上合理的局部学习算法做出了许多努力，例如对比赫布学习（前文提及） [36], 、用于学习调和机（受限玻尔兹曼机）的对比散度 [20],、用于学习亥姆霍兹机的清醒-睡眠算法 [21], ，以及平衡传播等算法[54]。另一些研究则将局部学习直接整合进深度学习流程，例如反冲 [2]和解耦神经接口 [23]。值得指出的是，PFF 与清醒-睡眠算法有相似之处，后者需要将生成模型与推理（识别）模型联合学习。然而，清醒-睡眠算法存在不稳定性：生成网络产生的随机幻想可能会损害识别网络，而推理网络较低的表征能力也可能妨碍生成网络（因此出现了重加权醒睡算法等变体 [7]）。相比之下，PFF 旨在基于表征电路来学习生成模型，并以局部适应的神经活动指导合成过程，而不是随机采样生成模型来创建识别网络的教学信号（这可能会用毫无意义的信号干扰其优化）。

3 实验

本节介绍了为测试所提出的部分前向传播过程的能力而设计的模拟/实验。我们利用多个图像数据集，定量评估 PFF 的分类能力（以测试集误差衡量），并定性评估其生成能力（通过目视检查样本重建和

表1：采用不同学习算法训练的系统的分类泛化结果（4-KNN 除外，它是非参数学习基线模型）。测量结果以五次实验试验轮次的均值和标准差表示。

模型	MNIST 测试误差(%)	K-MNIST 测试误差(%)	F-MNIST 测试误差(%)	N-MNIST 测试误差 (%)	Et-MNIST 测试误差（%）
4-KNN	2.860 ± 0.000	7.900 ± 0.000	14.030 ± 0.00	7.720 ± 0.000	2.120 ± 0.000
随机前馈神经网络	3.070 ± 0.018	14.070 ± 0.189	17.100 ± 0.102	9.916 ± 0.135	1.650 ± 0.040
反向传播前馈神经网络	$\mathbf{1.300 \pm 0.023}$	6.340 ± 0.202	10.720 ± 0.014	5.574 ± 0.09	$\mathbf{0.392 \pm 0.02}$
前向传播循环神经网络	1.320 ± 0.100	6.590 ± 0.420	10.750 ± 0.060	5.380 ± 0.040	0.470 ± 0.050
PFF-RNN	1.340 ± 0.010	$\mathbf{6.250 \pm 0.060}$	$\mathbf{10.400 \pm 0.010}$	$\mathbf{5.270 \pm 0.101}$	0.451 ± 0.070

模式合成）。我们将部分前向传播过程（PFF-RNN）与若干基线方法进行比较，包括 K近邻算法（使用 $K = 4$，即 4-KNN）、采用原始前向-前向算法[19] 训练的原始循环网络（FF-RNN），以及两种基于反向传播的模型：一种是使用反向传播调整所有内部突触的前馈神经网络（反向传播前馈神经网络），另一种是相同的网络，但只调整最顶层 Softmax输出层的参数，并固定隐藏层突触参数（Rnd-FNN）。这两种基于反向传播的网络都经过训练，以最小化各数据集所提供标签的分类交叉熵。部分训练模型，即 Rnd-FNN，可视为神经系统泛化能力的一种下限，因为仅凭随机隐藏特征检测器也能获得可观的分类性能（请注意，神经信用分配算法的表现不应低于这一水平）。

数据集：我们使用若干灰度图像集合进行实验，即 MNIST、Kuzushiji-MNIST（K-MNIST）、Fashion MNIST（F-MNIST）、Not-MNIST（N-MNIST）和埃塞俄比亚文（Et-MNIST）数据库。所有数据库均包含来自 10 个不同类别的 28×28 张图像（详情见附录）。

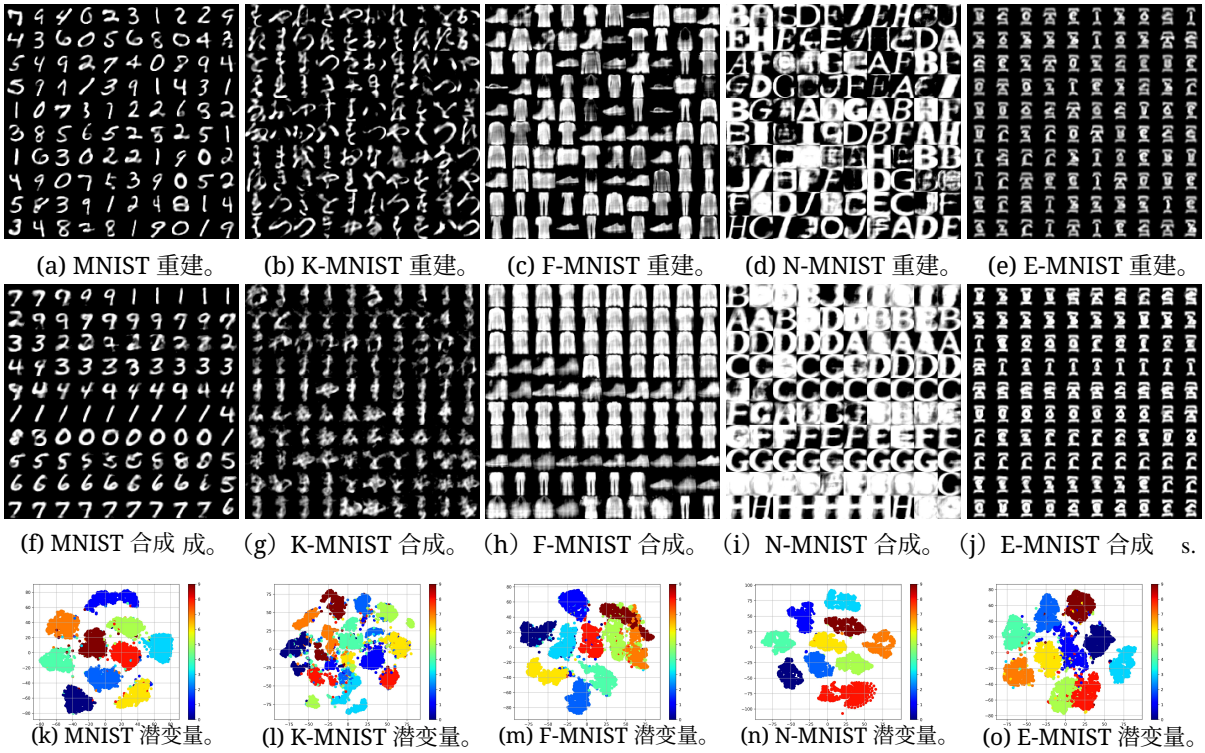
仿真设置：为确保比较公平，所有仿真模型均采用相似的架构。所有神经网络学习算法使用的网络均包含两个隐藏层，每层有 2000 个神经元（与 [19]中的前向-前向算法模型十分相似），初始突触权重值按照随机正交初始化方案 [53]（使用奇异值分解）选取。任何算法计算出突触的调整值后，均使用 Adam自适应学习率 [25]，通过每批包含 500 个样本的小批次来调整参数。前向-前向算法和 PFF 均采用阈值 $\theta_z = 10.0$；PFF 则使用 20 个潜变量（即 $\mathbf{z}_s \in \mathcal{R}^{20\times 1}$）、表征噪声 $\sigma^\ell = 0.05$和生成噪声 $\sigma_z = 0.025$。

讨论： 请注意表1：在本研究模拟的模型中，PFF 过程表现良好，在 MNIST 上取得了约 1.34% 的优异分类错误率，接近经过良好调优的分类器反向传播前馈神经网络（1.30%）。值得注意的是，PFF-RNN 模型在 3 个（可以说更难的）数据集上优于反向传播前馈神经网络。前向-前向算法和 PFF 均优于性能下限基准，即 4-KNN 和随机前馈神经网络，这表明它们学到了有助于实现良好判别性能的特征检测器。

从定性角度来看，请观察图 4（顶行）：PFF 学到了优质的图像输入重建模型和生成模型。重建出的符号（例如数字、汉字）效果出色，而且所有情况下的图像样本在各类别之间都呈现出一定的多样性（尽管略显模糊）。请注意，如前文“生成电路”一节所述，为了从 PFF 的有向生成模型中采样，我们对潜变量 $\mathbf{z}_s$ 的样本拟合了一个高斯混合模型，并通过期望最大化对该高斯混合模型（含 10 个分量）进行优化。最后，在图 4 底行，我们展示了由 PFF 生成电路最顶层活动层所诱导的潜在空间，可以看到所有情况下都理想地形成了以类别为中心的簇。在附录中，我们直观地检查了 PFF 电路中（最靠近感觉输入层的）突触的感受野，并详细讨论了部分前向传播过程带来的计算优势。

预测式前向-前向算法的计算优势：从硬件效率的角度来看，PFF 算法与 FF 过程[5], 很相似，都是在模拟和神经形态硬件上实现的潜在理想候选方案。之所以有这种可能，是因为 FF 和 PFF 只需进行前向传播[6] 即可完成推理和突触调节；这些算法不需要用于传递教学信号的独立计算路径（反向传播 [49] 和反馈对齐方法 [31, 37]都需要），甚至也不需要用于传递误差信息的路径（预测编码 [48, 4, 41], 、表征对齐 [45], 和目标传播过程 [30] 都需要）。更可取的是，这意味着无需专用硬件来计算导数（而计算激活函数需要导数——这意味着，即使使用离散且不可微的随机函数来实现 PFF，例如对伯努利分布进行采样，也能.....

[5]我们指出，这同样适用于其他仅前向学习算法 [27, 26]。[6]每个模拟时间步中，前向-前向算法需要进行两次前向传播，而 PFF 需要进行三次前向传播。



(a) MNIST 重建。 (b) K-MNIST 重建。 (c) F-MNIST 重建。 (d) N-MNIST 重建。 (e) E-MNIST 重建。

(f) MNIST 合成 成。 (g) K-MNIST 合成。 (h) F-MNIST 合成。 (i) N-MNIST 合成。 (j) E-MNIST 合成 s.

(k) MNIST 潜变量。 (l) K-MNIST 潜变量。 (m) F-MNIST 潜变量。 (n) N-MNIST 潜变量。 (o) E-MNIST 潜变量。

图 4：模型针对每个自然图像数据库重建的模式（“重建”，顶行）和生成的样本（“合成”，中行）。底行展示了 PFF-RNN 针对每个数据集的生成电路中最顶层潜变量活动层的可视化结果。

（可能/可行），也不适用于在内存中维护和调整独立的反馈传递突触（它们通常需要不同的调整规则 [45, 42]）。

要对 PFF 和前向-前向算法进行适当改写，使其完全兼容神经形态／模拟芯片，仍需付出努力，因为这两种方法仍使用矩阵乘法来模拟神经区域之间的信息传递。相比之下，脉冲神经系统（适用于在此类硬件上进行模拟）要求将活动表示为电压（与脉冲响应模型耦合），并将突触表示为电导，从而允许神经电荷（按单位时间计）累加。不过，可以采用类似脉冲神经编码 [40] 的方法，将前向-前向算法／PFF 转变为天然利用脉冲序列固有稀疏计算的形式。然而，由于前向-前向算法和 PFF 都是仅进行前向传递的信用分配算法，因此不再需要模数转换过程来支持反向传递（反向传播所需），以计算数值梯度。此外，PFF 表征电路中的侧向突触权重进一步带来了结构稀疏性，因此在直接支持稀疏计算的硬件上实现这些权重尤其有用。最后，正如正文所述，PFF 比基于对比赫布学习设置的仅前向算法更快 [36, 20, 54],，因为它不需要让负相计算以正相的统计特性为条件（反之亦然）。

交织的推理与学习与不朽计算：另一个重要观点是，PFF 算法及其未来的同类算法提供了一种计算框架，称为 [19] “不朽计算”，也就是软件与硬件不再泾渭分明、彼此分离的系统。从某种意义上说，不朽计算的理念与交织的推理与学习 [47, 45, 42, 41],一脉相承。后者指出，大脑中的神经生物学学习与推理并非两个截然不同、彼此独立的过程，而是相辅相成、相互依赖的过程，甚至依赖于执行这些过程的神经回路本身的结构。交织的推理与学习以预测编码／处理和对比赫布学习为典型代表，意味着没有推断过程，就不会发生突触调节；而没有突触调节，推理也就失去了意义（因为知识无法编码到任何形式的长期记忆中）。不朽计算可视为交织的推理与学习在逻辑上的自然延伸与结果，因为模拟突触权重／数值中编码的知识（以及它们最终执行／支持的计算）会随着硬件故障／损坏而“消亡”或消失。

接受不朽计算这一理念，意味着必须付出这样的代价：神经计算的计算程序／模拟不再能够轻易复制到数百万台计算机／设备上（以利用高性能计算系统）。但这可能大幅降低能源消耗，而能源消耗日益成为人工智能（AI）研究 [1, 8, 58, 55], 关注的问题；也就是说，我们如何避免设计出计算成本和碳排放迅速攀升的智能系统，或转而采用红色人工智能 [55], 以外的绿色人工智能方法。归根结底，这意味着，设计交织的学习与推理方法时，若能考虑这些方法在特定物理结构中的实现方式，例如硬件的形态与设计，将对未来的统计学习研究极具价值；这样便有望运行包含数万亿个突触的大型神经系统，同时仅消耗数瓦电力。值得注意的是，这或许会促成这样一种神经系统的设计：它们能够感知自身所运行硬件的状态，并据此进行调整，从而模拟界定生物学习与推理的稳态约束。然而，要设计出能在具体细节大多未知的硬件上高效运行的算法，仍有大量工作要做；研究 PFF 和 FF 等过程如何扩展至更大型的神经系统，可能是实现这一目标的重要一步。

关于自生成负样本： PFF 最重要的要素之一，是其整合的、联合学习的预测/生成神经回路。正如我们在正文中所展示的，这个生成模型不仅能够高质量地重建原始输入模式，还能通过从其潜在先验 $P(\mathbf{z}_s)$ 中采样来合成数据。由于最顶层神经活动所诱导的潜在空间具有高度多模态特性，我们将该先验设计为高斯混合模型。未来一个重要的研究方向，是运用基于 FF 的生物信用分配算法，考察不同的正相与负相调度方式，并与 [19] 及本研究中采用的同步方式进行比较。具体而言，一个亟待回答的重要问题是：正相与负相在多大程度上可以分离，同时仍能促进神经回路突触的稳定有效局部适应？我们以及未来的其他研究应深入探索这一问题，并利用 PFF 的联合生成/预测电路；一个可行的起点是扩展生成电路，使其以标签向量 $\mathbf{y}$ 为条件，并在训练的某些阶段运行该电路（例如，处理完 M 数据样本或批次后），以合成若干批次的模式，再将这些模式错误地映射到明显错误的标签输入上（即，选择$\mathbf{y}$ 以获得某个特定类别的样本，然后有意自动选取一个错误标签，同时知晓最初选定的标签）。

设计上述方案也有助于开发有趣的持续学习自监督方法，使 PFF 系统能够利用某种内部自我诱发的（记忆）重放来更新其关于较早数据点的知识，从而有望为应对人工神经网络面临的重大挑战——灾难性遗忘——提供一种高效手段。如 [19], 所述，如果能够将正相与负相分开，或使二者之间的间隔更长，就可能通过在一段时间内取消负相更新来建模或研究“严重睡眠剥夺”的影响。最后，从理论角度来看，部分前向传播过程中的信用分配对于计算神经科学、《认知科学》和统计学习均有启示，并为建立统一的理论框架、同时涵盖前向-前向学习与预测处理，提供了重要的起点。

4 结论

在本研究中，我们提出了预测式前向-前向（PFF）过程，用于动态调整循环神经系统的突触效能，使其无需进行误差反向传播，便能联合学习对数据样本进行分类、重建和合成。我们的模型与信用分配过程整合了前向-前向算法的若干要素，例如基于良好度和对比学习的局部突触适应规则；同时还融合了一种新颖而简单的侧向突触竞争形式，以及预测编码的部分特征，例如以局部误差的赫布式方式调整生成性突触权重。我们的结果表明，PFF 学习过程为神经系统提供了一种颇具前景、受脑启发、仅向前传播且无需反向传播的信用分配形式。PFF 不仅有望用于构建在生物学上更忠实的神经模型，也可作为基于神经的认知架构的候选电路/模块，例如认知神经生成系统（CogNGen） [44], ，这类系统旨在解释高层认知功能如何从低层神经细胞活动中涌现。

参考文献

[1] AMODEI, D., AND HERNANDEZ, D. 人工智能与计算。https://openai.com/blog/ai-and-compute/，2018年5月。

[2] BALDUZZI, D., VANCHINATHAN, H., AND BUHMANN, J. M. Kickback cuts backprop’s red-tape: 神经网络中生物学上合理的信用分配。载于*AAAI* (2015)，第 485–491 页。

[3] BARTUNOV, S., SANTORO, A., RICHARDS, B., MARRIS, L., HINTON, G. E., AND LILLICRAP, T. 评估生物学启发的深度学习算法及架构的可扩展性。载于《神经信息处理系统进展》(2018)，第 9390–9400 页。[4] BASTOS, A. M., USREY, W. M., ADAMS, R. A., MANGUN, G. R., FRIES, P., AND FRISTON, K. J. 用于预测编码的典型微回路。《神经元》76, 4 (2012)，695–711。[5] BELILOVSKY, E., EICKENBERG, M., AND OYALLON, E. 卷积神经网络的解耦贪心学习。载于国际机器学习会议(2020)，机器学习研究论文集，第 736–745 页。[6] BENGIO, Y., LAUFER, E., ALAIN, G., AND YOSINSKI, J. 可通过反向传播训练的深度生成随机网络。载于国际机器学习会议(2014)，机器学习研究论文集，第 226–234 页。[7] BORNSCHEIN, J., AND BENGIO, Y. 重加权醒睡算法。arXiv预印本 arXiv:1406.2751 (2014)。[8] BREVINI, B. 黑箱并非绿色：神话化人工智能并忽略环境。《大数据与社会》7, 2 (2020)，2053951720935141。[9] CLANUWAT, T., BOBER-IRIZAR, M., KITAMOTO, A., LAMB, A., YAMAMOTO, K., AND HA, D. 面向日本古典文学的深度学习，2018。[10] CRICK, F. 关于神经网络的最新热潮。《自然》337, 6203 (1989)，129–132。[11] DELLAFERRERA, G., AND KREIMAN, G. 误差驱动的输入调制：无需反向传播即可解决信用分配问题。载于国际机器学习会议(2022)，机器学习研究论文集，第 4937–4955 页。[12] FLORIDI, L., AND CHIRIATTI, M. GPT-3：其性质、范围、局限与影响。《心灵与机器》30, 4 (2020)，681–694。[13] FRISTON, K. 自由能原理：一种统一脑理论？《自然综述：神经科学》11, 2 (2010)，127–138。[14] GARDNER, D. 神经网络的神经生物学。麻省理工学院出版社，1993年。[15] GROSSBERG, S. 竞争学习：从交互激活到自适应共振。《认知科学》11, 1 (1987)，23–63。[16] GUTMANN, M., AND HYVÄRINEN, A. 噪声对比估计：一种针对未归一化统计模型的新估计原理。载于第十三届国际人工智能与统计会议论文集(2010)，JMLR研讨会与会议论文集，第 297–304 页。[17] HEBB, DONALD O, E. A. 《行为组织》，1949年。[18] HELMHOLTZ, H. V. 《生理光学论著》，3 卷。论著(1924年)。[19]HINTON, G. 前向-前向算法：一些初步研究。https://www.cs.toronto.edu/ hinton/FFA13.pdf (2022)。[20] HINTON, G. E. 通过最小化对比散度训练专家乘积模型。《神经计算》14, 8 (2002)，1771–1800。[21] HINTON, G. E., DAYAN, P., FREY, B. J., AND NEAL, R. M. 无监督神经网络的“醒睡”算法。《科学》268, 5214 (1995)，1158–1161。[22] HINTON, G. E., AND MCCLELLAND, J. L. 通过循环传播学习表征。载于神经信息处理系统(1988)，第 358–366 页。[23] JADERBERG, M., CZARNECKI, W. M., OSINDERO, S., VINYALS, O., GRAVES, A., AND KAVUKCUOGLU, K. 使用合成梯度的解耦神经接口。arXiv预印本 arXiv:1608.05343 (2016)。[24] KAISER, J., MOSTAFA, H., AND NEFTCI, E. 深度连续局部学习的突触可塑性动力学（DECOLLE）。《神经科学前沿》14 (2020)，424。[25] KINGMA, D., AND BA, J. Adam：一种随机优化方法。arXiv预印本 arXiv:1412.6980 (2014)。[26] KOHAN, A., RIETMAN, E. A., AND SIEGELMANN, H. T. 信号传播：前向传播中学习与推理的框架。《IEEE神经网络与学习系统汇刊》(2023)。[27] KOHAN, A. A., RIETMAN, E. A., AND SIEGELMANN, H. T. 误差前向传播：重用前馈连接，在深度学习中传播误差。arXiv预印本 arXiv:1808.03357 (2018)。[28] LAROCHELLE, H., AND BENGIO, Y. 使用判别式受限玻尔兹曼机进行分类。载于第 25 届国际机器学习会议论文集(2008)，第 536–543 页。

[29] LECUN, Y. 手写数字MNIST数据库。http://yann. lecun. com/exdb/mnist/ (1998)。[30] LEE, D.-H., ZHANG, S., FISCHER, A., AND BENGIO, Y. 差分目标传播。载于机器学习与数据库知识发现联合欧洲会议（2015），施普林格，第498–515页。[31] LILLICRAP, T. P., COWNDEN, D., TWEED, D. B., AND AKERMAN, C. J. 随机反馈权重支持深度神经网络中的学习。arXiv预印本 arXiv:1411.0247（2014）。[32] LINNAINMAA, S. 将算法累积舍入误差表示为局部舍入误差的泰勒展开。博士论文，硕士论文（芬兰语），赫尔辛基大学，1970年。[33] MALI, A., ORORBIA, A., KIFER, D., AND GILES, L. 研究使用循环神经网络进行动态计数时的反向传播替代方法。载于国际语法推断会议（2021），机器学习研究论文集，第154–175页。[34] MALI, A., ORORBIA, A. G., KIFER, D., AND GILES, C. L. 神经有损图像压缩系统中循环学习算法的实证分析。载于2021年数据压缩会议（DCC）（2021），IEEE，第356–356页。[35] MARBLESTONE, A. H., WAYNE, G., AND KORDING, K. P. 迈向深度学习与神经科学的融合。《计算神经科学前沿》（2016），94。[36] MOVELLAN, J. R. 连续霍普菲尔德模型中的对比赫布学习。载于《连接主义模型》。Elsevier，1991，第10–17页。[37] NØKLAND, A. 直接反馈对齐使深度神经网络能够学习。载于《神经信息处理系统进展》（2016），第1037–1045页。[38] NØKLAND, A., AND EIDNES, L. H. 使用局部误差信号训练神经网络。载于第36届国际机器学习会议论文集（2019年6月9–15日），K. Chaudhuri 和 R. Salakhutdinov 编，第97卷《机器学习研究论文集》，机器学习研究论文集，第4839–4850页。[39] OLSHAUSEN, B. A., AND FIELD, D. J. 使用超完备基组进行稀疏编码：初级视觉皮层采用的一种策略？《视觉研究》37, 23（1997），3311–3325。[40] ORORBIA, A. 基于数据流持续学习的脉冲神经预测编码。arXiv预印本 arXiv:1908.08655（2019）。[41] ORORBIA, A., AND KIFER, D. 用于学习生成模型的神经编码框架。《自然通讯》13, 1（2022），1–14。[42] ORORBIA, A., MALI, A., GILES, C. L., AND KIFER, D. 通过局部对齐分布式表征，实现循环神经架构的持续学习。arXiv预印本 arXiv:1810.07411（2018）。[43] ORORBIA, A., MALI, A., KIFER, D., AND GILES, C. L. 终身神经预测编码：累积学习时，稀疏性可减少遗忘。arXiv预印本 arXiv:1905.10696（2019）。[44] ORORBIA, A. G., AND KELLY, M. A. CogNGen：构建超维预测处理认知架构的核心。载于第44届认知科学学会年会论文集（多伦多，安大略省，2022），认知科学学会，第1322–1329页。[45] ORORBIA, A. G., AND MALI, A. 用于传播局部目标表征的生物学启发算法。载于AAAI人工智能会议论文集（2019），第33卷，第4651–4658页。[46] ORORBIA, A., REITTER, D., WU, J., AND GILES, C. L. 用于半监督分类的深度混合架构在线学习。载于机器学习与数据库知识发现联合欧洲会议（2015），施普林格，第516–532页。[47] ORORBIA II, A. G., HAFFNER, P., REITTER, D., AND GILES, C. L. 通过最小化差异学习适应。arXiv预印本 arXiv:1711.11542（2017）。[48] RAO, R. P., AND BALLARD, D. H. 视觉皮层中的预测编码：对某些经典外感受野效应的功能性解释。《自然神经科学》2, 1（1999）。[49] RUMELHART, D. E., HINTON, G. E., AND WILLIAMS, R. J. 通过反向传播误差学习表征。《自然》323, 6088（1986），533–536。[50] SALAKHUTDINOV, R., AND LAROCHELLE, H. 深度玻尔兹曼机的高效学习。载于第十三届国际人工智能与统计会议论文集（2010），JMLR研讨会与会议论文集，第693–700页。[51] SALVATORI, T., SONG, Y., HONG, Y., SHA, L., FRIEDER, S., XU, Z., BOGACZ, R., AND LUKASIEWICZ, T. 通过预测编码实现联想记忆。《神经信息处理系统进展》34（2021），3874–3886。

[52] SALVATORI, T., SONG, Y., XU, Z., LUKASIEWICZ, T., BOGACZ, R., LIN, H., FAN, Y., ZHANG, J., BAI, B., AND XU, ZHENGHUA, E. A. 通过预测编码实现反向微分。载于第*36*届*AAAI*人工智能会议论文集，*AAAI 2022*，温哥华，不列颠哥伦比亚省，加拿大，*2022*年*2*月*22*日至*3*月*1*日（2022），第10177卷，AAAI出版社，第507–524页。[53] SAXE, A. M., MCCLELLAND, J. L., AND GANGULI, S. 深度线性神经网络学习非线性动力学的精确解。*arXiv*预印本 *arXiv:1312.6120*（2013）。[54] SCELLIER, B., AND BENGIO, Y. 平衡传播：弥合基于能量的模型与反向传播之间的差距。《计算神经科学前沿》*11*（2017），24。[55] SCHWARTZ, R., DODGE, J., SMITH, N. A., AND ETZIONI, O. 绿色人工智能。《美国计算机协会通讯》*63*, 12（2020），54–63。[56] SHEPHERD, G. M. 真实神经元架构对神经网络模拟的重要性。计算神经科学（1990），82–96。[57] SILVER, D., HUANG, A., MADDISON, C. J., GUEZ, A., SIFRE, L., VAN DEN DRIESSCHE, G., SCHRIT-TWIESER, J., ANTONOGLOU, I., PANNEERSHELVAM, V., AND LANCTOT, MARC, E. A. 运用深度神经网络和树搜索掌握围棋。《自然》*529*, 7587（2016），484–489。[58] STRUBELL, E., GANESH, A., AND MCCALLUM, A. 自然语言处理深度学习中的能源与政策考量。载于第*57*届计算语言学协会年会论文集（佛罗伦萨，意大利，2019年7月），计算语言学协会，第3645–3650页。[59] WERBOS, P. J. 非线性灵敏度分析进展的应用。载于系统建模与优化：第*10*届*IFIP*会议论文集，纽约市，美国，*1981*年*8*月*31*日至*9*月*4*日（1981），施普林格，第762–770页。[60] WU, D. J., YANG, A. C., AND PRABHU, V. U. Afro-MNIST：面向低资源语言的MNIST风格数据集合成生成，2020年。[61] XIAO, H., RASUL, K., AND VOLLGRAF, R. Fashion-MNIST：用于机器学习算法基准测试的新型图像数据集。*arXiv*预印本 *arXiv:1708.07747*（2017）。

附录

PFF 学习过程

在算法1中，我们给出了 PFF 推理与学习过程的完整形式化规范。请注意，伪代码中使用的符号遵循主论文中定义/介绍的相同符号。

关于侧向竞争突触

在 PFF 算法中，我们将一种可学习形式的侧向抑制和自激引入表征电路的动力学中，如下所示（为方便起见，以下重述）：

$$\mathbf{z}^{\ell}(t) = \beta\Big(\phi^{\ell}\big(\mathbf{W}^{\ell}\cdot \mathrm{LN}(\mathbf{z}^{\ell-1}(t-1)) + \mathbf{V}^{\ell}\cdot \mathrm{LN}(\mathbf{z}^{\ell+1}(t-1)) - \mathbf{L}^{\ell}\cdot \mathrm{LN}(\mathbf{z}^{\ell}(t-1)) + \epsilon_r^{\ell}\Big) + (1-\beta)\mathbf{z}^{\ell}(t-1). \tag{18}$$

侧向突触权重矩阵$\mathbf{L}^{\ell}$也被指出可进一步分解如下：

$$\mathbf{L}^{\ell} = \hat{\mathbf{L}}^{\ell}\odot \mathbf{M}^{\ell}\odot(1-\mathbf{I}) + \hat{\mathbf{L}}^{\ell}\odot \mathbf{I} \tag{19}$$

其中$\mathbf{I}^{\ell}\in\{0,1\}^{J_\ell\times J_\ell}$是单位矩阵。第二项$\hat{}\mathbf{L}^{\ell}\odot\mathbf{I}$产生自激效应，即获胜神经元会抑制组内失败神经元的活动；而第一项$\hat{}\mathbf{L}^{\ell}\odot\mathbf{M}^{\ell}\odot(1-\mathbf{I})$产生交叉抑制效应（矩阵$\mathbf{M}^{\ell}$的设计细节稍后介绍）。请注意，$\hat{}\mathbf{L}^{\ell}$被强制为严格正值，这意味着在随机初始化后以及每次更新后，我们都会通过应用线性整流函数来实现这一点：$\hat{}\mathbf{L}^{\ell}(t+1) = \mathrm{ReLU}(\hat{}\mathbf{L}^{\ell}(t))$。我们指出，$\hat{}\mathbf{L}^{\ell}\in\mathcal{R}^{J_\ell\times J_\ell}$是一个可学习参数矩阵，而$\mathbf{M}^{\ell}\in\{0,1\}^{J_\ell\times J_\ell}$是一个二值掩码矩阵，旨在强制形成特定的侧向神经竞争模式。

掩蔽矩阵 $\mathbf{M}^{\ell}$ 为设计者/实验者/建模者提供了一种灵活而实用的方法，可用于在 PFF 表征电路的任意层中，为神经单元设计不同类型的竞争方案。尽管编程这一矩阵的方式数不胜数（从简单的赢家通吃方案，到类似于为神经生物学计算模型设计的复杂神经群组模式），我们选择设计一种“神经柱”形式的竞争（跨层抑制），即让预定群组内的神经元彼此竞争（一个层包含多个子群组），从而涌现出一种由突触驱动的 K 赢家通吃形式。形式上，$\mathbf{M}^{\ell}$ 按照以下公式创建：$\mathbf{M}^{\ell} = \widehat{\ }\mathbf{M}^{\ell}\odot(1-\mathbf{I})$。$\widehat{\ }\mathbf{M}^{\ell}\in\{0,1\}^{J_\ell\times J_\ell}$ 由实验者按照以下流程设定（在希望神经元对彼此产生侧向抑制的位置填入 1）：

1. 创建形状为 $J_\ell\times K$ 的 J_ℓ/K 零矩阵，即 $\{\mathrm{S}_1, \mathrm{S}2, \cdots, \mathrm{S}_k, \cdots, \mathrm{S}_C\}$ (其中 $C = J_\ell/K$) 2. 在每个矩阵 S_k 中，将坐标 $c = \{1,\cdots,k,\cdots,K\}$（列索引）和 $r = \{1+K*(k-1)\ \cdots, k+K*(k-1)\ \cdots, K+K*(k-1)\}$（行索引）的所有组合位置设为 1 3. 沿水平方向拼接这些 J_ℓ/K 矩阵，即 $\widehat{\ }\mathbf{M}^{\ell} = <\mathrm{S}_1, \mathrm{S}2, \cdots, \mathrm{S}_C>$。

可视化样本（展开）

图5和图6以更大的图像比例/尺寸展示了正文中 PFF 模型在 MNIST 和 K-MNIST 上的重建结果与合成样本。此外，我们还在每幅图的底部各行中，展示了 PFF 针对这两个数据集获取的感受野。可以看到，表示电路和生成电路所获取的感受野（即最靠近感觉输入层的那一层的突触）似乎都能提取与数字或汉字笔画相关的有用/有趣结构；正如全连接神经结构所预期的那样，它们往往会获取具有代表性的完整物体模板（如果希望每个感受野只获取单个笔画/组成特征，则需要额外施加先验，例如卷积，或 [3, 19] 中采用的局部连接感受野结构）。

数据集详情

在正文中，我们对若干（灰度）图像集合进行了实验，即 MNIST、Kuzushiji-MNIST（K-MNIST）、Fashion MNIST（F-MNIST）、Not-MNIST（N-MNIST）和埃塞俄比亚文（Et-MNIST）数据库。这些数据库都包含来自 10 个不同类别的 28×28 张图像。MNIST 数据集 [29] 具体

算法1 预测式前向-前向算法（PFF）的信用分配算法。 红色 表示表征电路的计算，而 蓝色 表示生成电路的计算。

输入： 样本 $(\mathbf{y}_i,\ \mathbf{x}_i)$，数据标签 c_i (二元标签： $1=$ "正"， $0=$ "负")，PFF 参数 Θ_r 和 Θ_g

超参数： 状态插值 β，随机梯度下降步长 η，噪声尺度 σ_r 和 σ_z，刺激时间 T

// 请注意，$\text{LN}(\mathbf{z}^\ell) = \mathbf{z}^\ell/(||\mathbf{z}^\ell||_2 + 1\text{e}{-8})$ 以及 $\leftarrow$ 表示变量/对象的覆盖

函数 SIMULATE($(\mathbf{y}_i,\ \mathbf{x}_i,\ c_i)\ \Theta_r,\ \Theta_g$)

  //运行前向传播以获得初始活动

  $\mathbf{z}^0 = \phi^\ell(\mathbf{W}^\ell \cdot \mathbf{z}^{\ell-1})$，对于 $\ell = 1,\ 2,\ \ldots..\ L,\quad \mathbf{z}^{L+1} = \mathbf{y}_i,\quad \mathbf{z}_s = \hat{\mathbf{z}}^{L+1} = \mathbf{0}$

  $\mathbf{z}_0 = \mathbf{x}_i$

  对于 $t = 1$ 到 T **执行**

    //运行表征电路

    对于 $\ell = 1$ 到 L **执行** ▷ 使用逐层参数 $\Theta_r^\ell = \{\mathbf{W}^\ell,\ \mathbf{V}^\ell,\ \hat{\mathbf{L}}^\ell\}$计算表征活动

      $\epsilon_r^\ell \sim \mathcal{N}(0, \sigma_r)$

      $\mathbf{z}^\ell(t) = \beta\Big(\phi^\ell\big(\mathbf{W}^\ell \cdot \text{LN}(\mathbf{z}^{\ell-1}(t-1)) + \mathbf{V}^\ell \cdot \text{LN}(\mathbf{z}^{\ell+1}(t-1))\big) - \mathbf{L}^\ell \cdot \text{LN}(\mathbf{z}^\ell(t-1)) + \epsilon_r^\ell\Big) + (1-\beta)\mathbf{z}^\ell(t-1)$

      计算局部优度损失 $\mathcal{L}(\Theta_r^\ell)$（使用数据标签 c_i 的方程 1 或 2）

      $\Delta\mathbf{W}^\ell = \left(2\ \frac{\partial\mathcal{L}(\Theta_r^\ell)}{\partial\sum_j^{J^\ell}(\mathbf{z}_j^\ell)^2} \odot \mathbf{z}^\ell\right) \cdot \big(\text{LN}(\hat{\mathbf{z}}^{\ell-1})\big)^T$

      $\Delta\mathbf{V}^\ell = \left(2\ \frac{\partial\mathcal{L}(\Theta_r^\ell)}{\partial\sum_j^{J^\ell}(\mathbf{z}_j^\ell)^2} \odot \mathbf{z}^\ell\right) \cdot \big(\text{LN}(\mathbf{z}^{\ell+1})\big)^T$

      $\Delta\hat{\mathbf{L}}^\ell = \left(\left(2\ \frac{\partial\mathcal{L}(\Theta_r^\ell)}{\partial\sum_j^{J^\ell}(\mathbf{z}_j^\ell)^2} \odot \mathbf{z}^\ell\right) \cdot \big(\text{LN}(\mathbf{z}^\ell)\big)^T\right) \odot \frac{\partial\mathbf{L}^\ell}{\partial\hat{\mathbf{L}}^\ell}$

      $\mathbf{W}^\ell \leftarrow \mathbf{W}^\ell - \eta\Delta\mathbf{W}^\ell,\ \ \mathbf{V}^\ell \leftarrow \mathbf{V}^\ell - \eta\Delta\mathbf{V}^\ell\ \hat{\mathbf{L}}^\ell \leftarrow \hat{\mathbf{L}}^\ell - \eta\Delta\hat{\mathbf{L}}^\ell$ ▷ 随机梯度下降，步长为 η（或使用 Adam [25]）

    //运行生成电路

    对于 $\ell = L$ 至 0，**执行** ▷ 使用逐层参数计算生成式预测$\Theta_g^\ell = \{\mathbf{G}^{\ell+1}\}$

      如果 $\ell < L$ **则**

        $\epsilon^\ell \sim \mathcal{N}(0, \sigma_z),\quad \hat{\mathbf{z}}^{\ell+1} = \phi^{\ell+1}(\mathbf{z}^{\ell+1} + \epsilon^{\ell+1})$

      $\bar{\mathbf{z}}^\ell = \phi^\ell(\mathbf{G}^{\ell+1} \cdot \text{LN}(\hat{\mathbf{z}}^{\ell+1}))$

      计算局部生成损失$\mathcal{L}_g^\ell(\mathbf{G}^{\ell+1}) = \frac{1}{2}\sum_j(\bar{\mathbf{z}}_j^\ell - \mathbf{z}_j^\ell(t))^2$

      $\mathbf{e}^\ell = \bar{\mathbf{z}}^\ell - \mathbf{z}^\ell,\ \ \Delta\mathbf{G}^{\ell+1} = \mathbf{e}^\ell \cdot \big(\text{LN}(\hat{\mathbf{z}}^{\ell+1}(t))\big)^T$ ▷注意，$\mathbf{e}^\ell = \frac{\partial\mathcal{L}_g^\ell(\mathbf{G}^{\ell+1})}{\partial\bar{\mathbf{z}}^\ell}$

      $\mathbf{G}^{\ell+1} \leftarrow \mathbf{G}^{\ell+1} - \eta\Delta\mathbf{G}^{\ell+1}$

      $\hat{\mathbf{z}}^{L+1} \leftarrow \hat{\mathbf{z}}^{L+1} - \gamma\frac{\partial\mathcal{L}_g^L(\mathbf{G}^{L+1})}{\partial\hat{\mathbf{z}}^{L+1}}$ ▷更新潜变量 $\mathbf{z}_s$ (迭代推理的一步)

  返回 $\Theta_g = \{\mathbf{G}^1, ..., \mathbf{G}^L, \mathbf{G}^{\ell+1}\}, \Theta_r = \{\mathbf{W}^1, ..., \mathbf{W}^{L-1}, \mathbf{W}^L\}$ ▷输出新近更新的 PFF 参数

包含展示 10 个不同类别的手写数字图像。Kuzushiji-MNIST 是 MNIST 的一种颇具挑战性的即插即用替代数据集，其中包含描绘手绘日语汉字的图像 [9]（每个类别对应其现代平假名写法，共有 10 个类别）。Fashion MNIST[61] 是 MNIST 的一种颇具挑战性的即插即用替代数据集，包含描绘 10 类服装物品的图像模式。NotMNIST 数据库[7] 是 MNIST 的一个难度更高的变体，它将数字替换为采用不同字体/字形的字符（字母 A-J）。最后，Ethiopic MNIST 是一个包含十种不同数字的数据库（通过对 Unicode 数字范例进行弹性变形生成），用于吉兹语（具体而言，该数据库包含从一到十的数字模式，因为其中不存在表示零的数字）；它是 [60] 中提出的低资源字形/数字识别数据集所涵盖的四种正字法之一。

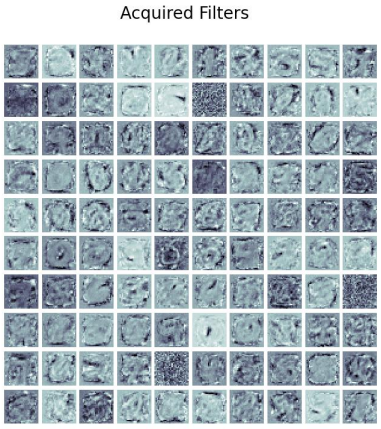
[7] http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html



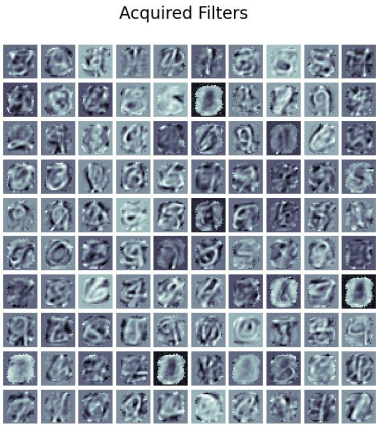
(a) PFF 重建图像。



(b) PFF 采样图像。



(c) PFF 表征感受野。

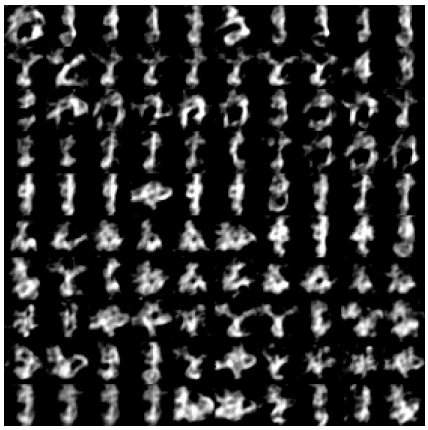


(d) PFF 生成式感受野。

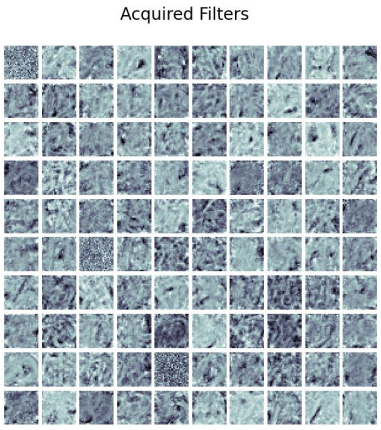
图 5：MNIST 模型重建（左）和生成（右）的样本。底行显示表征电路（左）最底层的感受野，以及生成电路（右）的感受野。



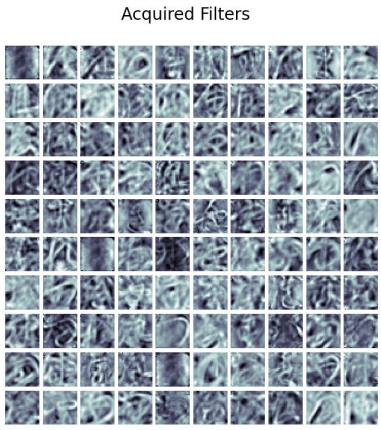
(a) PFF 重建图像。



(b) PFF 采样图像。



(c) PFF 表征感受野。



(d) PFF 生成式感受野。

图 6：上排为 Kuzushiji-MNIST 模型重建的样本（左）和生成的样本（右）。下排为表征电路最底层的感受野（左）以及生成电路的感受野（右）。