**Nｅｕｒｏｎ**

**---姜浩**

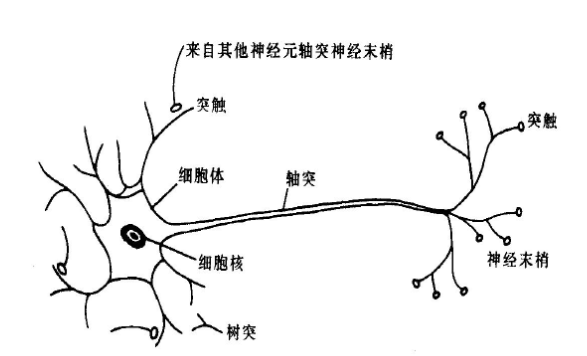
**（2020/9/18）**

**基本的神经元模型：**

（１）单个神经元的组成及工作方式：

* 树突：神经元的“输入”组织
* 轴突：神经元的“输出”阻值，把输出信号传送给其他神经元
* 胞体：神经元的“CPU”。完成非线性信号的处理，当输入信号超过一定的阈值，则产生一个输出信号

两个神经元之间的连接部分叫突触，发送信号的神经元为突触前神经元，接收信号的神经元为突触后神经元。



生物神经元示意图

（２）大脑神经元的工作原理

神经元之间通过电脉冲（动作电位或ｓｐｉｋｅ）的形式传输信号。

神经元发出的脉冲序列称为ｓｐｉｋｅ　ｔｒａｉｎ。ｓｐｉｋｅ是神经元传输的信息的基本单元，单个脉冲不包含信息，但序列中脉冲的数量和各脉冲中传输时刻含有信息。

突触传输信息的过程：化学突触和电突触，通常通过化学突触传递信号。

突触整合：使突触后神经元发生兴奋的突触为兴奋性突触，反之称为抑制性突触。通过测量细胞膜内外电位差，可以看到ｓｐｉｋｅ对突触后神经元的影响，如果兴奋性突触活动强度总和超过抑制性突触活动强度总和，并达到一定阈值，就能使该神经元的轴突起始段发生动作电位，产生神经冲动。出现神经冲动时，该神经元呈现兴奋，反之则表现为抑制。

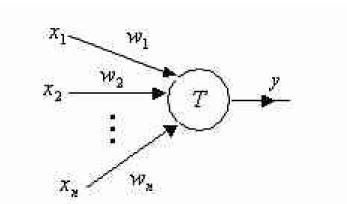
（３）神经元模型

* **MP模型**：由ＭｃＣｕｌｌｏｃｈ和Ｐｉｔｔｓ提出

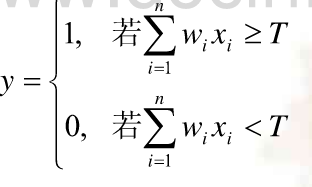
基本思想：神经细胞的工作方式是要么兴奋，要么抑制，并引入了硬极限函数来模拟这种机制，该函数形式后来常被其他神经网络（多层感知机，离散Ｈｏｐｆｉｅｌｄ网络）所采用。

具体实现：由于神经元之间的信号连接强度取决于突触状态，因此在ＭＰ模型中，神经元的每个突触的活动强度用一个固定的实数即权值模拟。于是每个神经元模型都可以从数十甚至数百个其他神经元接收信号，产生神经兴奋和冲动；同时，在其他条件不变情况下，不论何种刺激，只要达到阈值以上，就能产生一个动作电位。但如果输入总低于阈值，则不能引起任何可见的反应。

ＭＰ模型：



神经元输出为：



ＭＰ模型功能：

1. 可以实现与、或、与非、或非等二值逻辑运算（但不能实现异或操作）
2. 该模型曾因说明了人工神经网络可通过简单的计算产生相当复杂的行为，从而引起极大的轰动。

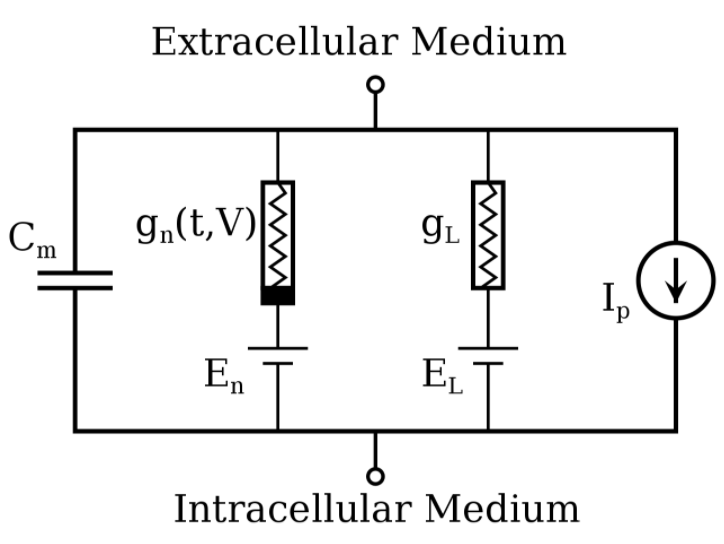
ＭＰ模型缺点：

1. 与实际神经元的工作机制还有很大差别
2. 只有静态神经元，结构固定，没有给出权值调节机制，因此缺乏一个关键性的要素，即学习能力

* **Hodgkin Huxley Model（HH模型）**

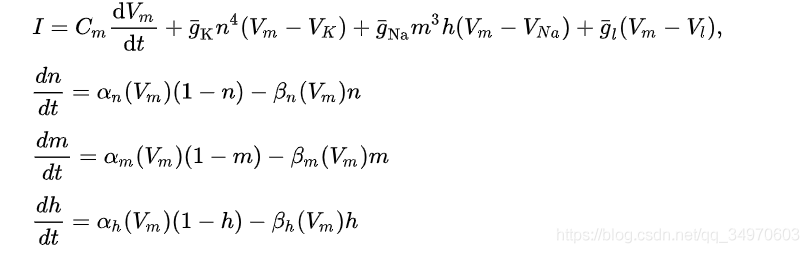
通过对乌贼的巨型轴突进行实验，运用数学方法分析以及三种离子描述了细胞膜电势的动态变化，是其他简化神经元模型的基础。

电路图如下；



Hodgkin-Huxley(HH)模型可以看作是rc电路模型的扩展，该模型除了泄漏通道外还包含钠(Na)和钾(K)通道。离子通道被建模为与电池串联的电阻器。电池表示特定离子的平衡电势，电阻器反映通道对特定离子的渗透性。与泄漏通道的固定电阻相比，Na和K通道的电阻有自己的动态特性，这取决于跨膜的电压。

更精确地说，对于给定的膜电位，每个通道都有相应的开启和关闭速率。这些通道的打开还是关闭是由霍奇金根据经验建立的函数和函数来描述的。这些和函数可以用在微分方程中来描述开放通道的比例如何随时间变化，

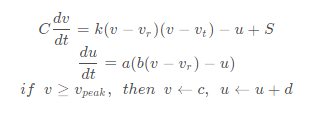
常微分方程：

由于HH模型需要使用4个常微分方程进行表示模型，故计算较为复杂，很难进行大规模仿真。但其精确地描绘出膜电压的生物特性，能够很好地与生物神经元的电生理实验结果相吻合

* **Izhikevich Model**

Izhikevich博士于2003年提出，使用二叉树对HH数学上的简化

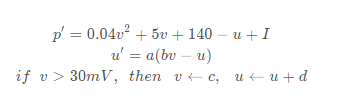
使用分叉方法将HH模型简化为二阶常微分方程，Izhikevich在上述基础上加入恢复变量，使得神经元具有更加丰富的动态特性，而没有显著提高计算复杂度。



各参数意义：

u是膜电位恢复变量，主要用于统计火花钾离子和钝化钠离子的流动规律，并对膜电位起负反馈作用。

简化模型：

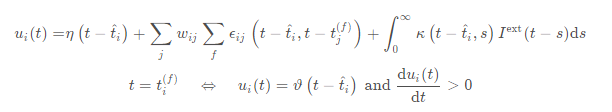


0.04v2+5v+140的取值是为了将膜电势限制在mv水平，时间限制在ms水平，选用其他参数亦可

* **Spike Response Model**

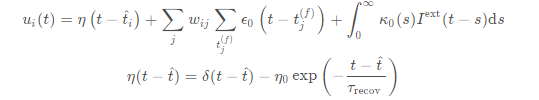
标准的SRM模型：标准的SRM主要考虑两个部分：一是脉冲发放后的膜电势变化，二是对外部刺激电流（电极电流或突触前神经元发放的脉冲的突触后电流）。

在标准的模型中，阈值使用恒定值：



简化模型ＳＲＭ：

简化后的SRM模型表述如下



* **Leaky Intergrate and Fired Model**

Leaky Integrate and Fire neurons简称LIF模型，是一种对HH模型的最基础的简化。

目前使用最多的是Leaky　integrity-Fire（LIF）模型。对于这类神经元模型来说，输入信号直接影响的是神经元的状态（膜电位），只有当膜电位上升到阈值电位时，才会产生输出脉冲信号。

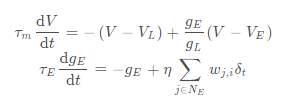
Leaky指泄露，表示如果神经元输入只有一个时，不足以让膜电势超过阈值，由于细胞膜不断进行膜内外离子交换，膜电势会自动发生泄露逐渐回落到静息状态；

Integrate指积分，表示神经元会接收所有与该神经元相连的轴突末端（上一个神经元）到来的脉冲；

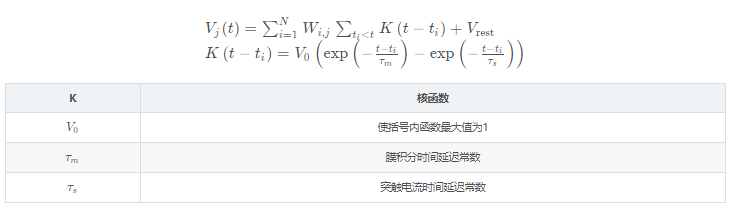
Fire指激发，表示当膜电势超过阈值时，神经元会发送脉冲。神经元发送脉冲后会进入超极化状态，然后是不应期（Refractory Period），在不应期内即使给予刺激也不会反应，即神经元不再接收刺激，保持静息电位。

LIF神经元在表示上有多种形式，以下为两种常见表示方式：

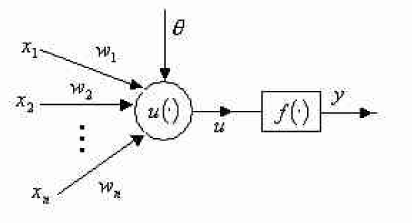
A:





B、

* **一般神经元模型**



神经元输入：ｘ＝（ｘ１，ｘ２，．．．，ｘｎ）＾Ｔ

可调权值：ｗ＝（ｗ１，ｗ１，．．．，ｗｎ）＾Ｔ

偏移或阈值：θ

基函数：ｕ（．）

激活函数：ｆ（．）

其中基函数类型包括：线性函数（较常采用），距离函数（主要用于RBF网），椭圆基函数，硬极限函数

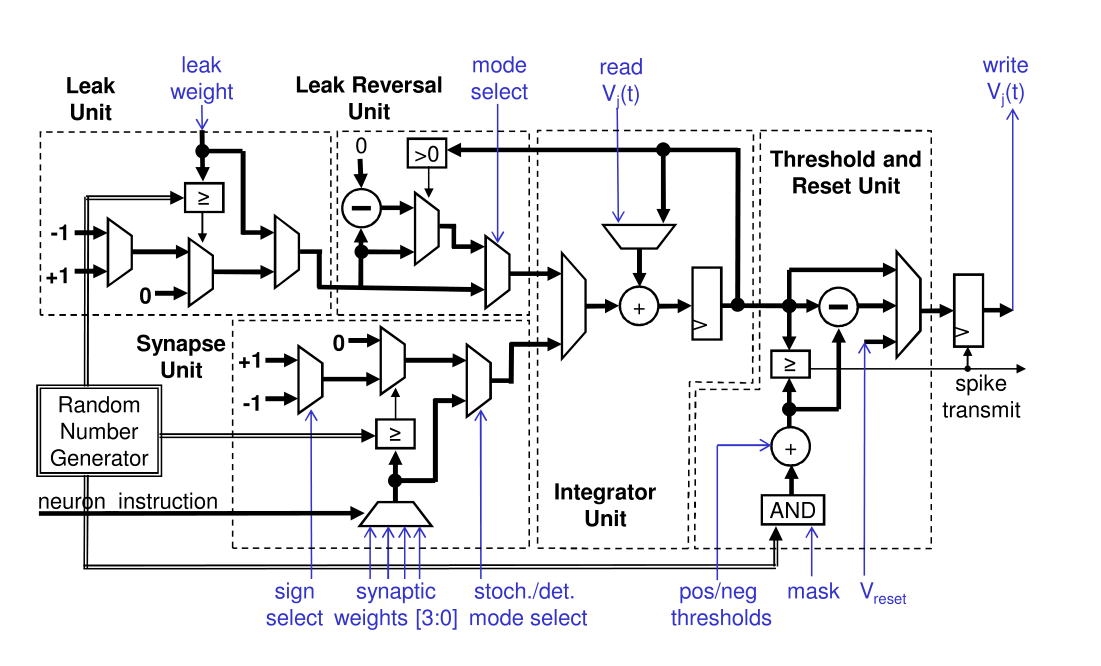
其中激活函数类型包括：线性函数，饱和线性函数，ｓｉｇｍｏｉｄａｌ函数，径向基函数。

**相关论文：**

1. **[TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip]**

**Neuron：**

神经元块是TrueNorth核的主要计算单元。在扩充的IF神经元模型的基础上，实现了一个双重随机和确定性神经元。简化复杂算法和逻辑运算的实现，神经元块采用同步逻辑实现，采用标准的专用集成电路(ASIC)设计流程。



神经元模块框图

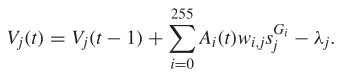
模式：同步

事件驱动：神经元块仅在两个条件成立时接收指令和时钟脉冲:存在活动的输入尖峰和相关联的突触活动。因此，神经元块以事件驱动的方式进行计算。

神经元块仅在两个条件成立时接收指令和时钟脉冲:存在活动的输入尖峰和相关联的突触活动。因此，神经元块以事件驱动的方式进行计算

神经元通过发送尖峰信号相互交流。可以使用尖峰的频率、时间和空间分布来编码所传送的数据。每个神经元整合其传入的尖峰并更新其膜电位。每个神经元的输出都连接到与之通信的轴突的输入缓冲区

上述框图描述了神经元块的五个主要元素，突触输入单元为四种不同的权重类型实现随机和确定性输入。泄漏和泄漏反转单元为神经计算的动力学提供一个恒定的偏差(随机或确定性)。积分器将前一个标记的膜电位与突触输入和泄漏输入相加。阈值和复位单元将膜电位值与阈值进行比较。如果膜电势值大于或等于阈值，神经元块将重置膜电势并传递一个尖峰事件。随机数发生器用于随机泄漏、突触和阈值函数。SRAM核位于模块的外部，存储突出连接性和权重值、泄露值、阈值、配置参数以及膜电位。在神经计算周期的开始和结束时，膜电位从核心SRAM加载，然后写回核心SRAM。（蓝色部分表示从SRAM核到SRAM核的输入/输出参数）

第t个神经元的膜电位Vj(t)：

调度器接收到spike后，逐个处理256个神经元

当令牌控制器完成对一个神经元的所有尖峰的积分后，它会向神经元块发送几个附加指令:一个用于减去泄漏λi，另一个用于检查尖峰条件。如果最终的膜电位Vi(t)高于可编程阈值，神经元模块会为路由器生成一个尖峰包(G)。路由器依次将新的尖峰数据包注入网络。最后将更新后的膜电位值(E)写回核心SRAM，完成单个神经元的处理。

**优点**：设计周期款快速灵活，采用事件驱动可以有效的最小化有效功耗。此外，为多个逻辑神经元复用单个神经元电路减少了块的面积，同时增加了并行性并降低了静态功耗。

**调度器**：调度器的功能是将传入的尖峰信号以适当的节拍传递到右侧轴突。

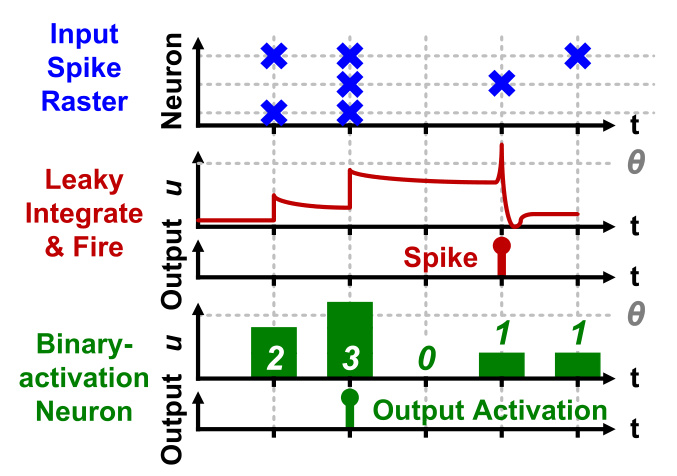
调度器包含尖峰写入/读取/清除控制逻辑和一个专用的12晶体管(12T) 16 × 256位SRAM。SRAM的256条位线对应于核心的256条轴突，而16条字线对应于16个传送节拍。

**2、[A 4096-Neuron 1M-Synapse 3.8-pJ/SOP Spiking Neural Network With On-Chip STDP Learning and Sparse Weights in 10-nm FinFET CMOS]**

采用的是LIF神经元，支配LIF神经元状态的方程为：



U（ｔ）代表在时间ｔ时的膜电位；λ是泄露因子，且０＜λ＜１，ｓ是输入脉冲，ｗ是脉冲权重，B是可训练偏差，I是SNN的主要输入。

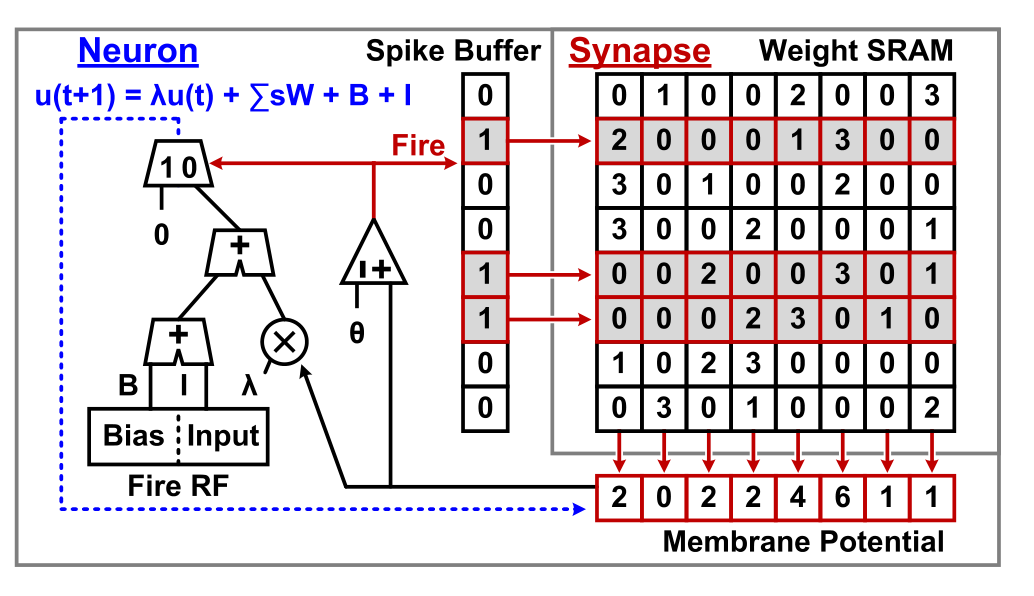


一个输入脉冲光栅的LIF响应

上图解释：上图为LIF神经元相应。膜电位对所有输入的spike的净效应进行累加，并在每个时间步长中保留。同时LIF神经元，在所有输入脉冲的净效应跨越时间导致膜电位超过一个阈值时，LIF神经元就会出现峰值。

这与二元激活神经元形成对比，其中膜电位(部分和)在每个时间步的末尾被重置

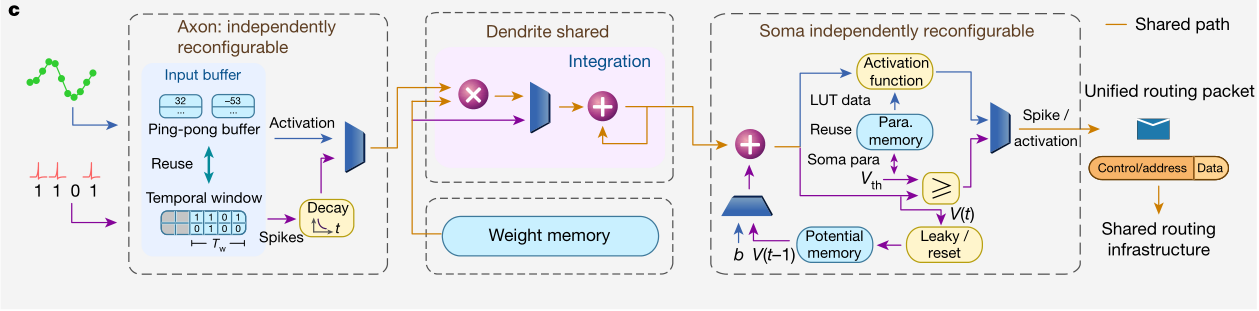
LIF神经元将加权峰值整合到膜电位上，并在输入的净效应超过阈值时产生输出峰值。



在每一个时间步骤中，神经元通过将权重矩阵与稀疏脉冲向量相乘来进行积分。通过使用地址事件表示(AER)将脉冲信号发送到突触，作为脉冲信号神经元的地址列表。对于非尖峰神经元，AER不发送任何数据以减少低尖峰率下的带宽需求。权重被访问的神经元只有尖峰，减少了计算和内存读取的数量。权重被整合到目标神经元的膜电位上。

神经元组每个时间步执行泄漏和发射操作一次。每个神经元将其膜电位与阈值进行比较。如果电位超过阈值，神经元就会出现峰值，其电位就会被重置为0。否则，造成输入尖峰效应的潜在泄漏会随着时间的推移而减弱。然后，神经元加入可训练偏差和SNN的主要输入。偏置实现了内在可塑性稳态来设置平均峰值速率。稳态电路监控峰值活动，并使用带有可编程减量的上/下计数器对偏置应用负反馈。

3、天机



跨范式神经元方案：设计**突触synapse和树突dendrite是共享的**，而**轴突axon和胞体soma可以独立地重新配置**。

**设计理念：**天机集成电路采用多核结构，并行性强。每个FCore包括几个块：轴突、树突(带突触)、胞体和路由器。使我们能够实现混合范式的关键设计是：独立可重构的轴突和胞体。轴突和胞体可以独立配置成不同的模式。轴突根据其模式配置接收和组织SNN输入或ANN输入。类似地，胞体根据其模式配置生成SNN输出或ANN输出。当轴突和胞体配置成相同的操作模式(ANN或SNN)时，FCore分别以纯ANN或SNN模式工作，我们称之为单范式FCore。当轴突和soma配置成不同的工作模式时，FCore处理ANN输入并触发SNN输出，或处理SNN输入并生成ANN输出;我们称之为混合FCore。

**树突整合共享**：用于处理SNN输入和ANN输入的树突状积分共享相同的计算器(乘数和累加器)，尽管它们有不同的处理操作和风格。在每个时间阶段，树突在处理ANN输入时执行高强度MACs。当处理SNN输入时，当积分时间窗长度大于1时，树突也执行MACs；如果这个时间窗口小于1，则树突只执行加法操作，并绕过乘数；如果没有收到尖峰，树突将跳过所有操作。

**补充：**

* **浮点：小数点非固定的数，可表示数据范围较广，整数，小数都可表示。包含float，double；**
* **定点：小数点固定，可表示整数，小数。int本质是小数点位于末尾的32位定点数而已；**

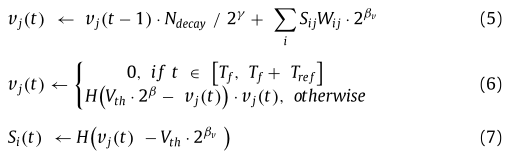
浮点数和定点数的转换是一种映射。将较为密集的数据空间（F32）映射到较为稀疏的空间（int8）；

**4、达尔文NPU【Darwin: A neuromorphic hardware co-processor based on spiking neural networks】**

基于泄漏整合和激发(LIF) SNN模型的高度可配置的神经形态硬件协同处理器——达尔文神经处理单元(NPU)。

为了用数字逻辑实现模型，需要有一个离散版本的LIF模型。

为了降低计算密度，需要将浮点变量转换为定点整型变量。通过合并参数简化状态更新：



S ij ={0,1}表示神经元i是否在时间步长t触发脉冲

Nleak：泄露常数

Ndecay：衰减常数

Wij：等价突触权重

Vth是触发阈值

H(x)是单位阶跃函数，x大于等于0时为1，否则为0

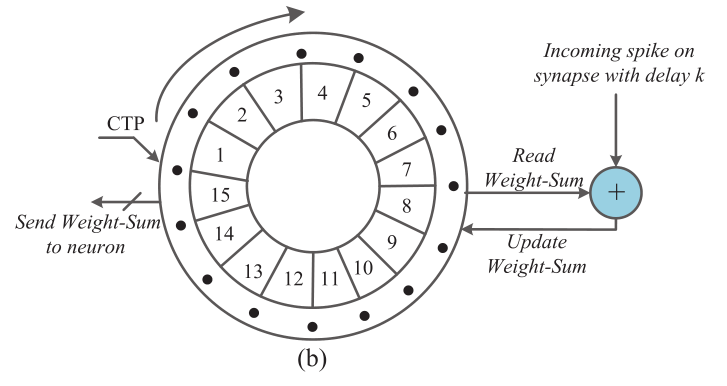
Ｎｄｅｃａｙ＝Ｎｌｅａｋ．２＾γ

由于膜电位Vj(t)和突触权重具有显著的不同动态范围，因此在浮点定点转换中分别应用了不同的缩放因子βv和βw，差值为βd。  
D:\win10\TEMP\1600247088(1).png

形成一组核方程，由NPU执行，以执行LIF神经元网络的模拟。

**神经元状态的更新**：每个神经元根据方程式进行状态更新。神经元首先从局部状态中获取更新生物神经元的当前状态，然后从权和队列中获取当前步长的权值和。如果产生一个输出spike，则以AER数据包的形式发送到spike路由器。

权值和队列中的每个条目都包含一个权值和的中间结果，该结果要在一定的延迟后发送给神经元。假设最多可移植15个时延值，权值和队列由15个条目组成，每个条目对一个范围内的时延值[1,15]做出响应。第k个条目存储输入突触权重的总和



权和队列的循环缓冲区设计

在每一个时间步长，当前的时间步长指针(CTP)被移动一个条目，并且CTP之前指向的条目的内容被发送到神经元。

**６、[A 2.56-mm2718GOPS Configurable Spiking Convolutional Sparse Coding Accelerator in 40-nm CMOS]**

一个可配置的spiking卷积稀疏编码加速器

【未涉及具体的ｎｅｕｒｏｎ单元的设计】

**７、[A 75-µW, 16-Channel Neural Spike-Sorting Processor With Unsupervised Clustering]**

第一个多通道尖峰排序DSP芯片，包括在线，无监督聚类。

【未涉及具体的ｎｅｕｒｏｎ单元的设计，但是涉及到调度问题】

[调度]

**８、[A Digital Neurosynaptic Core Using Embedded Crossbar Memory with 45pJ per Spike in 45nm]**

制作了一个神经突触核：它拥有256个数字集成和激活神经元和一个1024×256位的SRAM交叉条存储器，用于突触，使用IBM的45nm SOI进程。

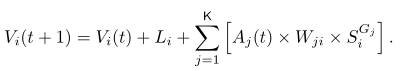
轴突是神经元的输出线，突触是神经元之间的连接

在每个时间步长t，每个轴突j呈现一个活动位**Aj(t):** 表示其对应神经元在前一时间步中是否触发。

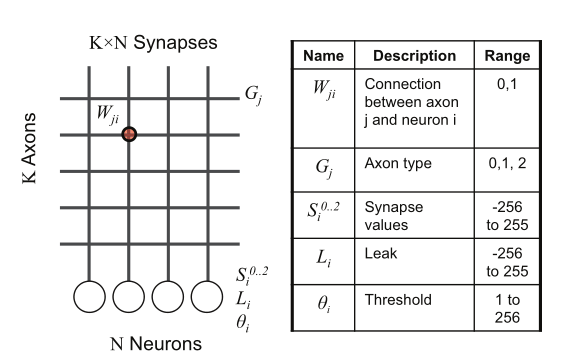
神经元i将来自类型为Gj的轴突j的突触输入∈{0,1,2}称为SGj, 神经元i从j轴突接收到如下输入:

D:\win10\TEMP\1600257168(1).png

在本文章中的神经元，使用了一个由膜电位V (t)、泄漏L、阈值θ和三个突触值S0、S1、s2对应不同类型的轴突参数化的泄漏整合和激发模型(单个室)。神经元i的膜电位在每个时间步更新为：



当V (t)超过它的阈值后，神经元产生一个尖峰，它的电位被重置为0。同时还强制负膜电位在每个时间步长结束时压缩回0。



神经突触核心的基本组成部分是轴突、突触和神经元

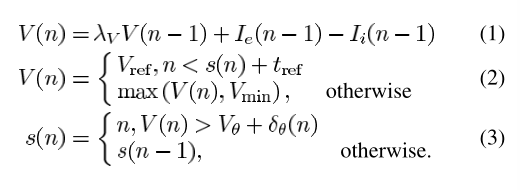
在核心中，信息从轴突流向由突触调节的神经元，核的结构由K个轴突组成，通过K×N个二值突触连接到N个神经元。我们将轴突j与神经元i之间的连接称为Wji

核心由轴突组成，以行表示;树突，表示为列;突触，表示为行-柱连接;以及从树突接收信息的神经元。描述核心的参数具有指定的整数范围。

**９、[Implementing Spiking Neural Networks for Real-Time Signal-Processing and Control Applications: A Model-V alidated FPGA Approach]**

提出了两个版本的硬件处理体系结构，用于建模大型的泄漏-集成-触发(LIF)神经元网络;与第一个版本相比，第二个版本提供了性能增强特性。两种版本的架构都使用定点算法，并使用单一的现场可编程门阵列(FPGA)实现.

神经元模型：这里解释的模型是离散的，如下所示



方程（1）表示的是突触后神经元j的膜差分方程，λv是衰减常数，

Ie是被整合的兴奋性突触电流。（post）

Ii是被整合的抑制性突触电流 。(post)

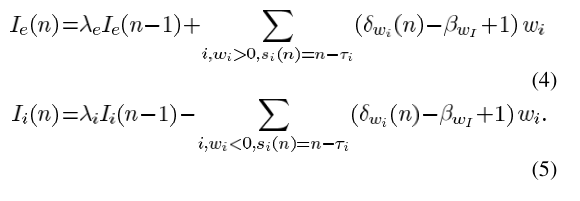
Vθ:膜阈值电压

δθ：噪声贡献的膜阈值电位，取自高斯噪声分布。附加噪声

对于（2），当动作电位或脉冲事件开始时，膜电位在设定的一段时间tref内保持在复位电位 ，表示绝对的不应期。在这段时间之后，膜的更新过程继续。

（3）是动作电位条件：V（n）超过附加噪声贡献和膜阈值电位

突触后兴奋性和抑制性电流对神经元的贡献的差分方程如(4)和(5)所示：



λx:衰减常数 δwij（n）瑞利分布的噪声贡献。

根据本实验中使用的器件，使用16-b的整数。**定点算法需要比特移位操作来最大化范围**，同时保持正确的缩放

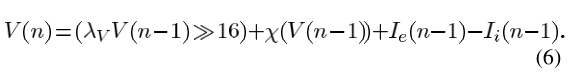
**定点数表示法：**将方程（1）中的浮点膜更新为下（6）所示：

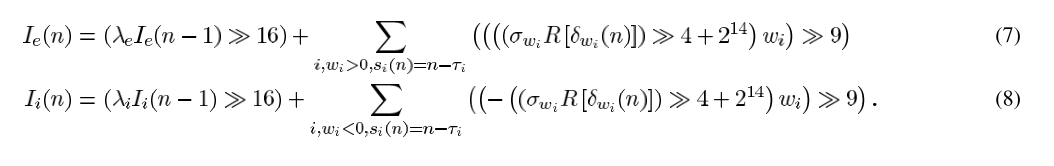
乘法首先存储在一个32-b的结果字段中，然后使用一个16-b的右移位操作<<16进行截断和重新调整，还有一个提取符号位，返回1或0值的函数x（.）。

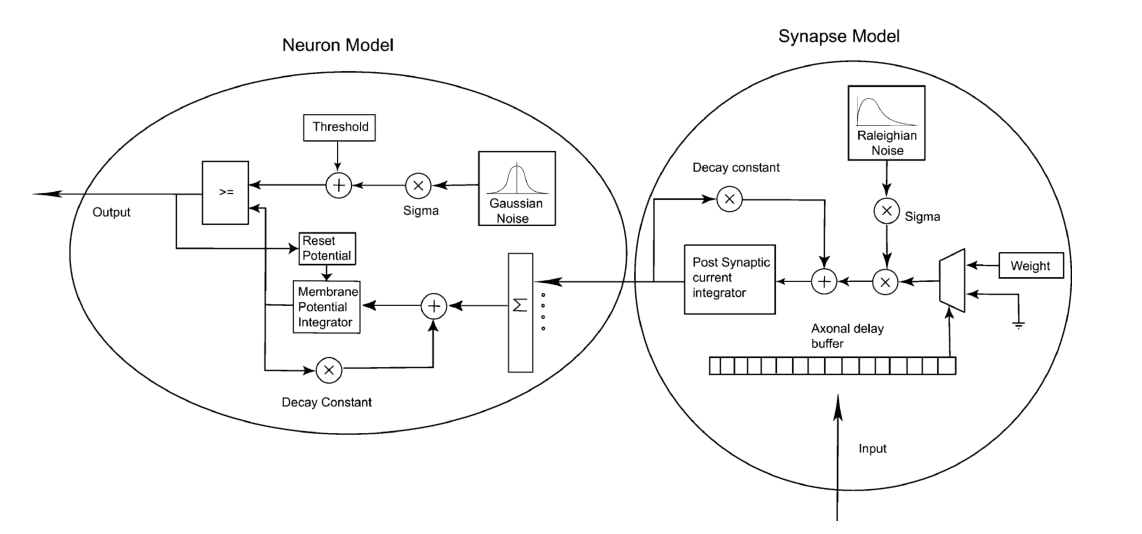
突触后电流的更新方程也包含类似的函数来最大化定点数系统的范围，如（7）（8）所示。

Δwij：突触噪声效能设置参数。

噪声来源：随机数产生







每个NPE内实现的硬件神经元和突触的框图

**[１０] An Efficient and Reconfigurable Synchronous Neuron Model**

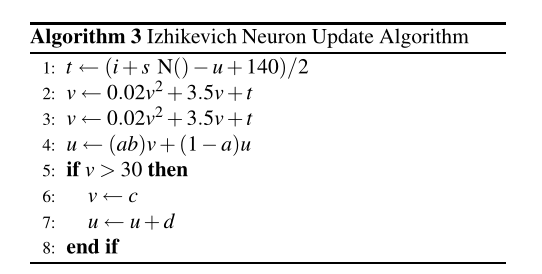
一种高效、可重构的同步神经元模型

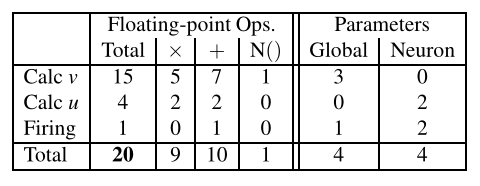
提出了一种可重构且高效的二维神经元模型，该模型能够扩展到更高的维数。

同步：该模型的同步特性使该方法适合于大规模流水线实现，称细胞模型为“同步”的原因是，在每个时钟周期中，动态变量在时间上的演变是相同的。

**[1１] FPGA Accelerated Simulation of Biologically Plausible Spiking Neural Networks**

**Izhikevich神经元模型：**能够在效率和生物学合理性之间达到最佳平衡的模型

****

****

Izhekivich神经元模型在浮点运算和常量运算中的代价。

该模型使用了两个神经元状态变量v和u，以及五个神经元参数a、b、c、d和s，可以显示出所有已知的脉冲模式，如间歇脉冲和混沌脉冲。基本的更新算法在每个时间步长上使用13个浮点运算，此处v分两步进行稳定性处理

函数N()提供了一个随机正态变量

**［１２］Loihi: A Neuromorphic Manycore Processor with On-Chip Learning**

Loihi:一个具有片上学习功能的神经形态多核处理器

（未涉及具体的神经元的设计）