



# 短期科学研究探索 阶段一报告

HAN YAOTING

3220101611@zju.edu.cn

数学科学学院

脑机接口国家重点实验室

2025 年 6 月 15 日



# 目录

目录

内容简介

SNN 的基本情况

SNN 的手写数字分类

附录

参考资料



# 主要内容

- SNN 的相关基本概念
- 了解熟悉 SNN 中的一种神经元、架构、编解码和学习策略
- 搭建一个简单的 SNN 网络完成对 MNIST 的训练和评估
- 粗略阅读一篇前沿文章

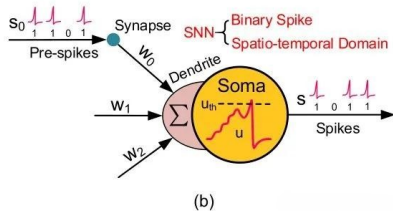
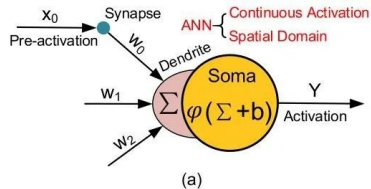


# 整体思路

- 了解 SNN 的基本情况
- 通过综述文章了解 SNN 的相关研究进展（神经元模型、架构等）
- 通过 snnTorch 对每个版块熟悉一种结构
- 使用 PyTorch 框架自己实现一个 SNN 网络并完成一个简单的任务
- 阅读一篇前沿文章，了解 SNN 的最新研究进展

# SNN 简介

- SNN 将 ANN 中的“仿射变化 + 激活函数”的人工神经元替换为了生物合理性更强的“微分方程刻画的膜电位变化 + 脉冲触发”的类生物神经元。
- SNN 是为了更好地模拟生物大脑中的神经元传递信息的方式，使之相比 ANN 更加高效节能，且在生物角度上更加合理。



# SNN 的结构简述

- **信息编码**：常见的编码方式分为速率编码和时间编码两种.
- **神经元**：SNN 中最基本的结构，基于类 R-C 电路微分方程给出神经元膜电位变化，在膜电位达到阈值后触发一个脉冲.
- **神经回路**：连接若干神经元的网络，它一定是一个前馈网络，但可以通过加入前/反/侧馈抑制来实现更多的功能.
- **学习策略**：一般有 ANN2SNN 的间接训练和有/无监督直接训练两种大的策略.

# 信息编码

- **速率编码**：基于脉冲发放频率来表示信息，脉冲发放频率与像素强度成正比，常见的有计数/密度/群体频率编码，例如，计数频率编码的表示如下：

$$v = \frac{N_{\text{spike}}}{T}$$

- **时间编码**：通过神经元接收到输入刺激后首次发放脉冲的时间来量化输入刺激的强度，常见的有首达时间/排名顺序/相位编码，例如，首达时间编码的表示如下：

$$t_{\text{spike}} = \min\{t \geq 0 \mid V(t) \geq V_{\text{thresh}}\}$$

- 两种编码理论上是在大脑中协同工作的，速率编码倾向于刻画长时持续的刺激行为，时间编码倾向于刻画单次发生的刺激行为。

# 简化的 Leaky Integrate-and-Fire (LIF) 神经元

常用的神经元模型，它假设神经元的膜电位是由一个积分器和一个漏电导组成的，当膜电位超过一定的阈值时，神经元会被激发，并产生一个动作电位。然后，膜电位会被重置并进入不应期。其数学定义如下：

$$C \frac{dV}{dt} = -g_L(V(t) - E_L) + I(t)$$

如果我们取膜时间常数  $\tau = RC$ ，则上式变为

$$\tau \frac{dV_{\text{mem}}}{dt} = -[V_{\text{mem}}(t) - V_{\text{rest}}] + RI(t)$$

上述模型经过简化和向前欧拉方法后得到

$$U(t + \Delta t) = \left(1 - \frac{\Delta t}{\tau}\right) U(t) + \frac{\Delta t}{\tau} I_{\text{in}}(t) R$$

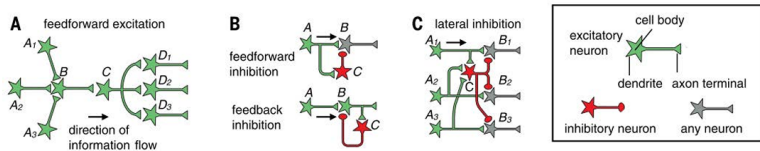
同时在触发后膜电位会重置，结合后得到表达式：

$$U[t + 1] = \underbrace{\beta U[t]}_{\text{decay}} + \underbrace{WX[t + 1]}_{\text{input}} - \underbrace{S[t] U_{\text{thr}}}_{\text{reset}}$$



# 神经回路

- **前馈兴奋**：如图 A，在每一层中，每个神经元通过汇聚连接从多个前突触节点接收输入，并自身发散连接到多个后突触等效物。
- **前馈/反馈抑制**：如图 B，FB 抑制指的是一层前突触兴奋性神经元会刺激投射回前突触兴奋性层的后突触兴奋性神经元以及抑制性神经元，反过来，FF 抑制指的是后突触抑制性神经元群体连接到后突触神经元群体的情况。
- **侧抑制**：如图 C，包括一些神经元呈现以减少并行通路活动的容量。通过这种方式，可以在减少传输不太相关脉冲的同时激发某些活动。



## 替代梯度法的 BP 策略

这里我们考虑有监督学习的直接训练方法，ANN 中的常见学习算法是对损失函数做关于参数矩阵的进行梯度下降以最小化损失函数，通过 BP 更新网络中的权重，但对 LIF 神经元，由链式法则得到

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial S} \underbrace{\frac{\partial S}{\partial U}}_{\{0, \infty\}} \frac{\partial U}{\partial I} \frac{\partial I}{\partial W}$$

其中 S 和 U 之间关系由下 Heaviside 阶跃函数替代

$$S[t] = \Theta(U[t] - U_{\text{thr}})$$

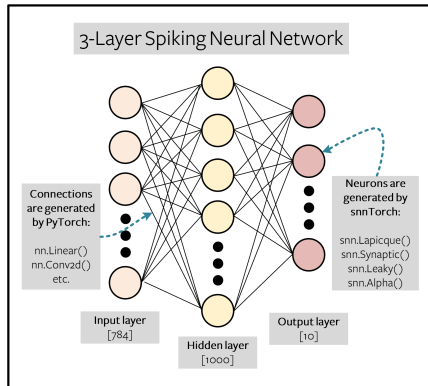
它的导数是狄拉克函数，显然在定义域上非 0 即无穷，无法直接进行学习，一种可行的解决方案是在 FP 时保持 Heaviside 函数的不变，在 BP 时使用它的平滑函数去拟合导数结果，常用的函数是正反切函数，其导数如下：

$$\frac{\partial \tilde{S}}{\partial U} \leftarrow \frac{1}{\pi} \frac{1}{(1 + (U\pi)^2)}$$

这就使得 BP 时的导数计算成为可能。

# SNN 结构简介

一个 3 层全连接 SNN，每层神经元数量为 784、1000、10，使用归一化静态 MNIST 特征作为输入进行类速率编解码，简化的 LIF 神经元，前馈兴奋神经回路，替代梯度的反向传播有监督学习策略，网络结构大致如下：



# 训练结果

参考 snnTorch 用户手册，在 copilot 辅助下使用 snnTorch 框架和 PyTorch 框架分别搭建上述网络，在 A800 上用 5 个 epoch 训练得到结果如下：

模型名称	训练时间 (秒)	评估时间 (秒)	准确率
tutorial_snn_MNIST	143.47	1.54	97.47%
simple_snn_MNIST	98.18	0.51	97.45%

在基础配置完全相同的情况（归一化策略、epoch 数量等）下，二者的表现接近，训练与评估时间上的差别是由于代码实现不同（初始化方法、加速与否等）导致的。

额外地，simple\_snn\_MNIST 的 forward 只返回最后一层的脉冲均值，其损失计算并不在每个时间步计算累加。

# Loss 曲线

我们对训练过程中的 Loss 曲线绘制如下：

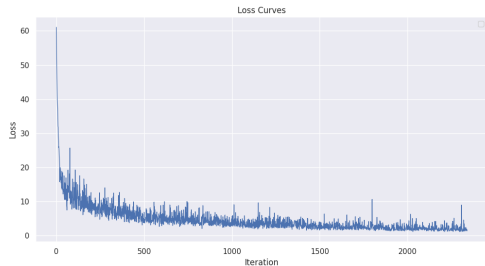


Fig: snnTorch 搭建网络损失函数

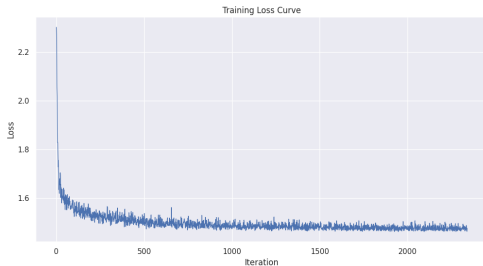


Fig: PyTorch 搭建网络损失函数

二者在误差允许范围内是相似的且下降速度符合预期。

# 改进空间





- 网络对 MNIST 针对性极强，需要进一步调整网络层数、宽度和数据输入策略.
- 本质上这只是将 ANN 中的神经元和学习策略做了替换得到的，生物合理性不高，需要进一步调整数据的编解码方式.
- 完善 SNN 的相关功能，扩展神经元类型、学习策略等.



# 附录

- 源文件：  
存储在 github 项目 [https://github.com/DrypotTofu/BMI\\_project](https://github.com/DrypotTofu/BMI_project) 中。
- 清单：
  - Transformer ?
  - 混合编码形式 ?
  - ANN2SNN 的转化过程 ? STBP ? TDBN ? FT ?
  - 神经元连接的可生长性 ?

# 参考资料

-  脉冲神经网络 (SNN) 概览, 脉冲神经网络概览, 2023,  
<https://www.bilibili.com/video/BV1dG4y1D7b2/>
-  João D. Nunes, Marcelo Carvalho, Diogo Carneiro, and Jaime S. Cardoso, Spiking Neural Networks: A Survey, 2022
-  Jason K. Eshraghian, Max Ward, Emre O. Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu, Training Spiking Neural Networks Using Lessons From Deep Learning, 2023
-  Mengting Xu, De Ma, Huajin Tang, Qian Zheng, and Gang Pan, FEEL-SNN: robust spiking neural networks with frequency encoding and evolutionary leak factor., 2025