论生物物理突触的优势

Julian Lemmel、Radu Grosu

维也纳工业大学信息学院，奥地利。
julian.lemmel@tuwien.ac.at, radu.grosu@tuwien.ac.at

arXiv:2303.04944v1 [cs.NE] 8 Mar 2023

摘要

人工神经网络及其循环人工神经网络实例的逼近能力，与这些网络中包含的参数数量密切相关。然而，人类理解上的复杂性障碍，可能与网络中的神经元和突触数量，以及相关的非线性变换有关。本文表明，使用液体时间常数网络中的生物物理突触有两大优势。首先，在给定神经元和突触数量的情况下，它们能够容纳更多参数。其次，它们能够将非线性网络变换表述为具有状态依赖系数的线性系统。这两点都能提高可解释性：对于给定任务，它们能够学习一个关于输入特征呈线性的系统，其规模小于当前最先进的方法。我们通过多种时间序列预测任务验证了上述观点，并认为研究结果适用于任何前馈或循环人工神经网络。

引言

受脉冲神经元的启发，人工神经元（AN）在一个单元中，将生物神经元的加性行为与其突触的分级非线性行为结合起来（Bishop 1995；Goodfellow、Bengio 和 Courville 2016）。从生物物理学的角度来看，这使人工神经元显得不合情理，也阻碍了它们在神经科学中的应用。

然而，人工神经网络（ANN）纠正了这一生物学谬误。在人工神经网络中，神经元和突触分别意味着什么并不重要，重要的是网络本身的数学表达。这一点在残差网络中体现得最为明显：出于技术原因，它们不得不将加性变换与分级变换分离，并引入新的状态变量；这些变量是加性神经元单元的输出，而非非线性突触单元的输出（He 等人，2016 年）。

这种区分使我们能够将残差网络与液体时间常数网络（LTCs）联系起来。LTCs 是一种针对非脉冲神经元的生物物理模型，与基因调控网络共享一些架构特征，例如激活、抑制、序列化、互斥和同步（Lechner 等，2019、2020；Hasani 等，2021；Alon，2007）。LTCs 能够捕捉大型物种视网膜中神经元的行为（Kandel 等，2013），也能捕捉小型物种中神经元的行为例如线虫 C. elegans 中的神经元（Wicks、Roehrig 和 Rankin，1996）。在 LTCs 中，神经元是一个电容器，其变化率等于漏电流与突触电流之和。突触的电导会随突触前神经元的电位以分级的非线性方式变化，再乘以突触后神经元的电位差，从而产生突触电流。因此，分级非线性变换是突触所执行的变换，这确实符合自然界的情况，而神经元并不执行这种变换。

版权所有 © 2022，人工智能促进协会（www.aaai.org）。保留所有权利。

与 ResNet、神经常微分方程和连续时间循环神经网络（Chen 等，2018；Funahashi 和 Nakamura，1993）不同，液体时间常数网络会将电导与电势差相乘（或进行门控）。这是由物理学决定的，因为要获得电流就必须如此。门控使每个神经元都可以解释为一个具有状态依赖系数的线性系统（Alvarez-Melis 和 Jaakkola，2018；C¸imen，2008）。此外，液体时间常数网络将每个激活函数与一个突触相关联（如自然界中的情况），而不是与一个神经元相关联（如人工神经网络中的情况）。因此，对于给定数量的神经元和突触，液体时间常数网络能够在网络中容纳多得多的参数。由于人工神经网络和液体时间常数网络的逼近能力与其可学习参数的数量密切相关，液体时间常数网络能够用小得多的网络逼近相同的行为，而且该网络可通过其架构模式加以解释。我们认为，非线性和神经网络的规模是人类理解网络的主要复杂性障碍。

在任何人工神经网络中，都可以将激活函数移至突触，从而实现与液体时间常数网络同样的网络规模缩减优势。S形激活函数的门控可以用双曲正切激活函数取代。然而，这样会失去神经网络的生物物理学解释、神经元的线性解释，以及突触的极性。

我们比较了液体时间常数网络与连续时间循环神经网络、（增强型）神经常微分方程网络、长短期记忆网络和连续时间 GRU 在各种循环任务上的表达能力。在这项比较中，我们考察了采用神经元激活函数和突触激活函数的液体时间常数网络与连续时间循环神经网络。我们还研究了使用电势差门控 Sigmoid 激活的益处。结果表明，突触激活能显著减少完成任务所需的神经元及其相关突触数量，这不仅适用于液体时间常数网络，也适用于连续时间循环神经网络。我们还表明，双曲正切激

活函数在连续时间循环神经网络中的表达能力与电势差门控 Sigmoid 函数相近，但它失去了线性解释。

本文其余部分的结构如下。首先，我们将重新审视人工神经网络、残差网络、神经常微分方程、连续时间循环神经网络和液体时间常数网络。这为随后展示生物物理突触在各种循环任务中的优势奠定了基础。最后，我们将讨论研究结果，并简要谈谈未来的工作。

重新审视神经网络

人工神经网络

人工神经元接收一个或多个输入，以线性方式将它们相加，再将结果传递给非线性激活函数，其偏置 b 是神经元触发（产生脉冲）的条件。不过，激活是渐变的（不产生脉冲），呈平滑的（例如 S 形）曲线。形式化地：

$$y_i^{t+1} = \sigma(\sum_{j=1}^{n} w_{ji}^t \, y_j^t + b_i^{t+1}) \quad \sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$

其中，如图 1 所示，y_i^{t+1} 是第 $t+1$层神经元 i 的输出，y_j^t 是第 t层神经元 j 的输出，w_{ji}^t 是第 t 层神经元 j 与第 $t+1$层神经元 i 之间突触对应的权重，b_i^{t+1} 是第 $t+1$层神经元i 的偏置（阈值），σ 是激活函数，例如上文的逻辑函数。具有一个输入层、一个输出层和 $N \geq 2$ 个隐藏层的网络称为深度神经网络（DNN）（Goodfellow、Bengio 和 Courville，2016）。人工神经网络是通用逼近器。

尽管人工神经元在生物物理层面并不合理，但人工神经网络实际上与非脉冲神经网络密切相关。为了说明这一点，我们先来看看残差网络（He 等人，2016 年）。

残差神经网络

具有大量隐藏层的深度神经网络会遇到退化问题，即使经过处理消除了梯度消失，这个问题依然存在。直观而言，深度神经网络无法准确学习恒等映射。因此，人们通过跳跃连接的形式将恒等映射直接加到深度神经网络中（He 等人，2016 年）。

如图 1 所示，由此得到的架构称为残差神经网络（ResNet）[1]。在残差网络中，求和的输出 x_i^t 与 Sigmoid 函数的输出 y_i^t有所区别。形式化地说：

$$x_i^{t+1} = x_i^t + \sum_{j=1}^{n} w_{ji}^t \, y_j^t \qquad y_j^t = \sigma(x_j^t + b_j^t) \tag{2}$$

从生物物理学的角度来看，这一区分非常重要。其核心思想是，神经元只是求和单元，而 S 形变换发生在突触中。事实上，也可以将权重置于突触变换中，从而得到等价的方程：

$$x_i^{t+1} = x_i^t + \sum_{j=1}^{n} y_{ji}^t \qquad y_{ji}^t = w_{ji}^t \, \sigma(x_j^t + b_j^t) \tag{3}$$

[1]在（He 等人，2016 年）中，x_i^t 跳过第一个求和操作，并直接与 x_i^{t+2} 相加。因此，图 1 所示的架构可以视为具有最细粒度跳跃连接的残差网络。

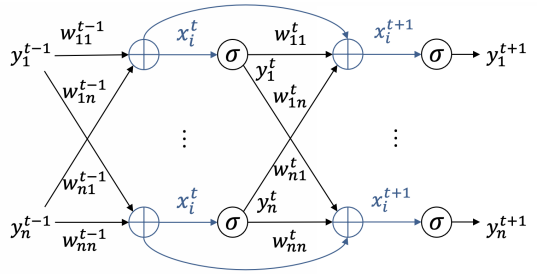

图 1：深度神经网络（黑色）和 ResNet（黑色和蓝色）。

这里的 w_{ji}^t 可以被视为输入相关突触变换 $\sigma(x_j^t + b_j^t)$ 的最大电导。这种变换本质上是分级的，也就是说，不产生脉冲。由于残差网络是特定的深度神经网络，并以恒等映射作为线性激活，因此它们也是通用逼近器。

神经常微分方程

方程 (3) 是一组微分方程的欧拉离散化，其中时间步长简单地取为 1（E 2017；Chen 等人 2018）。数学形式如下：

$$\dot{x}_i(t) = \sum_{j=1}^{n} y_{ji}(t) \quad y_{ji}(t) = w_{ji}(t) \, \sigma(x_j(t) + b_j(t)) \tag{4}$$

在这些方程中，x、y以及参数 w 和 b 随时间连续变化。现在假设我们令参数保持恒定。我们还能得到一个通用的常微分方程逼近器吗？答案是肯定的，正如我们将在下一节中展示的那样。微分方程如下：

$$\dot{x}_i(t) = \sum_{j=1}^{n} y_{ji}(t) \quad y_{ji}(t) = w_{ji} \, \sigma(x_j(t) + b_j) \tag{5}$$

这就是神经常微分方程（NeuralODEs）的形式（E 2017；Chen 等人 2018）。[2] 将网络的状态视为某个和的 S 形函数 y，等价于将状态视为多个 S 形函数之和 x。

定理 1（神经常微分方程）。设 x 和 y 为状态向量。那么 $\dot{y} = \sigma(Wy + b)$ 等价于 $\dot{x} = W\sigma(x + b)$。

证明。 取 $x = Wy$。则有：

$$\dot{x} = W\dot{y} = W\sigma(Wy + b) = W\sigma(x + b) \qquad \square$$

一种称为 ANODE 的小幅扩展见（Dupont、Doucet 和 Teh 2019），它将输入嵌入内部状态，并将状态投影到输出 S，如下所示：

$$x(t_0) = [x, 0]^T \quad y = \pi_S(x(t_N)) \tag{6}$$

神经常微分方程比残差网络更难训练。训练时，可以使用伴随方程，并采用高效的数值求解器（E，2017；Chen 等，2018）。

[2]严格来说，NeuralODEs $\dot{x}(t) = f(x)$ 中用于函数 f 的神经网络层数量可以任意。

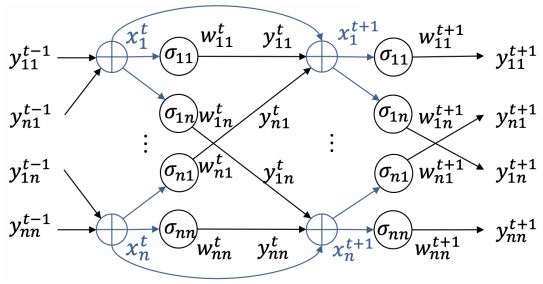

图2：突触激活深度神经网络和 ResNe t.

连续时间循环神经网络

自治情形。 在这种连续时间循环神经网络形式中，输入是初始状态。我们称之为自治连续时间循环神经网络，它们在神经常微分方程前添加了一项 $-w_i x_i(t)$（Funahashi 和 Nakamura，1993）。其数学形式如下：

$$\begin{aligned}\dot{x}_i(t) &= -w_i x_i(t) + \sum_{j=1}^{n} y_{ji}(t) \\ y_{ji}(t) &= w_{ji}\,\sigma(x_j(t) + b_j)\end{aligned} \tag{7}$$

首项使系统回到平衡状态（此时没有输入）。因此，微小扰动会被遗忘，也就是说，系统是稳定的。与神经常微分方程类似，可以互换求和与激活的顺序。

定理2 (自治连续时间循环神经网络)。 设 x 和 y 为状态向量。那么 $\dot{y} = -w * y + \sigma(Wy + b)$ 和 $\dot{x} = -w * x + W\sigma(x + b)$ 是等价的常微分方程，其中 $*$ 是向量的逐点乘积。

证明。 设 $x = Wy$。那么（Funahashi 和 Nakamura，1993）：

$$\begin{aligned}\dot{x} &= W\dot{y} \\ &= W(-w * y + \sigma(Wy + b)) = -w * x + W\sigma(x + b)\end{aligned}$$ □

自治连续时间循环神经网络是通用逼近器，在这方面，稳定化并不重要（Funahashi 和 Nakamura，1993）。因此，神经常微分方程也是通用逼近器。

突触激活。与人工神经网络一样，自治连续时间循环神经网络会为每个激活函数指定一个神经元。因此，我们将其称为 NA-ACT-RNN，其中 NA 代表神经元激活。

然而，如图2所示，任何人工神经网络都可以改写为将激活函数与突触相关联的形式。我们将这种形式的自治连续时间循环神经网络称为 SA-ACT-RNN，其中 SA 代表突触激活。若再为每个激活函数添加一个方差 a，则可得到以下显式形式：

$$\begin{aligned}\dot{x}_i(t) &= -w_i x_i(t) + \sum_{j=1}^{n} y_{ji}(t) \\ y_{ji}(t) &= w_{ji}\,\sigma(a_{ji} x_j(t) + b_{ji})\end{aligned} \tag{8}$$

SA-ACT-RNN 的优势在于，在神经元和突触数量相同的情况下，它能在网络中容纳更多参数。例如，一个由 32 个神经元以全连接方式构成的 SA-ACT-RNN 可容纳 3104 个参数。这大致相当于一个由 54 个神经元构成、可容纳 3132 个参数的 NA-ACT-RNN。

虽然神经网络的逼近能力与其参数数量密切相关，但我们坚信，阻碍人类理解的复杂性障碍在于神经元和突触的数量。

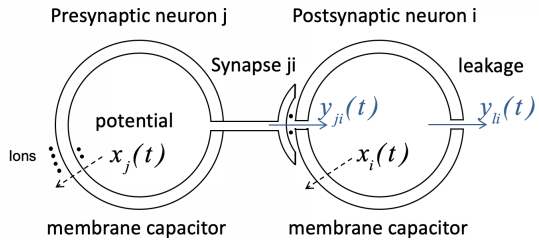

图3：非脉冲神经元的电表示。

一般情况。 连续时间循环神经网络通常也具有随时间变化的输入信号 u，也就是说，它们是循环神经网络。如何处理输入信号起着非常重要的作用。

一种常见的添加输入信号 u的方法，是将 sigmoid 函数中的求和项扩展为再加上一个对应于输入的求和项。用向量形式表示如下：

$$\dot{y} = -w * y + \sigma(Wy + Vu + b) \tag{9}$$

这种形式具有出色的收敛性质，但无法扩展到突触激活。因此，我们更倾向于采用以下形式，它具有相同的收敛性质：

$$\dot{x} = -w * x + W\sigma(a^x * x + b^x) + V\sigma(a^u * u + b^u) \tag{10}$$

其中 a^x，b^x 和 a^u，b^u 分别表示状态向量和输入向量的方差与偏置向量。最后，另一种常见的将输入以线性方式添加到连续时间循环神经网络的方法如下。

$$\dot{x} = -w * x + W\sigma(a * x + b) + Vu \tag{11}$$

突触激活。与 SA-ACT-RNN 类似，最后两种连续时间循环神经网络也可以通过将每个激活与一个突触相关联来改写。为区分这两个变体，我们分别称其为 NA-CT-RNN 和突触激活连续时间循环神经网络。其 S形输入版本的标量形式如下。线性输入版本与之非常相似：

$$\begin{aligned}\dot{x}_i(t) &= -w_i x_i(t) + \sum_{j=1}^{n} y_{ji}(t) + \sum_{j=1}^{m} z_{ji}(t) \\ y_{ji}(t) &= w_{ji}\,\sigma(a^x_{ji} x_j(t) + b^x_{ji}) \\ z_{ji}(t) &= v_{ji}\,\sigma(a^u_{ji} u_j(t) + b^u_{ji})\end{aligned} \tag{12}$$

现在考虑一个包含 32 个神经元、彼此全部连接且有 32 个输入的 SA-CT-RNN。它总共包含 6176 个参数。这大致相当于一个包含 54 个神经元、总共包含 6102 个参数的 NA-CT-RNN。

液态时间常数网络

自治情形。 液体时间常数网络是一种用于小型物种神经系统的生物物理模型（Wicks、Roehrig 和 Rankin，1996；Lechner 等，2019、2020；Hasani 等，2021），也可用于大型物种的视网膜（Kandel 等，2013）。由于这些神经系统的尺寸很小（≤1 毫米），神经信号在模拟域中被动传输，且不会显著衰减。因此，为了准确传输信号，神经元无需产生脉冲。

如图 3 所示，神经元膜是一种绝缘体，其内外两侧都有离子。从电学角度来看，

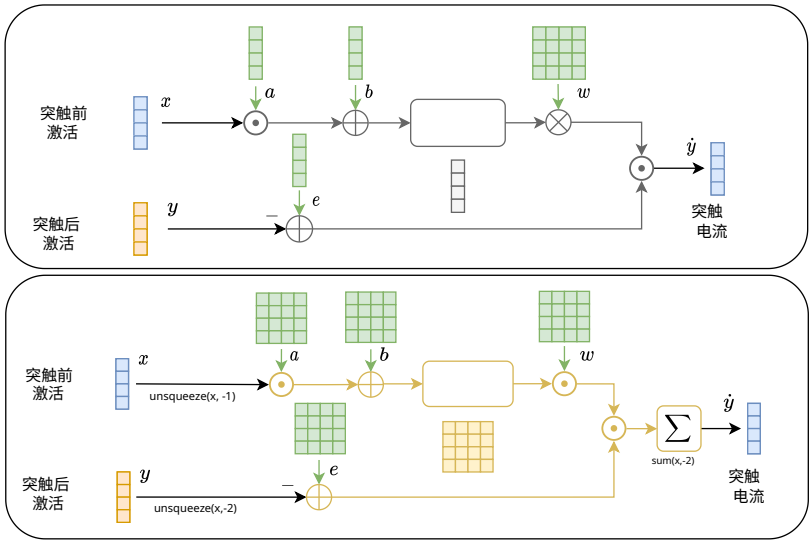

图4：LTC 中的突触层，包括突触激活（上）和神经元激活（下）。

它是一个电容器。内外离子浓度之差决定了膜电位（MP）x。x 的变化率取决于流经膜的电流 y。这些电流包括外部电流（在 ALTC 中忽略）、漏电流和突触电流。为简单起见，我们只考虑化学突触。电容器方程如下：

$$\begin{aligned} C\,\dot{x}_i(t) &= w_{li}\,(e_{li}-x_i(t)) + \textstyle\sum_{j=1}^{n} y_{ji}(t) \\ y_{ji}(t) &= w_{ji}\,\sigma(a_{ji}x_j(t)+b_{ji})\,(e_{ji}-x_i(t)) \end{aligned} \tag{13}$$

其中，C 为膜电容，e_{li} 为静息电势，w_{li} 为漏电导，e_{ji} 为突触电势。对于兴奋性突触，这些电势为 0mV（电势 x_i 为负，因此电流为正）；对于抑制性突触，这些电势为 -90 mV（此时电流为负）。

方程（18）与 SA-ACT-RNN 的方程（7）非常相似。它们都包含漏电流，这确保了 ALTC 的稳定性；同时还包含由突触前神经元控制的突触电导 σ，其最大电导为 w_{ji}。该电导与电势差 $e_{ji}-x_i$（t）相乘，从而得到电流。这一生物物理约束使它们区别于 SA-ACT-RNN。那么，从机器学习的角度来看，这一门控项有何意义？正如我们将在下文证明的那样，它对 ALTC 的可解释性具有重要影响。

定理3（可解释性）。每个自适应液体时间常数网络神经元都可以解释为对其输入进行线性回归。

证明。设 $x(0)$ 为输入。该输入随时间传播为 $x(t)$。设 $w_{ji}\,\sigma(a_{ji}x_j+b_{ji})$ 为从神经元 j 到神经元 i 的状态依赖权重。那么，根据方程（18），对于每个 i，$\dot{x}_i(t)$ 都是关于 x 的线性回归。此外，x 的微小扰动会导致 $\dot{x}$ 发生微小变化。 □

ALTC 能够容纳比 SA-ACT-RNN 更多的参数。例如，一个由 32 个神经元全连接而成的 ALTC，能够容纳 4192 个参数，而 SA-ACT-RNN 只能容纳 3104 个参数。这大致相当于一个由 64 个神经元组成、可容纳 4224 个参数的 NA-ACT-RNN。

神经元激活。自然界中的每个突触都有各自不同的动力学，但在某些特定情况下，我们可能希望考虑神经元的所有输出突触都具有相同的激活参数。我们将这种 ALTC 称为 NA-ALTC，其中 NA 代表神经元激活。我们也会交替使用 ALTC 和 SA-ALTC。形式化地说：

$$\begin{aligned} C\,\dot{x}_i(t) &= w_{li}\,(e_{li}-x_i(t)) + \textstyle\sum_{j=1}^{n} y_{ji}(t) \\ y_{ji}(t) &= w_{ji}\,\sigma(a_j x_j(t)+b_j)\,(e_i-x_i(t)) \end{aligned} \tag{14}$$

若令 e_{li} 为零、a_j 为一，则除门控项 $e_{ji}-x_i(t)$ 外，NA-ALTC 与 NA-ACT-RNN 相同。如上所述，该项使 NA-ALTC 成为具有状态依赖系数的线性系统，这不仅提高了其可解释性，还使得在自动合成非线性控制器时能够应用状态依赖的 Ricatti 方程（C.imen 2008）。

在我们的实验中，我们发现，激活函数为双曲正切的连续时间循环神经网络，与激活函数为 S 形函数的液体时间常数网络具有非常相似的收敛性和学习准确度特性。然而，双曲正切函数无法像门控S形函数那样捕捉突触通道的开启程度及其相关极性：S形函数能够准确捕捉这一程度以及电势差所体现的门控极性。

一般情形。与连续时间循环神经网络一样，液体时间常数网络通常也关联一个时变输入信号 u。对于连续时间循环神经网络，我们同时考虑 S 形输入版本和线性输入版本。用标量形式表示时，前者可写成如下形式：

$$\begin{aligned} C\,\dot{x}_i(t) &= w_{li}(e_{li}-x_i(t)) + \textstyle\sum_{j=1}^{n} y_{ji}(t) + \sum_{j=1}^{m} z_{ji}(t) \\ y_{ji}(t) &= w_{ji}\,\sigma(a^x_{ji}x_j(t)+b^x_{ji})(e_{ji}-x_i(t)) \\ z_{ji}(t) &= v_{ji}\,\sigma(a^u_{ji}u_j(t)+b^u_{ji})(e_{ji}-x_i(t)) \end{aligned} \tag{15}$$

一个包含 32 个神经元、以全连接方式连接且有 32 个输入的 SA-LTC，共有 8288 个参数，

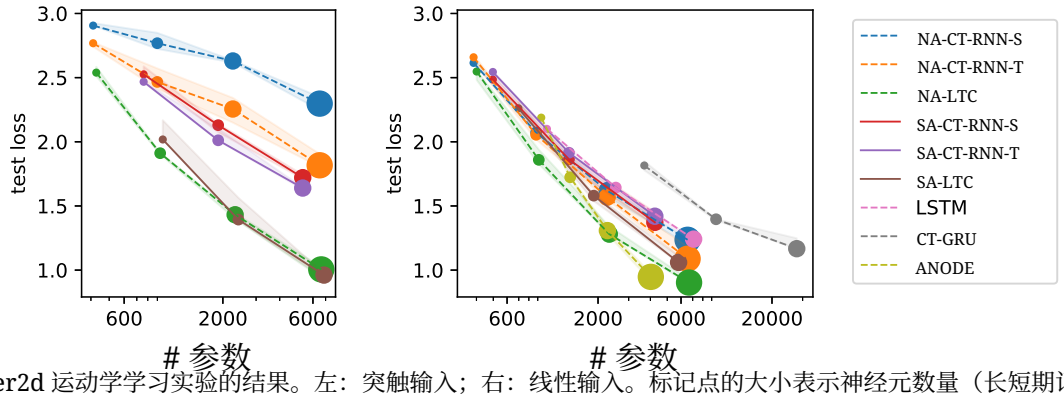

图 5：Walker2d 运动学学习实验的结果。左：突触输入；右：线性输入。标记点的大小表示神经元数量（长短期记忆网络中的细胞数量）：8、16、32 或 64（从小到大）。

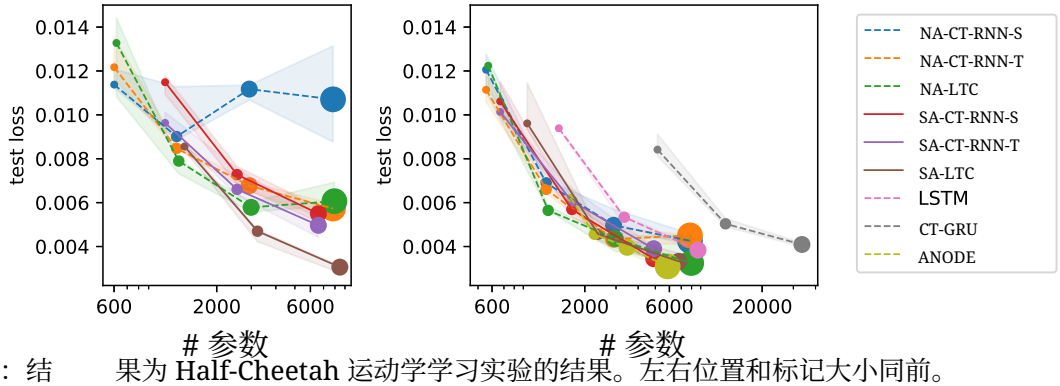

图 6：结　　果为 Half-Cheetah 运动学学习实验的结果。左右位置和标记大小同前。　　.

而 SA-CT-RNN 只有 6176 个参数。这大致相当于一个包含 63 个神经元、共有 8253 个参数的 NA-CT-RNN。

SA-LTC 的线性输入版本非常相似。形式上，它描述如下：

$$C\,\dot{x}_i(t) = w_{li}(e_{li} - x_i(t)) + \sum_{j=1}^{n} y_{ji}(t) + \sum_{j=1}^{m} z_{ji}(t)$$

$$\begin{aligned} y_{ji}(t) &= w_{ji}\,\sigma(a^x_{ji}x_j(t) + b^x_{ji})(e_{ji} - x_i(t)) \\ z_{ji}(t) &= v_{ji}\,u_j(t) \end{aligned} \tag{16}$$

一个包含 32 个神经元、采用全连接方式连接且具有 32 个输入的 SA-LTC 包含 5216 个参数。这大致对应于一个包含 51 个神经元的线性输入 NA-CT-RNN，后者总共包含 5355 个参数。

神经元激活。 与自主情况类似，也可以考虑 NA-LTC，其中一个神经元的所有输出突触都具有相同的激活参数。对于 S 形输入版本，可以得到以下方程：

$$C\,\dot{x}_i(t) = w_{li}(e_{li} - x_i(t)) + \sum_{j=1}^{n} y_{ji}(t) + \sum_{j=1}^{m} z_{ji}(t)$$

$$\begin{aligned} y_{ji}(t) &= w_{ji}\,\sigma(a^x_j x_j(t) + b^x_j)(e_i - x_i(t)) \\ z_{ji}(t) &= v_{ji}\,\sigma(a^u_j u_j(t) + b^u_j)(e_i - x_i(t)) \end{aligned} \tag{17}$$

线性输入版本与之类似，但在这种情况下 $z_{ji}(t) = v_{ji}\,u_j(t)$。与 ALTC 类似，NA-LTC 与 NA- 连续时间循环神经网络非常相似，区别在于门控项 $e_{ji} - x_i(t)$。

如前所述，可以使用双曲正切激活函数来去除门控，但代价是失去线性特性和生物物理意义。

LTC 是通用逼近器（Hasani 2020年；Hasani 等，2021）。这对其自主形式和一般形式，以及突触输入和线性输入都成立。

实验评估

序列模型结构

本节所有实验均采用三层序列结构。设 $u(t)$ 表示时间 t 的输入，$y_i(t)$ 表示层 i 在时间 t 的输出。最终层的输出即为预测输出 $\hat{y}(t) = y_3(t)$。

第一层将输入映射到一个 RNN 层，并采用线性（$y_1(t) = A_{in}u(t) + b_{in}$）变换或突触变换（如上所述），分别称为线性输入映射或突触输入映射。对于突触输入映射，这些感觉突触的实现方式与所用的 RNN 模型相一致，即采用突触激活或神经元激活；对于液体时间常数网络（LTCs），还会乘以电势差。

第二层包含 RNN 细胞，其输出通过常微分方程求解器计算。所求解的常微分方程由模型类型决定。不同变体已在上一节中说明。与传统的神经常微分方程不同，我们的模型中的 RNN 细胞在每个时间步之后都会保留状态（因而保留信息）t。

第三层与所用的具体模型类型无关，

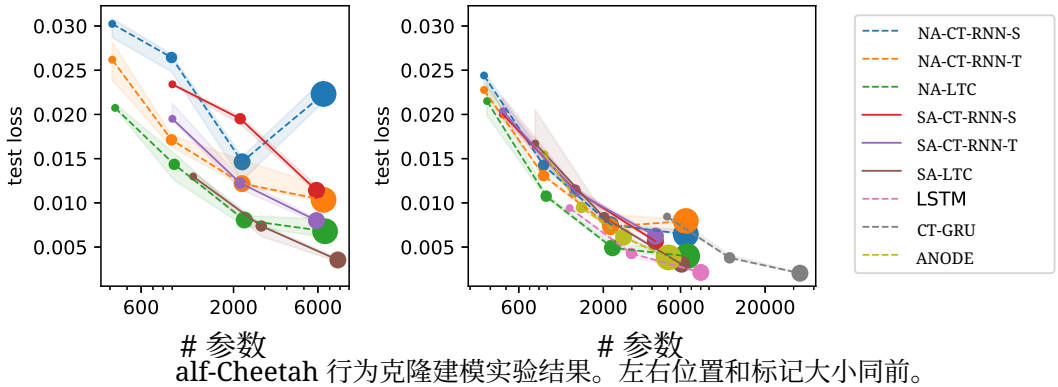

图7：Half-Cheetah 行为克隆建模实验结果。左右位置和标记大小同前。

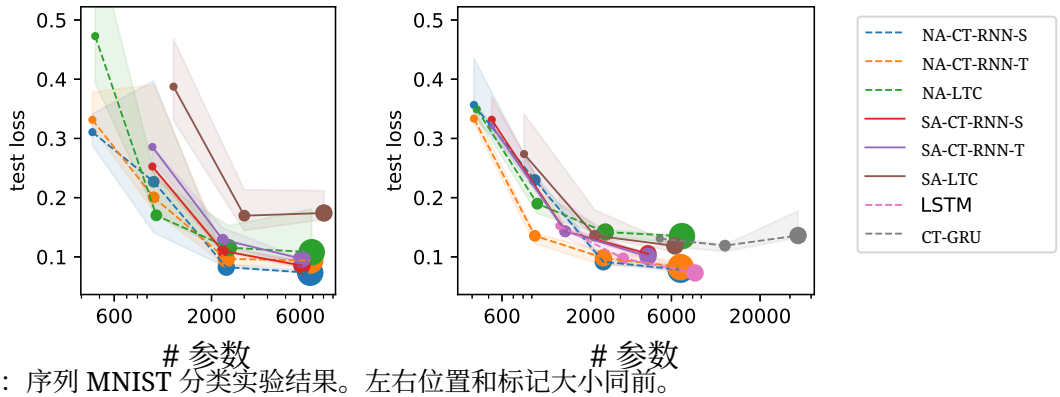

图8：序列 MNIST 分类实验结果。左右位置和标记大小同前。

它以线性方式将最终的循环神经网络状态 $y_2(t)$ 映射到输出向量$y_3(t)$，即 $y_3(t) = A_{out}\, y_2(t) + b_{out}$。

神经元和突触动力学通过 py-torch-lightning 模块实现，从而确保在中央处理器和图形处理器等不同设备间的可重用性和可移植性。实例化模块时，会指定所需的模型类型，并据此初始化参数。初始化范围取自（Hasani 等，2021），详见补充材料。为缩小参数空间，部分参数被固定为某个值，不参与通过反向传播进行的训练。

由于所得常微分方程组是刚性的，常微分方程求解器的选择会对性能产生重大影响。本文所有实验均采用包含 10 次展开的显式欧拉求解器；与龙格-库塔方法（四阶龙格-库塔法或 Dormand-Prince 五阶方法）等更复杂的求解器相比，该方法在时间复杂度较低的同时，精度也足够好。

具有神经元激活和突触激活的突触层计算链分别如图4上部和下部所示。这里，x 和 y 分别表示突触前神经元和突触后神经元的状态。液态时间常数循环神经网络通过将激活与电势差项相乘来扩展连续时间循环神经网络突触（见图中下部）。突触激活的实现方式是：在整个计算图中将向量扩展为矩阵，同时将每个对应的矩阵乘法替换为逐元素（即 Hadamard）乘积。具体而言，这意味着中间结果也以矩阵形式表示，而在神经元激活的情况下，它们则是向量。

机器人实验

为探究生物物理突触在参数打包和线性门控方面的优势，我们进行了四项监督式时间序列实验：Walker2d 预测、Half-Cheetah 预测、Half-Cheetah 行为克隆，以及序列 MNIST 分类。

我们使用隐藏层含有 8、16、32 和 64 个神经元（全部相互连接）的 CT-RNN 和液体时间常数网络，分别采用神经元激活和突触激活，以及突触输入和线性输入，评估参数打包的优势。我们使用分别采用 S 形函数和双曲正切激活函数的 CT-RNN，评估门控优势。

我们还提供了长短期记忆网络、连续时间门控循环单元和具有 10 个增广维度的人工神经常微分方程的结果。由于线性输入映射是前两者的默认设置，我们采用了这种映射。由于缺少稳定化项，前文讨论的简单 NeuralODE 表现不如 CT-RNN，因此未在结果中展示。

尽管我们的结果主要针对机器人控制，且仅涉及液体时间常数网络和 CT-RNN，但我们认为这些结果适用于任何前馈或循环人工神经网络。

测试的连续时间循环神经网络和液体时间常数网络模型，按神经元激活和突触激活分别缩写为 NA-CT-RNN 和 NA-LTC，以及 SA-CT-RNN 和 SA-LTC。连续时间循环神经网络采用 S 形函数或双曲正切激活，分别以后缀 S 和 T 标记。液体时间常数网络始终采用 S 形函数激活，因为双曲正切不仅无法捕捉突触的生物物理特性，还会使网络变得非常不稳定。对于所有模型，我们都对线性输入映射和突触输入映射进行了实验，因为

后者更能体现感觉神经元的特性。所有实验均使用 Adam 优化器（Kingma 和 Ba，2014）。

学习 Walker-2D 的运动学。这项机器人任务受到（Rubanova、Chen 和 Duvenaud，2019）中的物理仿真启发，并通过使用（Brockman 等人，2016）中的 Gym环境实现。它用于评估各种循环神经网络学习运动学动力学的能力。

为了创建训练数据集，我们使用了一个非循环策略，该策略通过近端策略优化（PPO）（Schulman 等人，2017）和 Rllib（Liang 等人，2017）强化学习框架进行了预训练。为了提高任务复杂度，我们使用了该预训练策略在 4 个不同训练阶段（经历 500 至 1200 次 PPO迭代）的版本。随后，我们在 Walker2d-v2 OpenAI Gym环境和 MuJoCo物理引擎（Todorov、Erez 和 Tassa，2012）上进行了 400 条各含 1000 步的轨迹，收集了 17 维观测向量。请注意，由于策略是确定性的，因此无需将动作纳入训练集。我们将数据集的 15% 用于测试，10% 用于验证，其余部分用于训练。

我们将轨迹对齐为长度为 20 的序列，然后对每个模型分别训练三次，每次训练 200 个训练轮次。对于 8、16、32 和 64 个循环神经网络单元，均采用了这种做法。

图 5 展示了每个模型在三次运行中的测试损失中位数及其最小值和最大值，并列出了对应的神经元数量及参数数量。

在线性输入映射下，连续时间循环神经网络表现更好；而在突触输入映射下，自适应突触液态时间常数网络表现更好。生物物理突触的参数打包优势体现在：突触激活连续时间循环神经网络和自适应突触液态时间常数网络所能打包的参数数量，基本与神经元激活连续时间循环神经网络和 NA-LTC 相同，但神经元数量只有一半。门控优势则体现在连续时间循环神经网络-T型的表现优于连续时间循环神经网络-S型。液体时间常数网络在所有情况下都优于连续时间循环神经网络。当其单元数量等于自适应突触液态时间常数网络的神经元数量时，长短期记忆网络和连续时间门控循环单元通过更复杂的结构化单元概念实现了（甚至更高的）参数打包效率，但准确率未必更高。由于长短期记忆网络和连续时间门控循环单元默认采用线性输入映射，因此只出现在右图中。右图还展示了 ANODE。其表现与 NA-LTC 相当。

学习 Half-Cheetah 的运动学。与 Walker-2D 类似，我们学习了 Half-Cheetah 的运动学。

在这项实验中，我们使用通过截断分位数评论家（TQC）（Kuznetsov 等人，2020年）训练的控制器，收集了 100 条轨迹。这次仅使用了 stable-baselines zoo（Raffin，2018年）提供的一个版本，因此任务相较于前一项相对容易。与之前一样，每条轨迹由一系列 1000 个数据点组成，每个数据点包括 MuJoCo物理引擎生成的 27 维观测向量，以及控制器产生的 6 维动作向量。以下两项任务使用了相同的数据：

1. 运动学建模。在观察到前 20 个观测值后，预测下一个观测值。此任务不使用动作向量，因为观测值既作为输入，也作为标签。

2. 行为克隆。在观察到前 20 个观测值后，预测下一个动作。此任务中，观测值作为输入，而动作作为标签。

图 6 展示了 Half-Cheetah 运动学建模的结果，图 7 展示了 Half-Cheetah 行为克隆的结果。与之前一样，图中给出了测试损失的中位数、最小值和最大值。两幅图中的结果都与 Walker-2D 的结果呈现出非常相似的趋势。不过，在本例中，CT-RNN-T 的优势仅在突触输入映射方面显而易见。液体时间常数网络仍表现更优。在图 7 中，长短期记忆网络和连续时间门控循环单元的准确率略高于自适应突触液态时间常数网络，但代价是需要更多参数。

序列 MNIST 分类。 MNIST 数据集包含 70,000 张灰度图像，每张图像为 28×28 像素，内容为手写数字（LeCun，1998）。为了将此任务转化为序列任务，我们将图像转换为长度为 28 的序列，即按时间顺序将每一行向量作为输入。期望输出是一个独热编码向量，表示从 0 到 9 的整数。因此，训练模型时使用了交叉熵损失。

图 8 所示结果与之前相同，分别为每项三次运行的测试损失中位数、最小值和最大值。其趋势与前几幅图类似，但采用突触输入的液体时间常数网络稳定性较差，并且在神经元数量最多时未能充分收敛。人工神经元-连续时间循环神经网络和长短期记忆网络取得了最佳准确率。

讨论与结论

本文的主要目标是研究生物物理突触所具有的突触激活和线性门控优势，这些优势存在于液体时间常数网络中。为此，我们提出了以下问题：

- 在 LTC 中使用神经元激活会产生什么影响?
- 在 LTC 中取消线性门控会产生什么影响?

由此得到两个版本的 LTC，以及四个版本的 CT-RNN：输入分别为线性输入或突触输入，激活函数分别为 S 形函数或双曲正切。我们全面考察了这些网络在神经元数量逐渐增加时的准确率和参数压缩能力，并将其与 ANODE、长短期记忆网络和连续时间门控循环单元进行了比较。

我们观察到，对于相同的神经元数量，采用突触激活的 LTC 和 CT-RNN 只需一半数量的神经元，就能达到与采用神经元激活的 LTC 和 CT-RNN 基本相同的准确率和参数压缩能力。LTC 的线性门控进一步提高了准确率。我们还观察到，LTC 在准确率和参数压缩能力方面的优势与长短期记忆网络中的单元相当。不过，后者依赖于一种复杂得多的结构化单元概念。

我们声称，生物物理突触的优势适用于任何人工神经网络。然而，我们只对液体时间常数网络和连续时间循环神经网络明确展示了这些优势。因此，本文的完整版需要为这一主张提供依据。例如，对于前馈卷积神经网络，可以采用标准的卷积神经网络架构，并始终用突触激活替换其神经元激活。同样，对于长短期记忆网络和连续时间门控循环单元，可以将其循环连接改为突触连接，为每个突触使用一个双曲正切激活，而不是为每个细胞使用一个。

为清晰起见，我们尽量简化了神经常微分方程，将其右端变换限制为一层。不过，也可以使用更强大的变换，这或许能带来更好的结果。尽管如此，我们认为，前文讨论的优势仍然适用。

最后，生物物理突触或许也更适合稀疏网络，正如（Lechner 等人，2020）所提出的那样。不过，出于显而易见的篇幅限制，对这一主张进行全面研究只能留待未来工作。

参考文献

Alon, U. 2007. 网络模体：理论与实验方法。《自然评论》，8。Alvarez-Melis, D.; and Jaakkola, T. 2018. 借助自解释神经网络实现稳健的可解释性。载于*NIPS'18*论文集：第*32*届神经信息处理系统会议。加拿大蒙特利尔。Bishop, C. 1995. 《模式识别神经网络》。牛津，Claredon Press。Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym。*arXiv*预印本 *arXiv:1606.01540*。Chen, T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2018. 神经常微分方程。载于《神经信息处理系统进展》，6571–6583。Dupont, E.; Doucet, A.; and Teh, Y. 2019. 增广神经常微分方程。ArXiv，电子印本 1904.01681，arXiv:1904.01681。E, W. 2017. 基于动力系统的机器学习方案。《数学与统计通讯》，5：1–11。Funahashi, K.; and Nakamura, Y. 1993. 连续时间循环神经网络对动力系统的逼近。神经网络，6(6)：801–806。Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. 深度学习。麻省理工学院出版社。Hasani, R. 2020. 连续时间控制环境中的可解释循环神经网络。博士论文，维也纳。Hasani, R.; Lechner, M.; Amini, A.; Rus, D.; and Grosu, R. 2021. 液态时间常数网络。*AAAI*人工智能会议论文集，35(9)：7657–7666。He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. 深度残差网络中的恒等映射。*CoRR*，abs/1603.05027。Kandel, E.; Schwartz, J.; Jessel, T.; Siegelbaum, S.; and Hudspeth, A. 2013. 《神经科学原理》。McGraw-Hill Education / McGraw-Hill Medical，第5版。Kingma, D.; and Ba, J. 2014. Adam：一种随机优化方法。*arXiv*预印本 *arXiv:1412.6980*。Kuznetsov, A.; Shvechikov, P.; Grishin, A.; and Vetrov, D. 2020. 使用截断连续分布分位数评论家混合模型控制过高估计偏差。*arXiv:2005.04269 [cs,stat]*。Lechner, M.; Hasani, R.; Amini, A.; Henzinger, T.; Rus, D.; and Grosu, R. 2020. 可审计自主系统的神经电路策略。《自然·机器智能》，2：642–652。Lechner, M.; Hasani, R.; Zimmer, M.; Henzinger, T.; and Grosu, R. 2019. 用于可解释机器人控制的仿蠕虫神经网络设计。载于*2019*年国际机器人与自动化会议（*ICRA*）论文集。加拿大蒙特利尔。LeCun, B., Cortes. 1998. MNIST手写数字数据库，Yann LeCun、Corinna Cortes 和 Chris Burges。http://yann.lecun.com/exdb/mnist/。Liang, E.; Liaw, R.; Nishihara, R.; Moritz, P.; Fox, R.; Gon-zalez, J.; Goldberg, K.; and Stoica, I. 2017. Ray RLLib：一种可组合、可扩展的强化学习库。*CoRR*，abs/1712.09381。Poli, M.; Massaroli, S.; Yamashita, A.; Asama, H.; and Park, J. 2020. TorchDyn：神经微分方程库。*arXiv*预印本 *arXiv:2009.09346*。Raffin, A. 2018. 强化学习基线动物园。Rubanova, Y.; Chen, R. T.; and Duvenaud, D. 2019. 不规则采样时间序列的潜在常微分方程。*arXiv*预印本 *arXiv:1907.03907*。Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. 近端策略优化算法。*arXiv*预印本 *arXiv:1707.06347*。Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo：一种用于基于模型的控制的物理引擎。载于*2012*年*IEEE/RSJ*智能机器人与系统国际会议，5026–5033。IEEE。Wicks, S.; Roehrig, C.; and Rankin, C. 1996. 线虫自发缩回环路的动态网络仿真：基于行为标准对突触功能的预测。《神经科学杂志》，16(12)：4017–4031。C.imen, T. 2008. 状态依赖黎卡提方程（SDRE）控制：综述。第*17*届国际自动控制联合会世界大会论文集，6–11。

附录

我们的实现使用 torchdyn 软件包（Poli 等，2020），而该软件包基于 pytorch-lightning。它包含多个支持自动微分的常微分方程求解器。由于该软件包本身旨在实现自治神经常微分方程，因此需要一个小技巧来创建接收输入的连续时间循环神经网络和液态时间常数循环神经网络。自治神经常微分方程封装了一个非平凡函数（例如神经网络），用于在每个常微分方程求解器步骤计算导数 $\frac{dx}{dt} = F(x)$。为实现非自治行为 $\frac{dx}{dt} = F'(x, u)$，在每一步将输入传递给常微分方程求解器时，都将其附加到状态上，并将其导数设为零 $[\frac{dx}{dt}, 0] = F([x, u])$。因此，常微分方程求解器计算出的解，相当于将未经改变的输入附加到所需的最终状态上。

$$
\begin{aligned}
C\,\dot{x}_i(t) &= w_{li}\,(e_{li} - x_i(t)) + \textstyle\sum_{j=1}^{n} y_{ji}(t) \\
y_{ji}(t) &= w_{ji}\,\sigma(a_{ji}x_j(t) + b_{ji})\,(e_{ji} - x_i(t))
\end{aligned}
\tag{18}
$$

表2：循环层的参数及其初始化范围。 * 仅存在于液态时间常数循环神经网络中。

Name	描述	初始化
突触参数		
C *	膜电容	1（固定）
w	突触强度	0.01 - 1.0
b	突触中点	0.3 - 0.8
a	突触斜率	3 - 8
e *	反转电位	-1 或 +1
胞体参数		
e_l	静息电位	0（固定）
w_l	漏电导	0.01 - 1.0

表2展示了取自先前工作的LTC层参数初始化范围。由于如果不对参数加以约束，LTC动力学往往会迅速发散，因此训练期间施加了以下条件： $C \geq 0$, $w \geq 0$, $a \geq 0$。这些约束直接源自LTC所依据的式18电容器方程，其中电导（w）、电容（C_m）和突触门控（a）均不得为负。Hasani 等，2021中的通用逼近证明也采用了类似的约束。

类型描述符

NeuralODE 中使用的导数形式通过传入一个类型描述符进行配置，该描述符是一个格式如下的字符串：

连续时间循环神经网络[w 模式]激活[因子][递归类型]输入模式 [lis]

无论哪种情况，导数均使用以下公式计算：$\dot{x} = Synapses(x, x) + Input(u, x) - decay(x)$。类型描述符决定了各个项的计算方式。

E、 b 和 α 的大小取决于所使用的循环类型：

表 3：类型描述符组件及其生成的公式

w 模式	$Synapses(x,\ y)$
None	$a\ \sigma(wx + b)\ *Factor(y)$
r	$w\ \sigma(x + b)\ *Factor(y)$
v	$w\ \sigma(a(x + b))\ *Factor(y)$
act	$\sigma(x)$
'sigm'	$1/(1 + e^{-x})$
'双曲正切'	$tanh(x)$
因子	$Factor(x)$
None	1
'*'	$(1 - x)$
'+'	$(e - x)$
递归类型	
None	神经元激活
's'	突触激活
输入模式	$Input(u,\ x)$
'线性'	$Iu + b_i$
'突触'	$Synapses(u,\ x)$
lis	$decay(x)$
None	τx
'lis'	$\tau(x - x_0)$

循环类型	**参数形状**
神经元（无）	（模型大小）
突触（'s'）	（模型规模，模型规模）

不同的 w 模式可用于调整突触的参数化方式，激活决定所使用的激活函数，因子用于区分液体时间常数网络和 CT-RNN，递归类型用于在神经元与突触激活之间切换，输入模式决定输入模式，而

表 1：INSERT 实验的结果

模型	n = 8		n = 16		n = 32		n = 64	
	# 参数	均方误差 $\times 10^{-2}$	# 参数	均方误差 $\times 10^{-2}$	# 参数	均方误差 $\times 10^{-2}$	# Par	均方误差 $\times 10^{-2}$
ANODE	1663	0.62 ± 0.04	2263	0.46 ± 0.01	3463	0.40 ± 0.02	5863	0.31 ± 0.01
CT-GRU	5139	0.84 ± 0.05	12427	0.50 ± 0.02		-		-
LSTM	1427	0.94 ± 0.00	3339	0.53 ± 0.02	8699	0.38 ± 0.00		-
NA-CT-RNN 线性	555	$1.10 \pm nan$	1211	$0.60 \pm nan$	2907	0.41 ± 0.01	7835	0.45 ± 0.02
NA-CT-RNN 突触	601	1.04 ± 0.04	1249	0.58 ± 0.02	2929	0.45 ± 0.08	7825	0.57 ± 0.04
NA-LTC 线性	571	1.15 ± 0.07	1243	0.56 ± 0.01	2971	$0.40 \pm nan$	7963	0.32 ± 0.07
NA-LTC 突触	617	$1.08 \pm nan$	1281	$0.67 \pm nan$	2993	0.52 ± 0.01	7953	0.61 ± 0.06
SA-CT-RNN 线性	667	1.01 ± 0.02	1691	0.54 ± 0.00	4891	$0.36 \pm nan$		-
SA-CT-RNN 突触	1091	0.88 ± 0.04	2539	$0.61 \pm nan$	6587	0.50 ± 0.04		-
SA-LTC 线性	947	$0.93 \pm nan$	2379	0.46 ± 0.02	6779	0.33 ± 0.03		-
SA-LTC 突触	1371	0.82 ± 0.03	3227	0.47 ± 0.03	8475	$0.28 \pm nan$		-

lis 用于需要将初始状态（= 泄漏电势）设为可学习参数的情况。

循环类型决定循环连接的实现方式：

- 神经元：对应连续时间循环神经网络模型。连接可视为仅具有线性特性，特定细胞的所有输入突触共享相同的偏置，并且在应用激活函数之前进行求和。

- 突触：每个突触都有单独的偏置 b、缩放因子 w和参考电势 E。计算各个突触激活后，还会进行一次额外求和。$\dot{x} = \sum[Synapses(x, x)] + Input(u, x) - decay(x)$

示例

- 普通连续时间循环神经网络：连续时间循环神经网络 双曲正切 线性
- SA-LTC：连续时间循环神经网络 vsigm+s 突触