



(12)发明专利

(10)授权公告号 CN 104094293 B

(45)授权公告日 2017.08.15

(21)申请号 201380008188.8

(22)申请日 2013.02.07

(65)同一申请的已公布的文献号  
申请公布号 CN 104094293 A

(43)申请公布日 2014.10.08

(30)优先权数据  
13/369,080 2012.02.08 US

(85)PCT国际申请进入国家阶段日  
2014.08.06

(86)PCT国际申请的申请数据  
PCT/US2013/025219 2013.02.07

(87)PCT国际申请的公布数据  
W02013/119867 EN 2013.08.15

(73)专利权人 高通股份有限公司  
地址 美国加利福尼亚州

(72)发明人 J·F·亨泽格 V·阿帕林

(74)专利代理机构 上海专利商标事务所有限公司 31100

代理人 元云

(51)Int.Cl.  
G06N 3/04(2006.01)  
G06N 3/08(2006.01)

(56)对比文件  
CN 102193518 A,2011.09.21,  
CN 102341815 A,2012.02.01,  
Runchun Wang等.《A programmable axonal  
propagation delay circuit for time-delay  
spiking neural networks》.《IEEE》.2011,第  
867-872页.

审查员 王德方

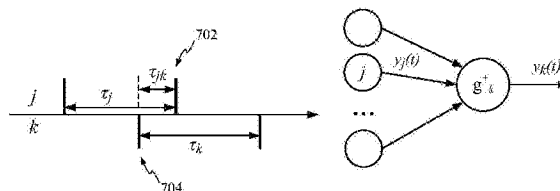
权利要求书2页 说明书34页 附图33页

(54)发明名称

用于尖峰神经计算的方法和装置

(57)摘要

本公开的某些方面提供用于一般线性系统的尖峰神经计算的方法和装置。一个示例方面是将信息编码在尖峰之间的相对定时中的神经元模型。然而,突触权重不是必需的。换句话说,连接可以存在(显著突触)或者不存在(不显著或不存在的突触)。本公开的某些方面使用二进制值的输入和输出并且不要求突触后滤波。然而,某些方面可涉及对连接延迟(例如,树突延迟)的建模。单个神经元模型可被用于计算任何一般线性变换 $x=AX+BU$ 到任何的任意性精度。这一神经元模型还可以能够学习,诸如学习输入延迟(例如,对应于缩放值)以达成目标输出延迟(或输出值)。学习还可被用于确定因果输入的逻辑关系。



1. 一种使用尖峰神经网络来学习的方法,包括:
  - (a) 根据与至神经元模型的输入相关联的当前延迟来延迟所述神经元模型中的输入尖峰,其中所述输入尖峰发生在相对于所述神经元模型的参考时间而言的输入尖峰时间处;
  - (b) 至少部分地基于经延迟的输入尖峰来从所述神经元模型发射输出尖峰;
  - (c) 确定从所述神经元模型发射所述输出尖峰与所述神经元模型的所述参考时间之间的实际时间差;以及
  - (d) 基于目标时间差与所述实际时间差之差、所述当前延迟、以及所述输入尖峰的输入尖峰时间来调节与所述输入相关联的所述当前延迟。
2. 如权利要求1所述的方法,其特征在于,进一步包括:  
在延迟所述输入尖峰之前初始化所述当前延迟。
3. 如权利要求1所述的方法,其特征在于,进一步包括:  
用经调节的延迟来重复(a)到(d),至少直至所述目标时间差与所述实际时间差之差小于或等于一阈值或者直至已经执行最大轮数的迭代。
4. 如权利要求1所述的方法,其特征在于,进一步包括:  
基于经调节的延迟来确定标量值,其中所述标量值是所述神经元模型的膜电位的变化系数的指数值的倒数,所述指数在取倒数之前被升至所述经调节的延迟次幂。
5. 如权利要求4所述的方法,其特征在于,所述标量值是线性变换的系数。
6. 如权利要求1所述的方法,其特征在于,所述目标时间差是涉及所述神经元模型的稳态过程的设定点。
7. 如权利要求1所述的方法,其特征在于,所述神经元模型具有指数性增长的膜电位,并在没有抑制性输入的情况下继续去极化,且其中兴奋性输入使所述神经元模型比所述神经元模型在没有所述兴奋性输入的情况下将进行的激发更早地激发。
8. 一种使用尖峰神经网络来学习的装置,包括:  
处理单元,其被配置成:
  - (a) 根据与至神经元模型的输入相关联的当前延迟来延迟所述神经元模型中的输入尖峰,其中所述输入尖峰发生在相对于所述神经元模型的参考时间而言的输入尖峰时间处;
  - (b) 至少部分地基于经延迟的输入尖峰来从所述神经元模型发射输出尖峰;
  - (c) 确定从所述神经元模型发射所述输出尖峰与所述神经元模型的所述参考时间之间的实际时间差;以及
  - (d) 基于目标时间差与所述实际时间差之差、所述当前延迟、以及所述输入尖峰的输入尖峰时间来调节与所述输入相关联的所述当前延迟。
9. 如权利要求8所述的装置,其特征在于,所述处理单元被进一步配置成:  
在延迟所述输入尖峰之前初始化所述当前延迟。
10. 如权利要求8所述的装置,其特征在于,所述处理单元被进一步配置成:  
用经调节的延迟来重复(a)到(d),至少直至所述目标时间差与所述实际时间差之差小于或等于一阈值或者直至已经执行最大轮数的迭代。
11. 如权利要求8所述的装置,其特征在于,所述处理单元被进一步配置成:  
基于经调节的延迟来确定标量值,其中所述标量值是所述神经元模型的膜电位的变化系数的指数值的倒数,所述指数在取倒数之前被升至所述经调节的延迟次幂。

12. 如权利要求11所述的装置,其特征在于,所述标量值是线性变换的系数。

13. 如权利要求8所述的装置,其特征在于,所述目标时间差是涉及所述神经元模型的稳态过程的设定点。

14. 如权利要求8所述的装置,其特征在于,所述神经元模型具有指数性增长的膜电位,并在没有抑制性输入的情况下继续去极化,且其中兴奋性输入使所述神经元比所述神经元模型在没有所述兴奋性输入情况下将进行的激发更早地激发。

15. 一种使用尖峰神经网络来学习的设备,包括:

(a) 用于根据与至神经元模型的输入相关联的当前延迟来延迟所述神经元模型中的输入尖峰的装置,其中所述输入尖峰发生在相对于所述神经元模型的参考时间而言的输入尖峰时间处;

(b) 用于至少部分地基于经延迟的输入尖峰来从所述神经元模型发射输出尖峰的装置;

(c) 用于确定从所述神经元模型发射所述输出尖峰与所述神经元模型的所述参考时间之间的实际时间差的装置;以及

(d) 用于基于目标时间差与所述实际时间差之差、所述当前延迟、以及所述输入尖峰的输入尖峰时间来调节与所述输入相关联的所述当前延迟的装置。

16. 如权利要求15所述的设备,其特征在于,进一步包括:

用于在延迟所述输入尖峰之前初始化所述当前延迟的装置。

17. 如权利要求15所述的设备,其特征在于,进一步包括:

用于用经调节的延迟来重复(a)到(d),至少直至所述目标时间差与所述实际时间差之差小于或等于一阈值或者直至已经执行最大轮数的迭代的装置。

18. 如权利要求15所述的设备,其特征在于,进一步包括:

用于基于经调节的延迟来确定标量值的装置,其中所述标量值是所述神经元模型的膜电位的变化系数的指数值的倒数,所述指数在取倒数之前被升至所述经调节的延迟次幂。

19. 如权利要求18所述的设备,其特征在于,所述标量值是线性变换的系数。

20. 如权利要求15所述的设备,其特征在于,所述目标时间差是涉及所述神经元模型的稳态过程的设定点。

21. 如权利要求15所述的设备,其特征在于,所述神经元模型具有指数性增长的膜电位,并在没有抑制性输入的情况下继续去极化,且其中兴奋性输入使所述神经元比所述神经元模型在没有所述兴奋性输入情况下将进行的激发更早地激发。

## 用于尖峰神经计算的方法和装置

[0001] 相关申请的交叉引用

[0002] 本申请涉及于2012年2月8日提交的且题为“METHODS AND APPARATUS FOR SPIKING NEURAL COMPUTATION (用于尖峰神经计算的方法和装置)”的美国专利申请S/N.13/368,994 (代理人案号113075U1)、以及于2012年2月8日提交的且题为“METHODS AND APPARATUS FOR SPIKING NEURAL COMPUTATION (用于尖峰神经计算的方法和装置)”的美国专利申请S/N.13/369,095 (代理人案号113075U3), 这两件申请均通过援引纳入于此。

[0003] 背景

[0004] 领域

[0005] 本公开的某些方面一般涉及神经网络, 并且更为具体地涉及操作由一个或多个神经元构成的尖峰神经网络, 其中单个神经元能够计算任何一般变换至任何的任意性精度。

[0006] 背景

[0007] 人工神经网络是由一群互连的人工神经元(即神经元模型)组成的数学或计算模型。人工神经网络可以源自于(或至少宽松地基于)生物神经网络(诸如在人脑中发现的那些生物神经网络)的结构和/或功能。由于人工神经网络能从观察中推断出函数, 因此这样的网络在因任务或数据的复杂度使得手工设计该函数不切实际的应用中是特别有用的。

[0008] 一种类型的人工神经网络是尖峰神经网络, 其将时间概念以及神经元和突触状态纳入到其工作模型中, 由此增加了此种类型的神经模拟中的真实性水平。尖峰神经网络基于神经元仅当膜电位达到阈值时才激发这一概念。当神经元激发时, 它生成一尖峰, 该尖峰行进至其他神经元, 这些其他神经元继而基于该收到的尖峰来升高或降低其膜电位。

[0009] 传统上, 信息被认为若非独占地也主要是被编码在神经元的激发率中。如果信息是被编码在神经元激发率中, 与仅将神经元建模为具有基于速率的学习规则(诸如Oja规则)的激发率变换相比, 要对具有膜动态、有时间精度的尖峰事件、以及尖峰定时依赖可塑性(STDP)的神经元进行建模会存在显著的计算开销。

[0010] 概述

[0011] 本公开的某些方面一般涉及尖峰神经计算, 并且更为具体地涉及使用尖峰神经网络中的一个或多个神经元, 其中单个神经元能够计算任何一般变换至任何的任意性精度, 并且其中信息被编码在各尖峰的相对定时中。

[0012] 本公开的某些方面提供一种用于实现尖峰神经网络的方法。该方法一般包括在第一神经元模型处接收至少一个输入; 基于该输入, 确定第一神经元模型的第一输出尖峰时间与参考时间之间的相对时间; 以及基于该相对时间从第一神经元模型发射输出尖峰。

[0013] 本公开的某些方面提供一种用于实现尖峰神经网络的装置。该装置一般包括: 处理单元, 被配置成在第一神经元模型处接收至少一个输入; 基于该输入, 确定第一神经元模型的第一输出尖峰时间与参考时间之间的相对时间; 以及基于该相对时间从第一神经元模型发射输出尖峰。

[0014] 本公开的某些方面提供一种用于实现尖峰神经网络的设备。该设备一般包括: 用于在第一神经元模型处接收至少一个输入的装置; 用于基于该输入, 确定第一神经元模型

的第一输出尖峰时间与参考时间之间的相对时间的装置;以及用于基于该相对时间从第一神经元模型发射输出尖峰的装置。

[0015] 本公开的某些方面提供了一种用于实现尖峰神经网络的计算机程序产品。该计算机程序产品一般包括具有指令的计算机可读介质,这些指令可执行用于:在第一神经元模型处接收至少一个输入;基于该输入,确定第一神经元模型的第一输出尖峰时间与参考时间之间的相对时间;以及基于该相对时间从第一神经元模型发射输出尖峰。

[0016] 本公开的某些方面提供一种使用尖峰神经网络的学习方法。该方法一般包括根据与至神经元模型的输入相关联的当前延迟来延迟该神经元模型中的输入尖峰,其中该输入尖峰发生在相对于该神经元模型的参考时间而言的输入尖峰时间处;至少部分地基于经延迟的输入尖峰来从该神经元模型发射输出尖峰;确定从该神经元模型发射输出尖峰与该神经元模型的参考时间之间的实际时间差;以及基于目标时间差与实际时间差之差、当前延迟以及输入尖峰的输入尖峰时间来调整与该输入相关联的当前延迟。

[0017] 本公开的某些方面提供一种用于使用尖峰神经网络来学习的装置。该装置一般包括处理单元,该处理单元被配置成:根据与至神经元模型的输入相关联的当前延迟来延迟该神经元模型中的输入尖峰,其中该输入尖峰发生在相对于该神经元模型的参考时间而言的输入尖峰时间处;至少部分地基于经延迟的输入来从该神经元模型发射输出尖峰;确定从神经元模型发射输出尖峰与该神经元模型的参考时间之间的实际时间差;以及基于目标时间差与实际时间差之差、当前延迟以及输入尖峰的输入尖峰时间来调整与该输入相关联的当前延迟。

[0018] 本公开的某些方面提供一种用于使用尖峰神经网络来学习的设备。该设备一般包括:用于根据与至神经元模型的输入相关联的当前延迟来延迟该神经元模型中的输入尖峰的装置,其中该输入尖峰发生在相对于该神经元模型的参考时间而言的输入尖峰时间处;用于至少部分地基于经延迟的输入来从该神经元模型发射输出尖峰的装置;用于确定从神经元模型发射输出尖峰与该神经元模型的参考时间之间的实际时间差的装置;以及用于基于目标时间差与实际时间差之差、当前延迟以及输入尖峰的输入尖峰时间来调整与该输入相关联的当前延迟的装置。

[0019] 本公开的某些方面提供了一种用于使用尖峰神经网络来学习的计算机程序产品。该计算机程序产品一般包括具有指令的计算机可读介质,这些指令可执行用于:根据与至神经元模型的输入相关联的当前延迟来延迟该神经元模型中的输入尖峰,其中该输入尖峰发生在相对于该神经元模型的参考时间而言的输入尖峰时间处;至少部分地基于经延迟的输入来从该神经元模型发射输出尖峰;确定从神经元模型发射输出尖峰与该神经元模型的参考时间之间的实际时间差;以及基于目标时间差与实际时间差之差、当前延迟以及输入尖峰的输入尖峰时间来调整与该输入相关联的当前延迟。

[0020] 本公开的某些方面提供一种使用尖峰神经网络的学习方法。该方法一般包括:在一个或多个学习神经元模型中的每一个学习神经元模型处提供一组逻辑输入,其中在该组逻辑输入上施加有真因果逻辑关系;在每组逻辑输入处接收输入尖峰之间的变化定时;以及对于这一个或多个学习神经元模型中的每一个学习神经元模型,使用接收到的输入尖峰来调整与每个逻辑输入相关联的延迟,从而该学习神经元模型根据对应于该真因果逻辑关系的一个或多个逻辑条件来发射符合目标输出延迟的输出尖峰。

[0021] 本公开的某些方面提供一种用于使用尖峰神经网络来学习的装置。该装置一般包括处理单元,该处理单元被配置成:在一个或多个学习神经元模型中的每一个学习神经元模型处提供一组逻辑输入,其中在该组逻辑输入上施加有真因果逻辑关系;在每组逻辑输入处接收输入尖峰之间的变化定时;以及对于这一个或多个学习神经元模型中的每一个学习神经元模型,使用接收到的输入尖峰来调整与每个逻辑输入相关联的延迟,从而该学习神经元模型根据对应于真因果逻辑关系的一个或多个逻辑条件来发射符合目标输出延迟的输出尖峰。

[0022] 本公开的某些方面提供一种用于使用尖峰神经网络来学习的设备。该设备一般包括:用于在一个或多个学习神经元模型中的每一个学习神经元模型处提供一组逻辑输入的装置,其中在该组逻辑输入上施加有真因果逻辑关系;用于在每组逻辑输入处接收输入尖峰之间的变化定时的装置;以及用于对于这一个或多个学习神经元模型中的每一个学习神经元模型,使用接收到的输入尖峰来调整与每个逻辑输入相关联的延迟,从而该学习神经元模型根据对应于该真因果逻辑关系的一个或多个逻辑条件来发射符合目标输出延迟的输出尖峰的装置。

[0023] 本公开的某些方面提供了一种用于使用尖峰神经网络来学习的计算机程序产品。该计算机程序产品一般包括具有指令的计算机可读介质,这些指令可执行用于:在一个或多个学习神经元模型中的每一个学习神经元模型处提供一组逻辑输入,其中在该组逻辑输入上施加有真因果逻辑关系;在每组逻辑输入处接收输入尖峰之间的变化定时;以及对于这一个或多个学习神经元模型中的每一个学习神经元模型,使用接收到的输入尖峰来调整与每个逻辑输入相关联的延迟,从而该学习神经元模型根据对应于该真因果逻辑关系的一个或多个逻辑条件来发射符合目标输出延迟的输出尖峰。

[0024] 本公开的某些方面提供一种用于一般线性系统的神经计算的系统。该系统一般包括具有膜电位的抗泄漏积分激发神经元,其中该膜电位在没有输入的情况下呈指数地增大,其中该膜电位在有输入之际步进增加,其中该膜电位在参考时间被复位到复位电位,并且其中如果膜超过阈值则该神经元发放尖峰;以及具有延迟但没有权重并且没有突触后滤波的一个或多个突触,其将输入连接至抗泄漏积分激发神经元。对于某些方面,抗泄漏积分激发神经元在膜电位超过阈值之际发放尖峰,并且参考时间是尖峰处或之后的时间,并且在复位之际,会受延迟影响的突触输入被清除。

[0025] 本公开的某些方面提供一种一般神经元建模的方法。该方法一般包括:在神经元的经延迟输入事件之际,将该输入应用于该神经元的状态并计算该神经元的预测将来尖峰时间;将该神经元的尖峰发放事件重新调度在该预测将来尖峰时间;在该神经元的尖峰事件之际,复位膜电位并计算该神经元的下一预测将来尖峰时间,其中膜电位是被复位至确保该神经元将在一历时内发放尖峰的值。

[0026] 本公开的某些方面提供一种使用尖峰神经元来计算线性系统的方法。该方法一般包括:基于输入值的对数的负值来确定相对于输入参考时间而言的输入时间尖峰时间;将输入延迟达与线性系数呈对数相关的时间延迟;以及基于抗泄漏积分激发神经元模型来计算相对于输出参考时间而言的输出尖峰时间。对于某些方面,该对数的底数等于作为膜电位函数的膜电位变化系数的指数值。根据某些方面,至突触后神经元的神经元输出藉由绝对值以及正系数抑制和负系数激发来表示负值,其中至突触后神经元的神经元输出藉由

绝对值以及负系数抑制和正系数激发来表示正值。对于某些方面,可负可正的输入值使用两个神经元来表示,一个表示正域以作为该值的经校正正值,而另一个表示负域以作为该值的经校正负值。

[0027] 本公开的某些方面提供一种在尖峰神经网络中转换定时信息的方法。该方法一般包括:将传播参照帧波作为输入施加到两群或更多群的一个或多个神经元,其中该参照帧波是振荡的兴奋性和/或抑制性电位,其在被施加到这两群或更多群中的每一群之前被延迟达一不同量;以及将信息编码在神经元的尖峰相对于被施加到该神经元(或该尖峰被输入到的神经元)的传播参照帧波而言的时间中和/或解码神经元的尖峰相对于被施加到该神经元(或该尖峰被输入到的神经元)的传播参照帧波而言的时间中的信息。

[0028] 本公开的某些方面提供了一种使用神经元将自指相对时间转换成非自指相对时间的装置。该装置一般包括:输入状态,其在有输入之际被设置并继该输入之后指数性衰减;输入锁存,其在后续输入处在输入状态被复位之前存储该输入状态;以及膜状态,其在有参考输入之际被递增达输入锁存值并在其后指数性增长直至超过阈值,在超过阈值之际该膜状态被复位并且其中膜状态在被复位之后不增长直至有参考输入。

[0029] 本公开的某些方面提供一种在尖峰神经网络中学习延迟的方法。该方法一般包括:将输入延迟达一当前输入延迟,其中输入是呈在相对于第一参考而言的时间处发生的尖峰的形式;将当前激发延迟确定为相对于第二参考而言的输出尖峰时间;计算目标激发延迟与当前激发延迟之差;以及将输入延迟调整达一取决于目标激发延迟与当前激发延迟之差、当前输入延迟、以及输入尖峰相对时间和学习速率的量。

[0030] 本公开的某些方面提供一种用于操作尖峰神经网络的方法。该方法一般包括:确定神经元模型处的输入尖峰的输入尖峰时间,该输入尖峰时间是相对于第一参考时间而言的;在存在多个输入尖峰的情况下,确定输出尖峰相对于第二参考时间而言的第一输出尖峰时间,该输出尖峰时间是基于相对于第一参考时间而言的输入尖峰时间;以及在有这多个输入尖峰的情况下,基于该神经元模型的去极化至尖峰延迟来确定该输出尖峰相对于第二参考时间的第二输出尖峰时间。

[0031] 本公开的某些方面提供一种用于操作尖峰神经网络的方法。该方法一般包括:在第一参考时间采样一值;将经采样值编码为延迟;以及通过在相对于第二参考时间而言的时间延迟生成输入尖峰而将该值输入至神经元模型。

[0032] 附图简述

[0033] 为了能详细理解本公开的以上陈述的特征所用的方式,可参照各方面来对以上简要概述的内容进行更具体的描述,其中一些方面在附图中解说。然而应该注意,附图仅解说了本公开的某些典型方面,故不应被认为限定其范围,因为本描述可允许有其他等同有效的方面。

[0034] 图1解说根据本公开的某些方面的示例神经网络。

[0035] 图2解说了根据本公开的某些方面的从实值域至尖峰定时域的变换。

[0036] 图3是解说根据本公开的某些方面在相对时间与经移位时帧之间的关系时序图。

[0037] 图4是解说根据本公开的某些方面的树突延迟的神经元模型的框图。

[0038] 图5解说了根据本公开的某些方面的神经元的指数性增长的膜电位和激发。

[0039] 图6是根据本公开的某些方面的单个抗泄漏积分激发(ALIF)神经元模型的架构的

框图。

[0040] 图7是解说根据本公开的某些方面的非自指突触后神经元(NSR-POST)参考时间与自指(SR)参考时间之差的时序图以及相关联的突触前和突触后神经元。

[0041] 图8解说了根据本公开的某些方面的导致神经元上的兴奋性和抑制性输入的正值和负值以及正和负缩放比例值的所有可能组合。

[0042] 图9解说了根据本公开的某些方面以正编码 $[-x_i(t)]_+$ 并连接为抑制性输入的神经元来表示负值。

[0043] 图10是根据本公开的某些方面的用于使用神经元模型来缩放标量值的示例操作的流程图。

[0044] 图11A解说了根据本公开的某些方面的当使用0.1ms的时间分辨率来对单个神经元模型的单个树突输入上的标量输入进行缩放时随时间推移的示例输入值、随时间推移的输出值、以及线性度。

[0045] 图11B解说了根据本公开的某些方面的当使用1.0ms的时间分辨率来对单个神经元模型的单个树突输入上的标量输入进行缩放时随时间推移的示例输入值、随时间推移的输出值、以及线性度。

[0046] 图12A解说了根据本公开的某些方面的当使用0.1ms的时间分辨率来对单个神经元模型的十个树突输入上的标量输入进行缩放时随时间推移的示例输入值、随时间推移的输出值、以及线性度。

[0047] 图12B解说了根据本公开的某些方面的当使用1.0ms的时间分辨率来对单个神经元模型的十个树突输入上的标量输入进行缩放时随时间推移的示例输入值、随时间推移的输出值、以及线性度。

[0048] 图13A解说了根据本公开的某些方面的当使用1.0ms的时间分辨率来对单个神经元模型的十个树突输入上的标量输入使用正和负缩放比例值进行缩放时随时间推移的示例输入值、随时间推移的输出值、以及线性度。

[0049] 图13B解说了根据本公开的某些方面的当使用1.0ms的时间分辨率、使用正和负缩放比例值两者但错误地忽略了要翻转到抑制性的情况下来对单个神经元模型的十个树突输入上的标量输入进行缩放时随时间推移的示例输入值、随时间推移的输出值、以及线性度,以供与图13A比较。

[0050] 图14解说了根据本公开的某些方面的当使用1.0ms的时间分辨率、使用正和负缩放比例值两者、并且向神经元模型的膜电位添加了噪声项的情况下对单个神经元模型的十个树突输入上的标量输入进行缩放时随时间推移的示例输入值、随时间推移的输出值、以及线性度。

[0051] 图15解说了根据本公开的某些方面的向两个神经元提供相同参考。

[0052] 图16解说了根据本公开的某些方面的将参考用于两个神经元的前馈情形。

[0053] 图17解说了根据本公开的某些方面的将参考用于两个神经元的反馈情形。

[0054] 图18解说了根据本公开的某些方面的用于一系列神经元的传播参考波。

[0055] 图19解说根据本公开的某些方面的具有图18的传播参考波的一系列神经元的示例时序图。

[0056] 图20解说了根据本公开的某些方面来使用示例g-神经元。

[0057] 图21解说了根据本公开的某些方面的至神经元模型的1、2、4、8、16和32个输入的线性度图表。

[0058] 图22是根据本公开的某些方面的用于基于相对时间来从神经元模型发射输出尖峰的示例操作的流程图。

[0059] 图22A解说了能够执行图22中示出的各操作的示例装置。

[0060] 图23解说了根据本公开的某些方面的很有可能影响突触后神经元的激发的输入尖峰、以及不会影响突触后神经元的激发的另一输入尖峰。

[0061] 图24解说了根据本公开的某些方面的五个代表性突触前神经元和突触后神经元。

[0062] 图25A和25B解说了根据本公开的某些方面的学习有噪二进制输入矢量的系数的示例结果。

[0063] 图26根据本公开的某些方面在延迟和权重的图表中针对每个输入解说了学习有噪二进制输入矢量的系数的示例结果。

[0064] 图27解说了根据本公开的某些方面的学习有噪实值输入矢量的系数的示例结果。

[0065] 图28A是根据本公开的某些方面的在逻辑OR(或)关系的第一轮迭代后的延迟的图表。

[0066] 图28B是根据本公开的某些方面的在逻辑AND(与)关系的第一轮迭代后的延迟的图表。

[0067] 图29A是根据本公开的某些方面的在逻辑OR关系的数轮迭代后的延迟的图表。

[0068] 图29B是根据本公开的某些方面的在逻辑AND关系的数轮迭代后的延迟的图表。

[0069] 图30是根据本公开的某些方面的用于学习逻辑关系的收敛(作为迭代轮数的函数)。

[0070] 图31解说了根据本公开的某些方面为了在尖峰神经网络中进行学习而实现否定和整合演绎两者。

[0071] 图32是根据本公开的某些方面的用于在尖峰神经网络中进行学习的示例操作的流程图。

[0072] 图32A解说了能够执行图32中示出的各操作的示例装置。

[0073] 图33是根据本公开的某些方面的用于在尖峰神经网络中进行因果性学习的示例操作的流程图。

[0074] 图33A解说了能够执行图33中示出的各操作的示例装置。

[0075] 详细描述

[0076] 以下参照附图更全面地描述本公开的各个方面。然而,本公开可用许多不同形式来实施并且不应解释为被限于本公开通篇给出的任何具体结构或功能。确切而言,提供这些方面是为了使得本公开将是透彻和完整的,并且其将向本领域技术人员完全传达本公开的范围。基于本文中的教导,本领域技术人员应领会,本公开的范围旨在覆盖本文中所披露的本公开的任何方面,不论其是与本公开的任何其他方面相独立地还是组合地实现的。例如,可以使用本文所阐述的任何数目的方面来实现装置或实践方法。另外,本公开的范围旨在覆盖使用作为本文中所阐述的本公开的各种方面的补充或者与之不同的其他结构、功能性、或者结构及功能性来实践的此类装置或方法。应当理解,本文中所披露的本公开的任何方面可由权利要求的一个或多个元素来实施。

[0077] 措辞“示例性”在本文中用于表示“用作示例、实例或解说”。本文中描述为“示例性”的任何方面不必被解释为优于或胜过其他方面。

[0078] 尽管本文描述了特定方面,但这些方面的众多变体和置换落在本公开的范围之内。虽然提到了优选方面的一些益处和优点,但本公开的范围并非旨在被限定于特定益处、用途或目标。相反,本公开的各方面旨在能宽泛地应用于不同的技术、系统配置、网络和协议,其中一些作为示例在附图以及以下对优选方面的描述中解说。该详细描述和附图仅仅解说本公开而非限定本公开,本公开的范围由所附权利要求及其等效技术方案来定义。

[0079] 示例神经系统

[0080] 图1解说根据本公开的某些方面的具有多级神经元的示例神经系统100。神经系统100可包括神经元级102,该级神经元102通过突触连接网络104连接到另一级神经元106。为简单起见,图1中仅解说了两级神经元,但在典型的神经系统中可存在更少或更多级神经元。

[0081] 如图1所解说的,级102中的每一神经元可接收输入信号108,输入信号108可以由前级(图1中未示出)的多个神经元所生成的。信号108可表示级102的神经元的输入(例如,输入电流)。此类输入可在神经元膜上累积以对膜电位进行充电。当膜电位达到其阈值时,该神经元可激发并生成输出尖峰,该输出尖峰将被传递到下一级神经元(例如,级106)。

[0082] 尖峰从一级神经元向另一级神经元的传递可通过突触连接(或简称“突触”)网络104来达成,如图1所解说的。突触104可从级102的神经元(相对于突触104而言的突触前神经元)接收输出信号(即尖峰)。对于某些方面,这些信号可根据可调整的突触权重  $w_1^{(i,i+1)}$  ...  $w_P^{(i,i+1)}$  (其中P是级102和106的神经元之间的突触连接的总数)来缩放。对于其它方面,突触104可以不应用任何突触权重。此外,(经缩放)信号可被组合以作为级106中每个神经元(相对于突触104而言的突触后神经元)的输入信号。级106中的每个神经元可基于对应的组合输入信号来生成输出尖峰110。随后可使用另一突触连接网络(图1中未示出)将这些输出尖峰110传递到另一级神经元。

[0083] 神经系统100可以软件或以硬件(例如通过电路)来仿真并且可以用在大范围的应用中,诸如图像和模式识别、机器学习、电机控制等等。神经系统100中的每个神经元(或神经元模型)都可以被实现为神经元电路。被充电至发起输出尖峰的阈值的神经元膜可被实现为例如对流经其的电流进行积分的电容器。

[0084] 示例尖峰神经计算

[0085] 已经提出了用于设计神经网络的各种方法。例如,最近由Eliasmith和Anderson提出了一种使用神经工程设计框架(NEF)来对神经模拟进行工程设计或合成的方法。参见例如,Chris Eliasmith和Charles H.Anderson所著的Neural Engineering(神经工程设计): Computation,Representation,and Dynamics in Neurobiological Systems(神经生物学系统中的计算、表示和动态),MIT出版社(2003),<http://compneuro.uwaterloo.ca>,以及Chris Eliasmith所著的A Unified Approach to Building and Controlling Spiking Attractor Networks(构建和控制尖峰吸引子网络的统一办法),Neural Computation(神经计算)第17期,1276-1314(2005)。这一方法依赖于将经编码值表示在神经元集(由i索引)的活动  $\{a_i\}$  中。值可由这些活动的线性函数来估计,

$$[0086] \quad \hat{x} = \sum_i a_i \phi_i$$

[0087] 为了表示该值,神经元集群的活动必须足够多样,从而存在能从这些活动中估计该值的集合  $\{\phi_i\}$ 。例如,如果所有神经元具有相同动态,则这些活动可能不够多样到可获得该值的准确或精确表示。

[0088] 神经元激发率可被用作神经元活动度量。然而,曾提议可通过将活动表达为经滤波尖峰串来纳入时间代码:

$$[0089] \quad a_i(t) = h_i(t) * y_i(t)$$

[0090] 其中  $y_i(t)$  是二进制值(1表示尖峰而0表示无尖峰)。但为了保留显著信息,这依赖于使每突触有突触后滤波器  $h_i(t)$  并且  $h_i(t)$  具有显著时间常数。Eliasmith和Anderson实际上假定了突触后滤波器的动态支配着神经元响应的动态并且用时间常数  $\tau$  来将该滤波器建模为

$$[0091] \quad h_i(t) = \frac{1}{\tau} e^{-t/\tau}$$

[0092] 有效地,此举将布尔时间序列  $y_i(t)$  转换为类似于激发率的实值时间序列  $a_i(t)$ 。遗憾的是,此举导致信息损失(或从另一角度来看,所表示的值的分辨率受限)。这可通过增加神经元集群大小来得到补偿。最后,该方法使用突触权重来线性地变换输入以达成期望函数(假定它不是恒等)。

[0093] NEF的全神经描述包括:(1)以上针对每个输入神经元  $i$  到每个神经元  $j$  描述的形式突触后滤波器  $h_{ij}(t)$ , (2) 每个输入神经元  $i$  到每个神经元  $j$  的用于解码输入的突触权重  $w_{ij}$ , (3) 用于对输入进行重新编码的突触权重,以及(4)胞体动态。

[0094] 然而,这一办法有数个问题。首先,它是尖峰神经元形式的低效解决方案,因为该方法仅可使用激发率来应用。换句话说,如果可以使用基于速率的神经网络,则使用尖峰神经网络的动机何在?该方法仅是将布尔序列转换为实值序列。第二,该方法是计算上复杂的,其要求每突触有突触后滤波器以及对变换和权重进行编码和解码。第三,必须使用神经元的多样集群来表示系统,因为必须有足够的基础可供用于编码/解码以表示值空间。这意味着系统要求具有不同调谐曲线(激发率是输入的函数)的神经元。第四,学习是未解决的问题。

[0095] 类似地,其它现有技术方法已经将信息表示在神经元的激发率或者在神经元集群的激发活动中。这种集群编码类型方案通常与神经元的概率性激发模型(诸如泊松激发神经元)联用。这些类型的方案要求尖峰行为的观测窗以确定集群编码的平均激发率,该平均激发率表示该信息。概率性方法也可能是固有有噪的,甚至在引入外部噪声之前便是固有有噪的并且因此处于不利。因为这种使用激发率作为代码、使用精确权重以及复杂的计算性方案的做法,这些方案没有一种能解决这些问题。

[0096] 因此,需要的是用于实现极为高效、生物学上一致的普遍适用的尖峰神经网络的技术和装置。因此,以上说明中提出了三个关键问题区:效率、生物学一致性、以及普遍性或通性。

[0097] 如本文所使用的,术语“效率”一般指不需要多个神经元或神经元集群来计算基础标量函数并且不需要花显著时间来观察输出、转换和使输出达到平均以获得激发率值的神经元计算系统。神经元还应是计算上不复杂的、非概率性的以及不要求滤波器计算或者精确浮点计算的。

[0098] 如本文所使用的,术语“生物学上一致”一般意指神经元动态和连接处理应该是有生物学动机的(现实的)并且不仅由工程设计或计算上的便利性来指定。此外,信息在理想情况下应被编码在神经元的尖峰发放中而非其平均激发率中。神经元参数、和参数范围(诸如突触强度分布)应该同样是生物学上一致的。

[0099] 如本文所使用的,术语“通用的”一般指等效于一般计算或控制系统的神经计算系统。此类系统可被建模为线性系统或线性子系统的组合。任何线性系统由以下方程来描述

$$[0100] \quad \dot{\mathbf{x}}(t) = \mathbf{W}\mathbf{x}(t) \text{ 或即 } \mathbf{x}(t + \Delta t) = \mathbf{W}\mathbf{x}(t)$$

[0101] 对线性系统的这一公式化是普适的,因为矢量 $\mathbf{x}$ 可包括输入、内部状态变量、或者输出(第一类型的典型公式化通过表达 $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$ 来将输入与状态分开)。任何线性控制系统可在这些项中被表达(尽管矩阵 $\mathbf{W}$ 的条目中的一些条目取决于系统可以是0,从而“输入”可被外部事件而非被该方程本身更新)。因此根本上所关心的是元素

$$[0102] \quad \dot{x}_k(t) = w_k \mathbf{x}(t) \text{ 或即 } x_k(t + \Delta t) = w_k \mathbf{x}(t)$$

[0103] 其表达矢量 $\mathbf{x}$ 的每个元素的动态。尽管本公开的各方面并不限于线性系统,但这是要在其中描述某些方面的非常一般化的框架,因为线性度能被期望的非线性函数所取代,或者线性度可被用于用线性系统模型来建模非线性函数。

[0104] 本公开的各方面包括一种用于尖峰神经网络的方法。然而,权重不是必需的。换句话说,连接可以存在(显著的突触)或者不存在(不显著或不存在的突触)。本公开的某些方面使用二进制值的输入和输出并且不要求突触后滤波。然而,本公开的某些方面可涉及对连接延迟的建模。

[0105] 根据本公开的某些方面,使用本文描述的方法,任何线性系统可使用尖峰神经元来高效计算,而无需突触权重或突触后滤波器,并且信息被编码在每个个体的尖峰中。信息可被编码在每个尖峰的相对定时中,从而无需累积活动以确定速率中的信息。神经元尖峰响应时间可被直接计算(确定性的)、在分立或连续时间中计算、以基于事件的方式计算,由此进一步节省计算和存储器访问。因为单个神经元能表示任意性精度的线性变换,所以本公开的某些方面可将最小数量的神经元用于任何变量集。此外,某些方面不需要具有不同调谐曲线的神经元来保留保真度。本公开的某些方面也不要求概率性激发或集群编码。某些方面还可与对经时间编码的模式进行分类的神经元对接。换句话说,本公开的某些方面可被用于计算输入的变换,该变换可随后被用于分类或识别输入的各方面。相反,神经元对时间模式的响应可被提供给根据某些方面操作的神经元集。本公开的某些方面展现了生物学上一致的行为。例如,某些方面不使用突触权重。在神经元之间或有连接,或没有连接。在生物学上,树突脊倾向于抑或生长成具有标称强度的突触(取决于位置)抑或消失。作为另一示例,某些方面使用传播参考帧波来将在时间上编码的信息从自指形式转换为非自指形式。在生物学上,已经观察到兴奋与抑制之间的局部振荡,包括在不同皮层区域之间的偏移。作为又一示例,某些方面使用处于或靠近去极化阈值的神经元模型,神经元的膜电位在该去极化阈值之上会增加,即使没有进一步的输入亦是如此。生物学神经电路可同样在这一去极化阈值处或其附近操作。

[0106] 本公开的某些方面还可与概率性或确定性激发模型联用。并且,在这些模型类型的任一者之上可添加(容忍)噪声。因为此类模型可在根本的确定性模式(无噪声)中操作,所以这些模型可执行快速且可靠的计算并从容地处置噪声的添加。

[0107] 本公开的某些方面还可减少功耗,因为它们包括基于事件的或经调度的神经元方法,其中计算成本可独立于时间分辨率。此举还可解释生物脑如何/为何利用稀疏编码来实现高效功耗。

[0108] 本公开的某些方面的另一优势在于这些方面可按异步或同步模式或形式来操作。在异步形式中,采样率可与输入的显著性成比例。当输入是可忽略的或者为低时,采样率可为低,由此节省功率和资源。当输入是显著的或为大时,采样率可以增加,由此增加了精度和响应时间。

[0109] 在以下描述中,这些以及附加优势将变得明显。注意,本公开的各方面的许多优势可表达在不具有其他常规方法和神经元模型的特定不利之处的方面上。

[0110] 将信息编码在相对时间中

[0111] 设想信息可被编码在神经元的各尖峰之间的相对时间中或者一个神经元和另一个神经元的尖峰之间(或者甚至更一般化地,如将在稍后变得更为明了的,一个神经元的尖峰与某个参考时间之间)的相对时间中。在某些方面,信息可以以下两种基本形式来表示在尖峰之间的相对定时中:(1)相同神经元的连贯尖峰之间的相对时间;以及(2)一个神经元的尖峰与另一神经元的尖峰(或参考)之间的相对时间。

[0112] 严格地说,在第一种情形中,相对时间不需要是连贯尖峰之间的相对时间,并且在第二种情形中,相对时间不需要是突触前神经元与突触后神经元之间的相对时间。无论如何,令此相对时间为 $\tau$ ,从而其与值 $x$ 如下相关:

[0113]  $x = q^{-\tau}$ 或即 $\tau = -\log_q x$

[0114] 在实效上,该相对时间是该值的以 $q$ 为底的对数。以上变换将相对时间(尖峰定时)域转换至实值(变量)域/将实值(变量)域转换至相对时间(尖峰定时)域。因此,对该值的任何缩放仅是相对时间上的时间差。

[0115]  $x_k = w_{jk} x_j \rightarrow \tau_k = \tau_j + \Delta \tau_{jk}$

[0116] 其中 $\Delta \tau_{jk} = -\log_q w_{jk}$ 。因此,以各尖峰之间的延迟来表示的值可仅通过添加一延迟来被缩放达任意性量。

[0117] 图2解说了根据本公开的某些方面的从实值域200至尖峰定时域210的变换。在实值域200中,神经元j202的输出值 $x_j$ 被缩放 $w$ 以获得神经元k204的输出值 $x_k$ 。在尖峰定时域210中,延迟 $z$ 被添加至神经元j202的相对时间 $\tau_j$ 以确定神经元k204的相对时间 $\tau_k$ 。

[0118] 注意,在因果性系统中,时间延迟是正的。然而,将一值缩放小于一的因子可通过将各尖峰之间的延迟缩短达一任意大的标称量并随后添加回足以达成总体缩放 $\Delta \tau_{jk}' = -T + \Delta \tau_{jk}$ 的量来达成,或即

[0119]  $\tau_k = \tau_j - T + \Delta \tau_{jk}'$

[0120] 如何用神经元来完成此举将在以下进一步详细说明。目前,可假定如果可达成等于或者大于一的任何缩放,则任何的任意性缩放均可使用这一概念来达成。在实际的网络中,相对时间延迟可被限定于某个最大值(在没有输入的情况下由标称神经元激发率给出,或者由从一个神经元到另一神经元的连接可能招致的最大延迟给出)。这不会限制可达成的缩放,因为期望值范围可通过缩放根本值来得到覆盖。

[0121] 相对定时的各形式之间的转换可包括:(1)将同一神经元 $i$ 的连贯尖峰之间的相对时间 $\tau_i$ 转换为神经元 $j$ 和神经元 $k$ (或参考 $k$ )的尖峰之间的相对时间 $\tau_{jk}$ ;以及(2)将神经元 $j$

和神经元k(或参考k)的尖峰之间的相对时间 $\tau_{jk}$ 转换为同一神经元i的连贯尖峰之间的相对时间 $\tau_i$ 。

[0122] 自指相对时间 $\tau_i$ 和非自指相对时间 $\tau_{jk}$ 之差可通过考量尖峰串来知晓。令神经元j的尖峰输出由 $y_j(t)$ 来描述,

$$[0123] \quad y_j(t) = \sum_s \delta(t - t_j^s)$$

[0124] 其中s是尖峰的索引(第一、第二、第三等)。因此, $y_j(t)$ 仅是二进制序列。神经元j的连贯尖峰之间的相对时间为

$$[0125] \quad \tau_j(s) = t_j^s - t_j^{s-1}$$

[0126] 然而,神经元j的尖峰s和神经元k(或参考k)的尖峰r之间的相对时间是

$$[0127] \quad \tau_{jr}(s,r) = t_j^s - t_k^r$$

[0128] 现在,设想来自神经元j的输出 $y_j(t)$ 被提交作为至神经元k的输入。如果是处于在处理了尖峰s之前的所有输入尖峰之后的时间t,则仅尖峰s是要考虑的,因此

$$[0129] \quad y_j(t) = \delta(t - t_j^s)$$

[0130] 现在,该时间还可被移位至任意性参考时间 $t_r$ ,其中 $t = t_r + t'$ 。现在,在经移位的时间,

$$[0131] \quad y_{jr}(t') = y_j(t_r + t') = \delta(t_r + t' - t_j^s) = \delta(t' - \tau_{jr}^s)$$

[0132] 现在,将相同时间移位应用于尖峰, $\tau_{jr}^s = t_j^s - t_r$ ,并使用速记 $\tau_{jr} = \tau_{jr}^s$ ,则得到

$$[0133] \quad y_{jr}(t') = \delta(t' - \tau_{jr})$$

[0134] 图3是解说如上所述地在相对时间与经移位时帧之间的关系时序图300。从时序图300和以上方程中可以看出,从一个神经元向另一神经元的任何尖峰串输入可在下一尖峰相对于参考时间的定时并计及连接中的延迟的意义上考虑。相对时间和时间参考移位的这些概念将是以下对本公开的某些方面的解释的中心所在。

### [0135] 一般尖峰神经系统

[0136] 已经提出了许多生物学上一致的神经元模型。生物学上尤其一致并且灵活的一种尖峰神经元模型是由以下差分方程描述的Izhikevich简单模型:

$$[0137] \quad \frac{dv}{dt} = c_1(v - v_r)(v - v_t) - c_2u + c_3I; \quad \frac{du}{dt} = c_4(c_5(v - v_r) - u)$$

[0138] 以及尖峰发放条件,

$$[0139] \quad \text{如果 } v(t) > v_\theta \rightarrow y(t) = 1, \text{ 则 } v(t+\epsilon) = v_r, u(t+\epsilon) = u(t) - c_6$$

[0140] 其中 $c_i$ 是神经元参数(从传导性等导出)并且 $v_r$ 、 $v_t$ 和 $v_\theta$ 是膜电压阈值,I是时间t处的净输入,并且其中一般而言 $v_\theta > v_t > v_r$ 且一般而言 $v > v_r$ ,除了在不应该时期以外。如果不满足尖峰发放条件,则输出 $y(t) = 0$ 。然而,该电压方程在两个域中操作:(1)  $v < v_t$ ,其中神经元像带泄漏积分激发(LIF)神经元那样操作;以及(2)  $v > v_t$ ,其中神经元像抗泄漏积分激发(ALIF)神经元那样操作。

[0141] 注意,当 $v < v_t$ 时,差 $v - v_t$ 的幅值越大,差 $v - v_r$ 的幅值就越小,从而这两者可被认为彼此相平衡。此外,不失一般性,可设置 $v_r = 0$ 并根据 $c_3$ 、 $c_4$ 和 $c_5$ 来缩放这些参数,从而可针对 $v < v_t$ 进行简化并写为,

$$[0142] \quad \frac{dv}{dt} = \rho v - \gamma u + I$$

[0143] 其中 $\rho < 0$ 。这里,不失一般性,电压域已经被移位达量 $v_t$ 。与之形成对比的是,当 $v > v_t$ 时,项 $(v - v_t)$ 以大于 $(v - v_r)$ 的斜率增长,并且因此可以有动机针对 $v > v_t$ 进行简化并写为,

$$[0144] \quad \frac{dv}{dt} = \rho v - \gamma u + I$$

[0145] 其中 $\rho > 0$ 。关键在于,符号 $\rho$ 被翻转,从而电压增长,并且增长得越来越快。

[0146] 一般来说,给定了任何背景输入活动、输入噪声电平、或者在前尖峰,则任何给定尖峰可将平衡从 $v < v_t$ 扭转至 $v > v_t$ 。以上推广反映了神经元电平可被维持在这一平衡的余量内,因此可推广并初始设置 $v_t = 0$ (在尖峰发放之后或者在继尖峰之后的不应期之后)。然而,由于这些推广,要调和这两个不同域中的电压最有可能需要补偿计算来在它们之间进行转换(即,当 $v$ 在 $\rho < 0$ 域中到达阈值 $v_\theta$ 时, $v$ 被转换为 $\rho > 0$ 域中的 $v = v - v_\theta \cong 0$ ,反之亦然)。尽管这样做是直截了当的,但是这对于本公开的各方面而言是不必要的,因为可以无需在这些域之间移动,如稍后将变得明了的。还可以观察到,对于 $c_2$ 、 $c_4$ 和 $c_5$ 的特定参数设置,状态变量 $u$ 可对这些动态具有可忽略的贡献,因此可一般化地简化为

$$[0147] \quad \frac{dv}{dt} = \rho v + I$$

[0148] 在实效上,可以看出,Izhikevich简单模型可取决于操作域被简化为(即,建模为)带泄漏或抗泄漏积分激发(ALIF)神经元。此外,如果没有假定有突触后滤波且没有假定有权重,则输入尖峰可被用作至经受某些连接延迟(诸如树突或轴突延迟)作用的神经元 $k$ 的输入,从而

$$[0149] \quad I_k = \sum_i y_i(t - \Delta\tau_{ik})$$

[0150] 图4是根据本公开的某些方面的将连接延迟描绘为树突延迟线的神经元模型410的框图400。关于输入 $y_i$ 的树突输入线被耦合至表示每个输入 $y_i$ 的连接延迟的树突延迟元件412。经延迟的输入随后由加法器414加总(线性地加总)以获得 $I_k$ 。

[0151] 现在,假设上标+表示 $\rho > 0$ 域,而上标-表示 $\rho < 0$ 域。对于神经元 $k$ 的抗泄漏域,

$$[0152] \quad \frac{dv_k^+(t)}{dt} = \rho^+ v_k^+(t) + \sum_l y_{c^-(k,l)}(t - \Delta\tau_{c^-(k,l)k})$$

[0153] 其中 $c^-(k,l)$ 是对应于突触后神经元 $k$ 的突触 $l$ 的突触前神经元的索引。此种神经元将一般具有指数性增长的膜电位,此指数性增长始于该电位被陡增至零以上(即,在零以上达某个量或者某个阈值)的参考时间 $t_r$ 。

[0154] 例如,图5解说了神经元的关于时间的此种指数性增长的膜电位和激发。在图5中, $y$ 轴504表示神经元的膜电位,而 $x$ 轴502表示时间。如图所示,当在参考时间 $t_r$ 该神经元的膜电位在零以上达某个量时,该电位指数性地增长。在该点之后,该神经元一般将会激发,除非有足以将电位带回到零或以下的抑制性输入。通常,兴奋性输入将仅仅导致神经元更早地激发。例如,如图5中所示,时间 $t_j$ 处来自神经元 $j$ 的输入导致该神经元更早地激发。这是一般公式化。例如,可将参考时间设为神经元激发的时间。随后,一旦激发,该神经元仅使该

电压立即复位至零以上的小值。

[0155] 可使用突触前神经元的索引来等效地写为

$$[0156] \quad \frac{dv_k^+(t)}{dt} = \rho^+ v_k^+(t) + \sum_j y_j(t - \Delta\tau_{jk})$$

[0157] 取拉普拉斯变换,可获得

$$[0158] \quad V_k^+(s) = \frac{v_k^+(t_k^-) + \sum_j e^{-(\tau_{jk} + \Delta\tau_{jk})s}}{s - \rho^+}$$

[0159] 其中根据推广,  $v_k^+(t_k^-) \neq 0$ 。目前,可使用实际上是神经元k的最后尖峰的参考时间,因此  $y_j(t) = \delta(t - t_k^- - \tau_{jk})$ 。在经移位时间  $\tau_k = t - t_k^-$  取拉普拉斯逆变换,得到

$$[0160] \quad v_k^+(\tau_k) = v_k^+(t_k^-) e^{\rho^+ \tau_k} + \sum_j e^{\rho^+ (\tau_k - (\tau_{jk} + \Delta\tau_{jk}))} = e^{\rho^+ \tau_k} \left[ v_k^+(t_k^-) + \sum_j e^{-\rho^+ (\tau_{jk} + \Delta\tau_{jk})} \right]$$

[0161] 针对  $\tau_k$  来求解,

$$[0162] \quad \tau_k = \frac{1}{\rho^+} \log v_k^+(\tau_k) - \frac{1}{\rho^+} \log \left[ v_k^+(t_k^-) + \sum_j e^{-\rho^+ (\tau_{jk} + \Delta\tau_{jk})} \right]$$

[0163] 进行操纵得到,

$$[0164] \quad v_k^+(\tau_k) e^{-\rho^+ \tau_k} = v_k^+(t_k^-) + \sum_j e^{-\rho^+ (\tau_{jk} + \Delta\tau_{jk})}$$

[0165] 定义  $q = e^{\rho^+}$ ,

$$[0166] \quad \left(\frac{1}{q}\right)^{\tau_k} = \Delta\omega_k + \sum_j \omega_{jk} \left(\frac{1}{q}\right)^{\tau_{jk}}$$

[0167] 其中  $\omega_{jk} = e^{-\rho^+ \Delta\tau_{jk}} / v_k(\tau_k)$  并且  $\Delta\omega_k = v_k(t_k^-) / v_k(\tau_k)$ 。

[0168] 在具有时间分辨率  $\Delta t$  的离散时间形式中,

$$[0169] \quad v_k^+(t) = (1 + \Delta t \rho) v_k^+(t - \Delta t) + \sum_j y_j(t - \Delta t - \Delta\tau_{jk})$$

$$[0170] \quad \text{定义 } q = (1 + \Delta t \rho)^{1/\Delta t}, v_k^+(t) = q^{\Delta t} v_k^+(t - \Delta t) + \sum_j y_j(t - \Delta t - \Delta\tau_{jk})$$

[0171] 在经移位时间  $\tau_k = t - t_k^-$  中的非递归形式中,

$$[0172] \quad v_k^+(\tau_k) q^{-\tau_k} = v_k^+(t_k^-) + \sum_j q^{-(\tau_{jk} + \Delta\tau_{jk})} u(\tau_k - (\tau_{jk} + \Delta\tau_{jk}))$$

[0173] 设想尖峰发生在自指时间  $\tau_k$  处,并假设所有被考虑的输入(即,不失一般性)发生在尖峰发放时间之前,

$$[0174] \quad \left(\frac{1}{q}\right)^{\tau_k} = \Delta\omega_k + \sum_j \omega_{jk} \left(\frac{1}{q}\right)^{\tau_{jk}}$$

[0175] 其中  $\omega_{jk} = q^{-\Delta\tau_{jk}} / v_k^+(\tau_k)$  且  $\Delta\omega_k = v_k^+(t_k^-) / v_k^+(\tau_k)$ 。

[0176] 如上所述,期望的线性系统可被如下表达:

$$[0177] \quad x_k(t) = \sum_j \omega_{jk} x_j(t) + \eta_k$$

[0178] 且从值到时间的变换可被表达为 $x = q^{-1}$ 或 $\tau = -\log_q x$ 。因此,可利用单个神经元k以力图通过以下转换来计算期望的线性结果:

$$[0179] \quad \tau_{jk} = -\log_q x_j$$

$$[0180] \quad \tau_k = -\log_q x_k$$

[0181] 为了使 $\eta_k$ (其由 $\eta_k = \Delta \omega_k$ 来标称建模)最小化,通常期望 $\Delta \omega_k \rightarrow 0$ 或者神经元尖峰阈值 $v_k^+(\tau_k) \gg v_k^-(\tau_k)$ 。使 $\eta_k$ 最小化不是严格必须的,但它提升了精确映射的动机。

[0182] 为了实现期望操作,可简单地设置延迟,

$$[0183] \quad \Delta \tau_{jk} = -\log_q (\omega_{jk} v_k^+(\tau_k))$$

[0184] 不失一般性,可设置 $v_k^+(\tau_k^-) = v_0 = 1$ ,从而 $\Delta \omega_k = 1/v_k^+(\tau_k) = 1/v_0$ ,并且因此可通过选取足够大的尖峰发放阈值来控制 $\eta_k$ 项的最低限度所需大小。

[0185] 还注意到,如果给定了无限时间分辨率(即, $\Delta t \rightarrow 0$ ) (在该情形中 $q = e^{\rho^+}$ ),则该单个神经元k可用无限分辨率来处理实值操作。然而,在实践中,这可能不是必需的。注意,这不一定是期望的,因为在连续时间中操作不要求连续更新。极有可能仅在在有事件(输入或尖峰,并且这些事件可在给定了以上提供的非嵌套表达的情况下被计算)之际更新方程。尽管如此,对于以时间步长 $\Delta t > 0$ 来操作的离散时间系统,

$$[0186] \quad q = (1 + \Delta t \rho^+)^{1/\Delta t}$$

[0187] 并且作为用于值和相对时间之间的转换的对数的底数,q以对数标度来影响时间分辨率将如何影响值分辨率。进而,这取决于参数 $\rho^+$ ,其受制于 $\rho^+ > 0$ 。

[0188] 最后,这可被标记为神经元操作 $\tau_k = g_k^+(\{\tau_{jk}\}_j)$ 。在尖峰定时域中,

$$[0189] \quad g_k^+(\{\tau_{jk}\}_j) = \tau_k = -\log_q \left[ \Delta \omega_k + \sum_j \omega_{jk} \left( \frac{1}{q} \right)^{\tau_{jk}} \right]$$

[0190] 但在值域中,

$$[0191] \quad g_k^+(\{x_j\}_j) = x_k(t) = \sum_j \omega_{jk} x_j(t) + \eta_k$$

[0192] 这一操作可能不实际上被计算,而是相反,由受差分方程管控的尖峰发放神经元来实施等效计算,

$$[0193] \quad \frac{dv_k^+(t)}{dt} = \rho^+ v_k^+(t) + \sum_j y_{jk}(t)$$

[0194] 其中

$$[0195] \quad y_{jk}(t) = y_j(t - \Delta \tau_{jk})$$

[0196] 并且尖峰发放条件,

$$[0197] \quad v_k^+(t) > v_0 = v_k^-(\tau_k) \rightarrow y_k(t) = 1, v_k^+(t + \varepsilon) = v_0 = v_k^-(\tau_k)$$

[0198] 如果没有输入,则神经元将在以下时间激发:

$$[0199] \quad t_k^{+'} = \log_q \frac{v_k^+(t)}{v_k^+(t_k^-)} = \log_q \frac{v_\theta}{v_0}$$

[0200] 因此,最大非自指输入时间(若要使该输入影响最快尖峰发放时间)为  $\tau_{jk} \leq t_k^{+'}$  (假设最小延迟)。类似地,对于延迟:  $\Delta \tau_{jk} \leq t_k^{+'}$ 。因此,

$$[0201] \quad \omega_{jk} \geq e^{-t_k^{+'}/\Delta t/v_\theta} \text{ 或即 } \omega_{jk} \geq q^{-t_k^{+'}/v_\theta}。$$

[0202] 相反,具有分辨率  $\Delta t$  的离散时间系统中的最小非零时间延迟为  $\Delta t$ 。因此,为有非零效应,

$$[0203] \quad \omega_{jk} \leq e^{-\Delta t/v_\theta} \text{ 或即 } \omega_{jk} \leq q^{-1/v_\theta}。$$

[0204] 类似地,神经元的输出(如果非零/非瞬时的)在范围上被限定于  $[\Delta t, t_k^{+'}]$ 。因此,经编码值在以下范围中,

$$[0205] \quad q^{-t_k^{+'}} \leq x_k \leq q^{-1}$$

[0206] 可注意到,值  $x_k = 1$  产生输入延迟 0, 并且值  $x_k = q^{-t_k^{+'}} \cong 0$  产生输入延迟  $t_k^{+'}$ 。因此,一般可用范围  $[0, 1]$  中的输入来操作。然而,由于权重范围,各输出可被偏移达  $\Delta \omega_k$  并且最多是  $N/qv_\theta$  (对于  $N$  个输入)。因此,各输出的含义可如下被任意性地重新定义:

$$[0207] \quad x_k' = \frac{(x_k - \Delta \omega_k)v_\theta}{N} \cong x_k v_\theta / N$$

[0208] 这在将一个神经元的输出馈送至另一神经元的输入方面意味着什么将在以下描述中解释。

[0209] 概言之,现已提供了用于线性表达的通用计算的神经网络,其强大到在给定了足够时间分辨率的情况下足以用单个神经元来计算任意精度的值。该神经元是抗泄漏积分激发 (ALIF) 神经元,其电压在尖峰发放之后或者在不应期之后复位至零以上的标称设置。该神经元的突触没有权重并且没有突触后滤波器。然而,这些连接具有延迟。图6是根据本公开的某些方面的单个抗泄漏积分激发 (ALIF)  $g_k^+$  神经元  $k$  的架构的框图600。

[0210] 神经元的模拟或实现可在连续时间或离散时间中进行。离散时间操作可通过以上离散时间方程的迭代来逐步行进。然而,连续时间(和离散时间)实现还可如下在时间  $t$  时发生输入事件之际以基于事件方式来执行:

[0211] 1. 基于在先前事件时间  $t_k^{last}$  处的状态、使用连续时间或离散时间方程来更新神经元状态变量  $v_k^+(t)$ ,

$$[0212] \quad v_k^+(t) = e^{\rho+(t-t_k^{last})} v_k^+(t_k^{last}) \text{ 或者 } v_k^+(t_k) = q^{(t-t_k^{last})} v_k^+(t_k^{last})。$$

[0213] 2. 添加输入,

$$[0214] \quad v_k^+(t) = v_k^+(t) + 1 \text{ (对于兴奋性)}$$

$$[0215] \quad v_k^+(t) = v_k^+(t) - 1 \text{ (对于抑制性)}$$

[0216] 3. 检查尖峰,如果  $v_k^+(t) \geq v_\theta$ , 则  $y_k(t) = 1$  且复位  $v_k^+(t) = v_0$ 。

[0217] 4. 在时间  $t + \Delta \tau_{kl}$  处,为具有来自神经元  $k$  的突触的每个神经元  $l$  调度事件

[0218] 5. 使用连续时间或离散时间方程来检查该神经元何时将在无需进一步输入的情况下激发,

$$[0219] \quad t_k^{\text{next}} = t + \log(v_0/v_k^+(t))/\rho^+ \text{ 或}$$

$$[0220] \quad t_k^{\text{next}} = t + \log_q(v_0/v_k^+(t))/\rho^+$$

[0221] 6. 将当前调度的“在无进一步输入情况下激发”事件重新调度到时间  $t_k^{\text{next}}$ 。注意，如果该神经元在没有进一步输入的情况下将不会激发，则时间  $t_k^{\text{next}}$  可有效地被设为无穷，或某个足够大的数或者其指示符（替换地，设为过去时间从而将永不被执行）。

[0222] 7. 当发生“在无进一步输入情况下激发”事件之际设置  $y_k(t)=1$  并复位  $v_k^+(t) = v_0$ 。

[0223] 注意，如果在离散时间中操作，则事件的经调度时间可以被舍入，或者另行被转换为时间分辨率  $\Delta t$  的最接近倍数，例如，

$$[0224] \quad t_k^{\text{next}} = \text{ceil}(t_k^{\text{next}} / \Delta t) \Delta t$$

[0225] 如上所述，输入中的信息被编码在输入尖峰时间702与突触后神经元（例如，神经元k）或相对于突触后神经元的非自指形式或即（NSR-POST）的在前输出尖峰时间704之间的尖峰时间差（ $\tau_{jk}$ ）中，如图7中所解说的。与此形成对比的是，输出中的信息被编码在突触后神经元或自指（SR）形式的连贯尖峰之间的时间差（ $\tau_k$ ）中。

[0226] 以上描述的方法可以被推广，并且对于输入或输出形式可使用SR或NSR（无论NSR是相关于突触后神经元还是某个第三神经元）。以下更详细地描述这一推广。

[0227] 转换外部输入或输出的信息

[0228] 输入可被提供给突触作为尖峰串  $y_i(t)$ 。然而，感官输入序列通常呈实值形式  $x_i(t)$ 。此种序列可以若干方式被转换为SR或NSR尖峰串。首先，考虑基本形式。

[0229] 序列  $x_i(t)$  可根据以下算法被转换为SR时间编码尖峰序列  $y_i(t)$ ：

[0230] 1. 令  $t_i^-$  为神经元i上一次发放尖峰的时间（任意地初始化为0）（即，将  $t_i^-$  设为  $t_i^+$  的前前值）。

[0231] 2. 计算  $\tau_i = -\log_q \bar{x}_i(t)$ ，其中  $\bar{x}_i(t)$  是  $x_i(t)$  在  $t_i^-$  之后一短时段T（或者替换地，在  $t_i^-$  之前一短时段T）上的函数（诸如均值）。

[0232] 3. 令下一尖峰的时间为  $t_i^+ = t_i^- + \tau_i$ 。

[0233] 为了执行反变换， $x_i(t)$  可被设为  $x_i(t_i^- < t)$  直至在时间t处发生尖峰。随后，设置  $x_i(t) = q^{-\tau_i}$ ，其中  $\tau_i = t - t_i^-$ 。

[0234] 序列  $x_i(t)$  可根据以下算法被转换为NSR时间编码尖峰序列  $y_i(t)$ ：

[0235] 1. 令  $t_k^-$  为神经元k上一次发出尖峰的时间（任意初始化为0）（即，将  $t_k^-$  设为  $t_k^+$  的前前值）。

[0236] 2. 计算  $\tau_{ik} = -\log_q \bar{x}_i(t)$ ，其中  $\bar{x}_i(t)$  是  $x_i(t)$  在  $t_k^-$  之后一短时段T（或者替换地，在  $t_k^-$  之前一短时段T）上的函数（诸如均值）。

[0237] 3. 令下一尖峰的时间为  $t_i^+ = t_k^- + \tau_{ik}$ 。

[0238] 为了执行反变换， $x_i(t)$  可被设为  $x_i(t_i^- < t)$  直至在时间t处发生尖峰。随后，设置  $x_i(t) = q^{-\tau_{ik}}$ ，其中  $\tau_{ik} = t - t_k^-$ 。

[0239] 这时，现在可以显见的是，可以使用单个神经元来实现从任何数目的输入到单个输出的线性变换。然而，通过使用多个神经元，从N个输入到M输出的任何变换，

$$[0240] \quad x_{\text{out}}(t) = Wx_{\text{in}}(t)$$

[0241] 可由具有在给定了矩阵W的各行的情况下计算出的延迟的M个神经元来计算。此外,这些输出中的一些输出可被反馈作为输入(即,可能存在交叉)。

[0242] 以上,仅考虑了对于每个输入帧而言将最快发生的尖峰。这是神经元模型中的微妙之处。在实效上,ALIF神经元是如上所述地用激发之际的附加操作来建模的(即,当检查有无尖峰时,如果 $v_k^+(t) \geq v_{th}$ ,则 $y_k(t) = 1$ ,复位 $v_k^+(t) = v_{th}$ 并清除 $t_{clear} < t$ 的所有输入 $y_k(t_{clear})$ )。这意味着一旦激发,神经元就遗忘尚未传播给胞体的任何输入。因此,神经元将不会仅仅是由于过剩的输入而再次激发。

[0243] 然而,这不是严格必须的。替换地,可允许神经元再次激发并且仅接受偶然误差。应对于此的另一方式是建立以类似帧的模式来驱动系统的振荡,其中仅特定输出被认为具有意义。另一方式是使用抑制来防止激发达某个时间并清除系统以供下一次计算。因此,有很多方式来规避这一点。本公开的其余部分将继续使用以上模型。

[0244] 还可注意到,如果期望的话,可为每个神经元使用不同的基数(例如,不同的q)。对于单个输入,假设 $\Delta \omega_k \rightarrow 0$ ,

$$[0245] \quad g_k^+(\tau_{jk}) = \tau_k = \Delta \tau_{jk} + \tau_{jk}$$

[0246] 如果对输入 $q_j$ 和输出 $q_k$ 使用不同的参数q,则

$$[0247] \quad \tau_{jk} = -\log_{q_j} x_j$$

$$[0248] \quad \tau_k = -\log_{q_k} x_k$$

[0249] 在值域中,这不仅提供另一缩放手段,还允许在不同范围中操作不同的神经元并使它们对接。

[0250] 对于某些方面,对以上转换的替换方案是使用代理神经元来将实值输入转换为SR尖峰或NSR尖峰(出于此目的,任何类型的神经元均可使用)。

[0251] 接着,考虑负权重(系数)。没有与值域中的负权重相对应的非复值延迟。为了应用负权重,输入可被切换至抑制性而非兴奋性。即,可仅仅是翻转该符号。回想

$$[0252] \quad \left(\frac{1}{q}\right)^{\tau_k} = \Delta \omega_k + \sum_j \omega_{jk} \left(\frac{1}{q}\right)^{\tau_{jk}}$$

[0253] 其中 $\omega_{jk} = q^{-\Delta \tau_{jk} / v_k^+(\tau_k)}$ 。因此,对于负值的 $\omega_{jk}$ ,使用 $\omega_{jk}$ 的绝对值来确定 $\Delta \tau_{jk}$ ,

$$[0254] \quad \Delta \tau_{jk} = -\log_q (|\omega_{jk}| v_k^+(\tau_k))$$

[0255] 但随后将输入设为负值以达成

$$[0256] \quad \left(\frac{1}{q}\right)^{\tau_k} = \Delta \omega_k + \sum_j \text{sign}(\omega_{jk}) q^{-\Delta \tau_{jk} / v_k^+(\tau_k)} \left(\frac{1}{q}\right)^{\tau_{jk}}$$

[0257] 图8解说了根据本公开的某些方面的导致神经元810上的兴奋性和抑制性输入806、808的正和负输入值802以及正和负缩放比例值804的所有可能组合。

[0258] 然而,不同的问题是表示同一变量的负和正值两者。克服此点的一种方式是使用输入的负值而非负权重。因此,该问题变为创建负输入以及正输入之一。

$$[0259] \quad x_i(t) = [x_i(t)]_+ - [-x_i(t)]_+$$

[0260] 然而,每一者仅可呈现正幅值(即,负输入不能表示双重负或正值,而正输入不能表示负值)或零。注意,零转译为相对于输入定时而言的无穷(或非常大)。

[0261] 现在,可对以上权重否定做等效处理以应对具有正权重的负输入。负值可用神经元被正编码  $[-x_i(t)]_+$  并连接为抑制性输入808来表示,如图9的上方示图902中所解说的。如果值和权重两者都是负的,则它们抵消并且不需要做任何别的事。尽管上方示图902解说了针对输入神经元906表示正和负值,但图9的下方示图904解说了针对输入神经元和输出神经元908两者表示正和负值。

[0262] 关键是为输出将域(+ve)和(-ve)(如果两者均被涉及)分开为分别的表示,并且随后应输入来重新组合它们(使用兴奋性和抑制性)。如果输入被约束于特定域(例如,正),则这一分开不需要进行。

[0263] 示例A:缩放一标量

[0264] 本公开的某些方面可用缩放示例来演示。在这一示例中,尖峰发放神经元可执行  $x_k(t) = a_{ik}x_i(t)$ 。换句话说,期望输出是输入的经缩放版本。这一示例在使用输出神经元的尖峰作为参考时间的异步帧模式中使用全尖峰网络(所有输入和输出均呈尖峰的形式)。回想时间分辨率可以是无穷(连续的)或离散的(无论以固定还是可变步长)。

[0265] 图10是根据本公开的某些方面的用于使用神经元模型来缩放标量值的示例操作1000的流程图。操作1000可始于1002和1004处的初始化。在1002,可计算与期望的线性计算的系数  $a_{ik}$  相对应的突触延迟  $\Delta \tau_{ik}$ 。在1004,各延迟可被量化到时间分辨率  $\Delta t$ 。其后,操作1000进入环路。在1006,输入值  $x_i(t)$  在输出神经元k的尖峰发放之际被采样。在1008,值  $x_i(t)$  被转换为相对于神经元k的上一尖峰而言的尖峰时间  $\tau_{ik}$ 。在1010,输入尖峰时间被量化到时间分辨率  $\Delta t$ 。在1012,输入尖峰在时间偏移  $\tau_{ik} + \Delta \tau_{ik}$  处被提交给神经元k的胞体。在1014,神经元k的输出尖峰时间  $\tau_k$  以分辨率  $\Delta t$  被确定。

[0266] 操作1000中的环路的每轮迭代对应于神经元k的一个输出尖峰。因此,定时是异步的,因为帧历时取决于神经元k的尖峰间区间。当使用这一种类的异步帧(参考)时,输入的采样率是可变的。具体地,如果总输入是大的(值域)因此输入延迟是小的,则神经元k较早激发并且因此,在短时间里再次对输入进行采样。如果总输入是小的(值域),则发生相反情况。因此,输入的采样率与输出值的幅值成比例。这具有优势,因为显著输入值被更频繁地采样,而不显著输入倾向于以低速率或最小的速率被采样,由此节省计算能力和资源。

[0267] 为了看到这是如何工作的,期望的标量线性结果可用仅单个尖峰发放神经元k来获得。如上所述,为了初始化,可计算  $\Delta \tau_{ik} = -\log_2(v_k^*(\tau_k))$ 。对于这一示例,离散时间系统(而非连续时间)被用于演示时间分辨率的效应。设想对于这一示例,  $\Delta t = 0.1\text{ms}$  并令  $\rho^+ = 0.1$ ,任意小的  $\Delta \omega_k = 0.01$ ,有  $v_0 = 1$ 。这得到  $v_0 = 100$  且最小神经元激发率为约21Hz或48ms的时段(在没有任何输入的情况下)。参数  $q \cong 1.1$ 。可操作系数范围是0.0001到0.01。为这一示例选取任意系数  $a_{ik}$ (例如,系数最大值的1/2),且提交如下形式的经时间偏移正弦:

[0268]  $x_i(t) = (1 + \sin(2\pi f_i(t + T_i))) / 2$

[0269] 如果选取  $a_{ik} = 1/2$  则输出将在0.01(由于  $\Delta \omega_k$ )到最大值0.02的尺度上。

[0270] 图11A解说了根据本公开的某些方面的当使用0.1ms的时间分辨率来对单个神经元模型的单个树突输入上的标量输入进行缩放时随时间推移的示例输入值的图表1102、随时间推移的输出值的图表1104、以及线性度图表1106。图11B解说了根据本公开的某些方面的当使用1.0ms的时间分辨率来对单个神经元模型的单个树突输入上的标量输入进行缩放时随时间推移的示例输入值的图表1112、随时间推移的输出值的图表1114、以及线性度

1116。如所解说的，神经元模型在时间编码域中是高度线性的。当然，精度取决于时间分辨率。因此，当时间分辨率降级10倍时，在输出中因时间分辨率导致的误差变得可察觉。

[0271] 示例B:线性变换

[0272] 在这一示例中，使用同一示例A，但添加了更多输入。期望的神经元模型是

$$[0273] \quad x_k(t) = \sum_i a_{ik} x_i(t)$$

[0274] 系数完全如在以上示例中那样被设置。使用十个输入，并跨该范围选取任意系数，比方最大系数值的分数 [0.5 0.65 0.55 0.5 0.39 0.59 0.4 0.81 0.87 0.35]。可使用与先前示例相同的输入格式(经偏移正弦)，区别在于将每个输入偏移达一不同的量。从三角法中知道，如果频率是相同的，则 $x_k(t)$ 的结果是经缩放正弦，因此这一示例针对每个输入使用不同频率以使其更为有趣。使用单个神经元来计算这一点，并且根据系数指派突触延迟，就像在以上示例中，得到以ms计的 [6.9 4.3 6 7 9.5 5.3 9.2 2.1 1.4 10.5] 的延迟。

[0275] 图12A解说了根据本公开的某些方面的当使用0.1ms的时间分辨率来对单个神经元模型的十个树突输入上的标量输入进行缩放时随时间推移的示例输入值的图表1202、随时间推移的输出值的图表1204、以及线性度图表1206。图12B解说了根据本公开的某些方面的当使用1.0ms的时间分辨率来对单个神经元模型的十个树突输入上的标量输入进行缩放时随时间推移的示例输入值的图表1212、随时间推移的输出值的图表1214、以及线性度图表1216。注意当 $\Delta t = 1\text{ms}$ 时，突触延迟还损失分辨率并变为 [7 4 6 7 10 5 9 2 1 11]。对时间分辨率的敏感度随添加输入而降低。可在图12B的线性度图表1216中的结果的垂直槽的数目中看到这一点。有效地，输出定时的范围随更多输入变得更大。这意味着对于 $\Delta t = 0.1\text{ms}$ 情形在时间上的平均输出误差低于0.01ms并且对于 $\Delta t = 1\text{ms}$ 情形低于0.1ms。基本上，这意味着该神经元在10个尖峰中大约有1个尖峰有一个时间分辨率的误差。

[0276] 示例C:正和负系数

[0277] 在这一示例中，使用与以上相同的示例，但包括负系数。使用十个输入，并跨该范围选取任意系数，比方分数 [0.5 0.65 0.55 0.5 -0.39 -0.59 -0.4 0.81 0.88 0.35]。这些系数的绝对值被用于计算延迟。然而，对于负值输入，突触被切换至抑制性。

[0278] 图13A解说了根据本公开的某些方面的当使用1.0ms的时间分辨率、使用正和负缩放比例值两者来对单个神经元模型的十个树突输入上的标量输入进行缩放时随时间推移的示例输入值的图表1302、随时间推移的输出值的图表1304、以及线性度图表1306。图13B解说了当使用1.0ms的时间分辨率、使用正和负缩放比例值两者但错误地忽略了要翻转到抑制性的情况下来对单个神经元模型的十个树突输入上的标量输入进行缩放时随时间推移的示例输入值的图表1312、随时间推移的输出值的图表1314、以及线性度图表1316。

[0279] 示例D:有噪ALIF

[0280] 可通过以各种适合方式中的任一种添加噪声项来将噪声添加至ALIF模型。一种简单的方式是将噪声项添加至差分方程系数 $\rho^+$ ：

$$[0281] \quad \rho^{+'} = \rho^+ + \rho_\epsilon$$

[0282] 这是噪声的激进形式，因为随着膜电位变得更大，该噪声项的同一值在膜电位中产生越来越大的变化。然而，这可争辩地是生物学上一致的，因为这是在当神经元更接近于要激发时有过剩输入命中该神经元的情况下将会发生的情况。

[0283] 在图14中所解说的这一示例中,重复以上示例C(具有正和负系数),区别在于已经添加了上述噪声项,该噪声项具有有零均值和 $\rho^+$ 值的10%的标准偏差的高斯分布,假设 $\Delta t = 1\text{ms}$ 。该噪声项被假定为是白噪声(即,加性高斯白噪声(AWGN))。

[0284] 图14解说了根据本公开的某些方面的当使用1.0ms的时间分辨率、使用正和负缩放比例值两者并且向神经元模型的膜电位添加了噪声项的情况下对单个神经元模型的十个树突输入上的标量输入进行缩放时随时间推移的示例输入值的图表1402、随时间推移的输出值的图表1404、以及线性度图表1406。可以看到,尽管在线性度图表1406中对于任何给定时间点而言对线性度均有显著的噪声破坏,但平均而言,该神经元能够非常好地遵循线性变换,如图表1404中所解说的。

#### [0285] 转换相对时间参考

[0286] 本节针对于在两种相对时间形式(即自指(SR)和非自指(NSR))之间转换。将自指(SR)时间转换为非自指突触后神经元时间(NSR-POST)是特别令人感兴趣的,因为以上描述(例如,关于图7)的神经元模型接受NSR-POST输入定时并输出SR定时。要将该神经元的输入馈送给另一个最有可能需要转换。NSR子形式的区别是突触前尖峰时间是相对于突触后神经元尖峰时间(NSR-POST)还是相对于第三神经元的尖峰时间(NSR-THIRD)之间的区别。

[0287] 关键洞察在于,以上定义的ALIT神经元模型包括在尖峰发放时间复位膜电位 $v_k^+(t) = v_0$ 。然而,将此重新定义为在尖峰发放之际复位 $v_k^+(t) = v_\phi$ (其中 $v_\phi \leq 0$ )是同样轻而易举的。在实效上,这建模了一简单神经元处在阈下条件中,直至由输入使电位增大到零以上。不失一般性,可将输入幅值设为 $v_0$ 。现在,所有SR输出定时均将被转译成相对于首先发生的输入而言的NSR定时。

[0288] 然而,如果首先发生的输入被指定为输入 $r$ ,则可跨诸神经元地来控制时间转换,因为向两个神经元(一个突触前以及一个突触后)提供相同输入 $r$ 将

$$[0289] \quad g_k^+(\{\tau_{jk}\}_j) = \tau_k$$

[0290] 转换为

$$[0291] \quad g_k^+(\{\tau_{jr}\}_j) = \tau_{kr}$$

[0292] 作为示例,图15解说了向突触前神经元1504和突触后神经元1506提供相同输入 $r$ 1502。

[0293] 然而,实际参考甚至不需要存在,因为无论哪个输入首先发生,该输入都将自动提供参考。唯一的先决条件是该信息被编码在诸输入之间的相对时间差中。这可用各种方式来达成,包括使用各输入之间的横向抑制以创建相互依赖性。替换地,该参考可通过连接所有输入作为至该参考的输入来被驱动,其中一个输入足以使该参考激发。因此,该参考将根据第一个输入激发而激发。这也解决了上述问题,并且还提供了单个元素(参考)作为对输出的相对读出的指示。图16解说了用于单个输入的该前馈情形。

[0294] 对于某些方面,还可使用反馈来完成以上操作,如图17中所描绘的。尽管这是不那么普遍的情形(因为紧接着的下游神经元被用作参考),但这是要示出的自然情形,因为它正好对应于 $\tau_{jk}$ 作为输入以计算 $\tau_k$ 。

[0295] 以上修改(即,将神经元的电位复位到不同于 $v_0$ 的值)可被用于另一目的。设想在尖峰发放(或其它参考时间)之际,不是复位 $v_k^+(t) = v_0$ ,而是令 $v_k^+(t) = v_+$ 。如果 $v_+ > v_0$ ,则

这将使得下一次激发提前(即,减小 $\tau_k$ )。替换地,如果 $v_+ < v_0$ ,则这将使下一次激发延迟(即,增大 $\tau_k$ )。这正是假定了该神经元的配置保持为在给定了 $v_0$ (即, $v_0$ 的值不被改变为等于 $v_+$ )的情况下所计算出那样。例如,假定在参考时间和假想时间之间本将没有输入,则神经元电位在它已被复位到 $v_0$ (而非 $v_+ > v_0$ )的情况下本将已经达到 $v_+$ , $\tau_k$ 上的变化由下式给出

$$[0296] \quad \Delta \tau_k = \log_q (v_0/v_+)$$

[0297] 相反情形(增大)可通过设置 $v_+ < v_0$ 来达成。回想值 $\tau_k = -\log_q x_k$ 的定义,以上情形对该值的效应为,

$$[0298] \quad x_k' = w_k' x_k = q^{-\Delta \tau_k} q^{-\tau_k}$$

[0299] 或即 $w_k' = q^{-\Delta \tau_k}$ 。因此,可简单地通过调节复位电压 $v_+$ 来将 $g_k^+$ 神经元的输出在值域中按因子 $w_k'$ 来任意性地变更。

$$[0300] \quad v_+ = v_0 q^{\Delta \tau_k} = v_0 w_k'^{-1}$$

[0301]  $g_k^+$ 神经元还可针对输入和输出在不同时间范围中操作。输出定时可一般在 $[\Delta t, t_k^+]$ 范围中变化,其中 $t_k^+ = \log_q \frac{v_0}{v_+}$ (假定复位电位为 $v_0$ )。该上界是神经元将在没有输入的情况下将会激发的时间。然而,因为输入定时和输入延迟的组合之故,最小观察到的时间通常将不会是 $\Delta t$ 。这将在实效上相对于上界压缩输出时间范围,从而可能期望重新扩展这一范围以供用于至后续神经元的输入。这可通过使用从先前参考延迟达所牵涉量的新参考来容易地实现。

[0302] 设想最小延迟是 $\tau_k^{\min}$ ,从而

$$[0303] \quad x_k' = q^{-\tau_k + \tau_k^{\min}}$$

[0304] 因此,通过改变为具有晚了(可以替换地是早了) $\tau_k^{\min}$ 的尖峰定时的参考,就可通过使时间范围发生偏移来对该值缩放 $q^{\tau_k^{\min}}$ 。例如,如果最小延迟 $\tau_k$ 为20ms,则可通过使用晚了20ms的参考来扣除20ms。这将把值域量缩放直至更大的范围。尽管以上使用了方程的离散时间形式,但如上所演示的,可以连续形式来进行类同的操作。

[0305] 对于某些方面,可使用传播参考波来对多神经元系统(例如,图1的神经元系统100)进行定时。传播参考波充当神经元的参考复位时间,其在不同时间复位不同的神经元。因此,通过提供恰当延迟的参考变化,突触前神经元的输出就可被恰当地转换为突触后神经元的输入。

[0306] 图18解说了根据本公开的某些方面的用于一系列神经元的传播参考波。在图18中,层n-1神经元1802接收输入r1502作为参考。输入r1502被延迟达第一延迟( $z_1$ )1804,并且经延迟的r(经延迟的')用作层n神经元1806的参考。经延迟的r进一步被延迟达第二延迟( $z_2$ )1808,并且结果所得的经延迟r(经延迟的'')用作层n+1神经元1810的参考。第一延迟1804可与第二延迟1808相同或不同。

[0307] 对于某些方面,该传播参考帧波还可被用作输入(写入)和输出(读出)的参考。该参考还可被提供为跨诸层传播的背景噪声或振荡电平。该参考帧波还可以如上所提及地是自生成的(即,由先前(诸)层、或由随后(诸)层的先前帧、或其组合)。

[0308] 该帧波的另一可任选好处在于它提供应对可能会导致突触后神经元的过剩激发(除非被清除掉)的后来输入的替换方式:使用兴奋和抑制波中的参考来在系统中对输出进

行时钟定时,从而仅对跟随参考之后规定时间以内的输出采取行动。

[0309] 图19解说了根据本公开的某些方面的具有图18的传播参考波的一系列神经元的示例时序图1900。层n-1神经元1802可被认为是神经元i,层n神经元1806可被认为是神经元j,而层n+1神经元1810可被认为是神经元k。

[0310] 对于某些方面,替换方式可以是使用被认为是 $g^-$ 神经元的神经元,如图20中所描绘的。这一类型的神经元的目的是将SR转换为NSR,或即将 $\tau_j$ 转换为 $\tau_{jr}$ ,如图20中所描绘的。一种完成此举的方式是以针对 $g^-$ 神经元的特殊动态操作,其中SR输入定时被锁存并被用于驱动该神经元的去极化。通过对参考输入(其与其它输入不同地对待)的去极化施加条件,就可用NSR形式确定输出定时。这在下面更详细地解释。

[0311] 令神经元 $g_j^-$ 的动态由下式来定义

$$[0312] \quad \frac{du_j(t)}{dt} = \alpha u_j(t) + y_{c^-(j)}(t)$$

$$[0313] \quad \frac{d\bar{u}_j(t)}{dt} = \lambda \bar{u}_j(t) + u_j(t) y_{c^-(j)}(t)$$

$$[0314] \quad \frac{dv_j^-(t)}{dt} = \beta v_j^-(t) + y_{c^-(j)}(t) \bar{u}_j(t)$$

[0315] 其中 $u_j$ 是输入状态, $\bar{u}_j$ 是输入锁存,并且 $v_j^-$ 是神经元膜电位或即神经元的状态。

[0316] 取拉普拉斯变换得到

$$[0317] \quad U_j(s) = \frac{u_j(0) + Y_{c^-(j)}(s)}{s - \alpha}$$

$$[0318] \quad \bar{U}_j(s) = \frac{\bar{u}_j(0) + U_j(s) Y_{c^-(j)}(s)}{s - \lambda}$$

$$[0319] \quad V_j^-(s) = \frac{v_j^-(0) + Y_{c^-(j)}(s)}{s - \beta}$$

[0320] 令 $c^-(j) = i$ 以及 $y_i(t) = \delta(t - t_i^-) + \delta(t - t_i^- + \tau_i)$ 。假定 $u_j(t_i^- - \varepsilon) = 0$ ,

$$[0321] \quad U_j(s) = \frac{e^{-t_i^- s}}{s - \alpha}$$

[0322] 在假定 $t \geq t_i^- + \tau_i$ 的情况下,计算拉普拉斯逆变换,

$$[0323] \quad u_j(t) = e^{\alpha(t-t_i^-)}$$

[0324] 假定 $\bar{u}_j(t_i^- + \varepsilon) = 0$ 且 $\lambda = 0$ ,

$$[0325] \quad \bar{u}_j(t) = u_j(t_i^- + \tau_i) = e^{\alpha \tau_i}$$

[0326] 进一步假定初始化至 $v_j(0) = 0$ ,

$$[0327] \quad V_j^-(s) = \frac{e^{-t_i^- s} + e^{\alpha \tau_i}}{s - \beta}$$

[0328] 在假定 $t \geq t_{c^-(j)}^-$ 的情况下,取拉普拉斯逆变换,

$$[0329] \quad v_j^-(t) = e^{\alpha \tau_i} e^{\beta(t-t_{c^-(j)}^-)}$$

[0330] 定义一旦在条件阈值 $v_j^-(t = t_j^+) \geq \bar{v}_j^-$ 之际尖峰发放即进行的操作为

$v_j^-(t_j^+) = 0, y_j(t_j^+) = 1$  (否则  $y_j(t_j^+) = 0$ ),

$$[0331] \quad \tau_{ij} \triangleq (t_{ci(j)}^+ - t_{ci(j)}^-) = \frac{1}{\beta} [\log v_j^- - \alpha \tau_i]$$

[0332] 因此, 可以看到, 从SR  $\tau_i$ 到NSR  $\tau_{ij}$ 的变换可被建模为

$$[0333] \quad \tau_{ij} = g_{ij}^-(\tau_i) \triangleq \eta^- + \mu^- \tau_i$$

$$[0334] \quad \eta^- = \frac{1}{\beta} \log v_j^-, \mu^- = -\frac{\alpha}{\beta}$$

[0335] 为了得到线性度,  $\mu^- = 1 \rightarrow \alpha = -\beta$ 。为了得到等效性,  $\eta^- = 0$ 。然而, 与之形成对比的是, 令

$$[0336] \quad \eta^- = -T^- \rightarrow v_j^- = e^{-\beta T^-}$$

[0337] 可类同地以离散时间分辨率  $\Delta t$ 进行同样操作:

$$[0338] \quad u_j(t) = (1 + \Delta t \alpha) u_j(t - \Delta t) + y_{c-(j)}(t - \Delta t)$$

$$[0339] \quad \bar{u}_j(t) = (1 + \Delta t \lambda) \bar{u}_j(t - \Delta t) + u_j(t - \Delta t) y_{c-(j)}(t - \Delta t)$$

$$[0340] \quad v_j^-(t) = (1 + \Delta t \beta) v_j^-(t - \Delta t) + y_{c-(j)}(t - \Delta t) \bar{u}_j(t - \Delta t)$$

[0341] 为了得到延迟分辨率  $\Delta \tau$ ,

$$[0342] \quad \bar{u}_{max} = (1 + \Delta t \alpha)^{\Delta \tau / \Delta t}$$

$$[0343] \quad v_j^- = (1 + \Delta t \beta)^{\Delta \tau / \Delta t} \bar{u}_{max} = (1 - (\Delta t \beta)^2)^{\Delta \tau / \Delta t}$$

[0344] 相当于以上,

$$[0345] \quad e^{-\beta T^-} = (1 - (\Delta t \beta)^2)^{\Delta \tau / \Delta t}$$

[0346] 因此,

$$[0347] \quad \beta = -\frac{\Delta \tau}{T^- \Delta t} \log(1 - (\Delta t \beta)^2)$$

[0348] 一般化地综合起来, 可选择  $\Delta t$  (例如  $\Delta t = 1$ )、 $\Delta \tau$  (例如  $\Delta \tau = T^-$ ) 以及  $\alpha = -\beta = \log(1 - (\Delta t \beta)^2) \cong 0.2$ 。

[0349]  $g_j^-$  神经元形式 (标称地将SR输入变换为NSR输出) 可因此与  $g_j^+$  神经元形式 (标称地将NSR-POST输入变换为SR输出) 结合使用 (与之对接)。在实效上, 可将这些神经元连接到相反类型。如果这些神经元被连接成使得没有神经元连接到相同类型的神经元, 则不需要在SR和NSR相对定时之间的其它形式的转换。

[0350] 用输入数目来表征ALIF线性度

[0351] 本节针对于以可变数目的输入来表征以上描述的神经元模型的线性度。为了实现这一表征, 使用随机输入和随机系数。任意选择离散系统以供用于这一表征, 并且出于简单化起见, 使用  $\Delta t = 1\text{ms}$ 的时间分辨率。在图21中分别针对1、2、4、8、16和32个输入示出了线性度图表2101、2102、2104、2108、2116和2132。随着输入数目的增加, 有效分辨率因范围增加而增加。然而, 精度可能降低。这仅是均匀分布的结果。注意, 该效应可取决于输出值的范围 (以及输入和系数) 而变化。精度通常还随更好的时间分辨率而在根本上改善, 如在以上示例中示出的。

[0352] 为何要ALIF?

[0353] 在某种程度上,LIF和其它神经元模型(包括简单模型)具有线性可预测激发。不同之处在于能使用线性预测器来进行预测的保真度或准确性。

[0354] 为了理解为何建议使用ALIF神经元,在时间域中检视 $g_k^+$ 神经元形式:

$$[0355] \quad g_k^+ (\{ \tau_{jk} \}_j) = \tau_k = -\log_q \left[ \Delta \omega_k + \sum_j \omega_{jk} \left( \frac{1}{q} \right)^{\tau_{jk}} \right]$$

[0356] 其中

$$[0357] \quad q = (1 + \Delta t \rho^+)^{1/\Delta t}$$

[0358] 如果选取 $\rho^+ > 0$ ,则该模型是ALIF。但是,如果选取 $\rho^+ < 0$ ,则该模型是LIF。实际上,如果 $\rho^+ < 0$ ,则 $q < 1$ (即,分数)且 $1/q > 1$ 。具有分数底的对数对于任何大于一的自变量而言都将是负的。在没有进一步输入的情况下的激发时间是

$$[0359] \quad t_k^{\text{next}} - t = \log_q (v_\theta / v_k^+(t)) / \rho^+$$

[0360] 因此,要使该时间为正(因果),则 $v_k^+(t) > v_\theta$ ,这是自相矛盾的,因为这意味着该神经元已经发放尖峰(超过阈值)。这并不意味着LIF不是在一定程度上可预测的。它仅意味着不能像用ALIF所做的那样使用LIF来容易地设计实现理想线性可预测性的神经元模型。

[0361] 图22是根据本公开的某些方面的用于基于相对时间来从神经元模型发射输出尖峰的示例操作2200的流程图。操作2200可以硬件(例如由一个或多个处理单元)、以软件或以固件来执行。

[0362] 该操作可在2204通过在第一神经元模型处接收至少一个输入开始。该输入可包括尖峰时间域中的输入,诸如二进制值输入尖峰或尖峰串。对于某些方面,该输入可包括实值域中的输入。

[0363] 例如,第一神经元模型可以是ALIF神经元模型。对于某些方面,第一神经元模型可以具有指数性增长的膜电位并且可在没有抑制性输入的情况下继续去极化。兴奋性输入可使第一神经元模型比第一神经元模型在没有兴奋性输入情况下的激发更早地激发。

[0364] 在2206,可基于接收到的输入来确定第一神经元模型的第一输出尖峰时间与参考时间之间的相对时间。当输入包括输入值(与输入尖峰相反)时,在2206确定相对时间可包括将输入值编码为相对时间。该编码可包括计算该相对时间为输入值的对数的负数,其中该对数的底等于作为第一神经元模型的膜电位的函数的膜电位的变化系数的指数值。

[0365] 在2210,可基于相对时间从第一神经元模型发射输出尖峰。对于某些方面,在2212处,在发射输出尖峰后,可将第一神经元模型的膜电位复位至零以上的标称设置。

[0366] 根据某些方面,参考时间可包括第一神经元模型的第二输出尖峰时间,第二输出尖峰时间发生在第一输出尖峰时间之前。对于其它方面,参考时间可包括第二神经元模型的第二输出尖峰时间,其中第一神经元模型的输出被耦合至第二神经元模型的输入并且其中第二输出尖峰时间发生在第一输出尖峰时间之前。第一神经元模型可具有针对第一神经元模型的第一膜电位的第一变化系数,而第二神经元模型可具有针对第二神经元模型的第二膜电位的第二变化系数,其不同于第一变化系数。对于某些方面,第二神经元模型可使用从第一神经元模型的参考时间延迟了的另一参考时间。

[0367] 对于某些方面,操作2200可包括基于由第一神经元模型在2202建模的函数(例如,缩放函数或其它线性变换)来确定该至少一个输入中的延迟,该建模可以在但不需要在

2204处接收该至少一个输入之前发生。在2208处相对时间可基于该延迟来调整,从而输出尖峰基于经调整相对时间来发射。例如,该函数可包括乘以一标量,其中在2202处确定该延迟包括计算该标量的绝对值以确定该延迟,其中如果该标量为负,则与至第一神经元模型的输入相关联的突触被用作抑制性突触,并且其中如果该标量为正,则与至第一神经元模型的输入相关联的突触被用作兴奋性突触。对于某些方面,该函数可以是基于稳态过程或目标输出延迟的学习函数,如以下更为详细地描述的。根据某些方面,在2202处确定延迟可涉及将该延迟量化到期望的时间分辨率,其中在2208处调整相对时间包括基于经量化延迟来调整相对时间。该函数的精度可取决于时间分辨率。

[0368] 根据某些方面,在2204处接收输入可涉及以基于期望时间分辨率的采样率来对输入进行采样。在此类方面,在2206处确定相对时间可包括将相对时间量化到该时间分辨率。

[0369] 对于某些方面,操作2200可进一步包括在2214处确定第一神经元模型的输出值。该输出值可基于发射输出尖峰的时间与参考时间之间的时间差来确定,其中该输出值是第一神经元模型的膜电位的变化系数的指数值的倒数,该指数在取倒数之前被升到该时间差次幂。在2216处,该输出值可以被输出至显示器或任何其他用于指示该输出值的适合装置。

### [0370] 学习

[0371] 尖峰网络中的学习通常是使用包括长期增强(LTP)和长期抑压(LTD)两者的逐对尖峰定时依赖可塑性(STDP)规则来建模的。LTP通常在突触后神经元在突触前神经元之后激发时增大突触权重。LTD通常在相反序出现时减小突触权重。通常,对这两者使用指数模型。

[0372] 具体地,使用基本STDP规则,如果突触后尖峰时间 $t_{\text{post}}$ 在突触前尖峰时间 $t_{\text{pre}}$ 之后发生,则权重被增大,而如果次序是相反的,则权重被减小。该变化可具有如由下式决定的不同幅值:

$$[0373] \quad \delta w = \beta_{\text{LTP}}(\Delta T > 0)e^{-|\Delta T|/\tau_{\text{LTP}}} + \beta_{\text{LTD}}(\Delta T < 0)e^{-|\Delta T|/\tau_{\text{LTD}}}$$

[0374] 其中 $\Delta T = t_{\text{post}} - t_{\text{pre}}$ 和 $\beta$ 值决定了LTP和LTD的学习速率并且 $\tau$ 是LTP和LTD的指数性衰退的时间常数(这些 $\tau$ 也可以不同)。

[0375] 然而,在本文描述的一般尖峰神经计算中,权重是不必要的。然而,连接可以被静默(断开/禁用)和撤销静默(重新连接/启用)。为了理解这如何适合本公开的各方面,可考虑参考帧。对于某些方面,信息被时间编码在尖峰与另一尖峰(或参考时间)之间的相对时间中。如果输入在神经元激发之前抵达(包括突触延迟),则该输入可影响激发时间。然而,如果输入在神经元激发之后抵达,则该输入(最多)仅可影响下一激发。

[0376] 例如,如图23中所解说的,突触后神经元k的一个输入尖峰(神经元j在2302处的输出尖峰加上连接延迟)在神经元k在2304处激发之前抵达,并且是及时的从而对2304处的激发具有影响。突触后神经元k的另一输入尖峰(神经元j在2306处的输出尖峰加上连接延迟)在神经元k在2304处激发之后抵达,并且太迟以至于该帧不能对这一激发具有任何影响。

[0377] 根据本公开的各方面,在值域中, $x = q^{-\tau}$ ,从而小值对应于长时间 $\tau$ (因为一般 $q > 1$ )。因此,如果输入具有足够小的值,则它相对于输出(结果)而言可以是不显著的,从而抵达得太迟以至于不能对输出尖峰定时具有任何影响。以上,描述了防止这一迟抵达影响下一激发时间的各种方式。然而,也有学习这一点的自动方式,其通过应用类STDP规则来暂时地静默实际上不显著的输入。如果稍后该输入变得显著,则突触可以被撤销静默。

[0378] 对于本公开的某些方面,突触延迟表示值域中的权重。学习该延迟对应于学习线性变换。因此,可采用学习规则用于学习值域中的值权重(系数),并且这些权重可被转译为(尖峰定时域中的)延迟。

[0379] 替换地,可通过变换至时域并在时域中执行延迟适配规则来将适配直接应用于延迟。

[0380] 为了看到如何可完成此举,考虑对给定输入集具有输出延迟 $\tau_k$ 的神经元k。然而,令目标输出延迟为 $\hat{\tau}_k$ 。为了获得目标输出延迟,根据下式的目标输入延迟 $\{\Delta\hat{\tau}_{jk}\}$ 是期望的:

$$[0381] \quad \hat{\tau}_k = -\log_q \left[ \Delta\omega_k + \sum_j \left( \frac{1}{q} \right)^{\tau_{jk} + \Delta\hat{\tau}_{jk}} / v_\theta^+ \right]$$

[0382] 取梯度得到

$$[0383] \quad \frac{\partial \hat{\tau}_k}{\partial \Delta\hat{\tau}_{ik}} = \frac{\left( \frac{1}{q} \right)^{\tau_{ik} + \Delta\hat{\tau}_{ik}}}{v_\theta^+ \Delta\omega_k + \sum_j \left( \frac{1}{q} \right)^{\tau_{jk} + \Delta\hat{\tau}_{jk}}}$$

[0384] 取倒数得到

$$[0385] \quad \frac{\partial \Delta\hat{\tau}_{ik}}{\partial \hat{\tau}_k} = v_\theta^+ \Delta\omega_k q^{\tau_{ik} + \Delta\hat{\tau}_{ik}} + 1 + \sum_{j \neq i} q^{(\tau_{ik} + \Delta\hat{\tau}_{ik}) - (\tau_{jk} + \Delta\hat{\tau}_{jk})}$$

[0386] 这为延迟提供了学习规则,

$$[0387] \quad \Delta\Delta\hat{\tau}_{ik} = \varphi \left[ v_\theta^+ \Delta\omega_k q^{\tau_{ik} + \Delta\hat{\tau}_{ik}} + 1 + \sum_{j \neq i} q^{(\tau_{ik} + \Delta\hat{\tau}_{ik}) - (\tau_{jk} + \Delta\hat{\tau}_{jk})} \right] (\hat{\tau}_k - \tau_k)$$

[0388] 该式可被简化为(假定 $\Delta\omega_k \rightarrow 0$ )

$$[0389] \quad \Delta\Delta\hat{\tau}_{ik} = \varphi \left[ \sum_j q^{(\tau_{ik} + \Delta\hat{\tau}_{ik}) - (\tau_{jk} + \Delta\hat{\tau}_{jk})} \right] (\hat{\tau}_k - \tau_k)$$

[0390] 假定特定输入i对以上梯度的分母的贡献很小,则该式可合并甚至进一步简化为

$$[0391] \quad \Delta\Delta\hat{\tau}_{jk} = \varphi q^{\tau_{jk} + \Delta\hat{\tau}_{jk}} (\hat{\tau}_k - \tau_k)$$

[0392] 该式取决于目标激发时间与实际激发时间之差、以及输入时间与当前延迟。参数 $\varphi$ 控制适配速率。

[0393] 无论使用哪个版本,推荐不为抵达胞体时间在突触后神经元激发时间之后的那些输入调节延迟。相反,此类输入可没有延迟调节,或者如上所述地根据类STDP方式来被调节(即,递减极小量)。

[0394] 接着,考虑目标。该目标仅取决于期望从该神经元得到什么。该目标可能甚至被任意设置,并且可随后允许神经元学习各输入的系数。替换地,可有意地选取目标以利用在该范围的边界处的非线性度。该学习可被用于确定因果性输入的逻辑关系或者计算任何任意线性方程。

[0395] 最后,可注意,目标可被解读为调控神经元的激发率、活动、或资源(能量)使用的稳态过程。从这一观点来看,延迟学习规则可被称为“稳态学习”。

[0396] 示例E:学习有噪二进制输入的系数

[0397] 在这一示例中,延迟学习规则被用于学习有噪二进制输入矢量的系数。设想有15个输入(突触前神经元)并且从每个输入到一突触后神经元有一个突触/连接(对于总共15个输入)。通过重复地使突触后神经元暴露于相同的二进制输入组合(随机选取),该延迟学习规则允许该突触后神经元学习导致在期望目标时间激发的延迟。早的输入(即,短相对输入时间)可被用于表示布尔1,而迟的输入(即,长相对输入时间)可被用于表示布尔0。图24解说了根据本公开的某些方面的这15个突触前神经元(A-E) 2402和突触后神经元(F) 2404中的五个代表性神经元。

[0398] 在时间域中向输入添加所有高斯全白噪声(AWGN),其具有2.5ms的标准偏差。这是相当大量的噪声,因为它是总时间范围的很大分数。然而,本公开的各方面仍能够成功学习。

[0399] 图25A和25B解说了根据本公开的某些方面的学习输入延迟以达成目标输出延迟的示例结果。出于演示目的,示出两种不同情形。在图25A中,各延迟被初始化为很大,而目标被设为低。在图25B中,各延迟被初始化为很小,而目标被设为高。在任一情形中,学习都是成功和快速的。

[0400] 图26根据本公开的某些方面在延迟(以ms计)的图表2600和系数(值域实值)的图表2610中为每个输入(x轴)解说了学习有噪二进制输入矢量的系数的结果。注意,各延迟已经被适配成学习输入对应性。

[0401] 示例F:学习有噪实值输入的系数

[0402] 在这一示例中,延迟学习规则被用于学习有噪实值输入矢量的系数。这一示例与以上示例相同,除了输入矢量中的各值是实值而非布尔值。

[0403] 图27解说了根据本公开的某些方面的学习有噪实值输入矢量的系数的示例结果。图27的图表2700、2710和2720中的结果示出该延迟学习规则对于有噪实值输入工作起来同样良好。

[0404] 示例G:学习布尔输入的有噪因果性逻辑关系

[0405] 在这一示例中,延迟学习规则被应用以学习变化的布尔输入矢量的因果性逻辑关系。这里,输入矢量随时间变化,但施加了输入中的一致逻辑关系以便观察延迟学习规则是否能够学习到该逻辑关系是什么。在第一种情形中,选取一组三个输入(设为1)以表示OR(或)关系。在第二种情形中,选取所有输入(设为1)以表示AND(与)关系。与先前示例相同地添加噪声。对于这一示例,这些设置与先前示例是相同的。

[0406] 图28A是OR关系的第一轮迭代之后的延迟的图表2800,而图28B是AND关系的第一轮迭代之后的延迟的图表2810。可注意到,其中一些延迟在此单轮迭代之后已从初始值1ms略有增加。还可注意到,可以看出输入中的关系。在图28A中,这些突触前神经元中仅有一个早激发,从而仅三个输入早到达胞体(~10ms)。其它的则较晚(~25ms)。在图28B中,所有突触前神经元均早激发。然而,神经元尚不是在目标时间激发;确切而言,它早很多地激发。

[0407] 图29A是对应于图28A的OR关系的数轮迭代之后的延迟的图表2900,而图29B是对应于图28B的AND关系的数轮迭代之后的延迟的图表2910。现在,能看到,突触后神经元已经学习到编码了该逻辑表示(根据输入为OR或AND)的延迟。此外,输出时间是正中目标的(附图中未示出)。在图29A中,一个突触前早激发就足以(即,逻辑OR)引起突触后神经元正中目

标地激发(其它输入具有使它们太迟以致于没有作用的延迟,如在给定了其总延迟超过30ms(目标)的情况下可看到的)。在图29B中,已经学习到逻辑关系AND,因为要使突触后神经元准时激发一般需要所有输入。这极有可能涉及较大延迟。

[0408] 图30解说了用于学习逻辑关系的收敛(作为迭代轮数的函数)。图表3000解说了逻辑OR关系的收敛,而图表3010解说了逻辑AND关系的收敛。

[0409] 一般逻辑条件或因果性学习

[0410] 令 $x_i$ 为(分别)表示或假或真的逻辑变量 $i$ 的布尔值(0或1)。令真逻辑关系 $\{i\} \rightarrow j$ 由以下真因果函数来定义,

$$[0411] \quad x_j = f_c(\{i\} \rightarrow j)$$

[0412] 例如,要是仅 $A \wedge B \wedge \bar{C} \wedge \bar{D} \rightarrow E$ 和 $\bar{A} \wedge \bar{B} \wedge C \wedge D \rightarrow E$ ,

$$[0413] \quad x_E = f_c(\{A, B, C, D\} \rightarrow E) = (x_A x_B (1-x_C) (1-x_D)) + ((1-x_A) (1-x_B) x_C x_D)$$

[0414] 令线性传递函数被如下定义:

$$[0415] \quad h(\mathbf{a}_j^k, \{i\}, \mathbf{n}^k) = \sum_i a_{ij}^k \left( \frac{1-n_i^k}{2} + n_i^k x_i \right)$$

[0416] 其中 $k$ 上标指示集合中的实例,而 $\mathbf{n}^k$ 是否定置换矢量。该函数得到代表延迟(尽管不必然是延迟)的值。否定条目被编码为否定-1或非否定+1。在实效上,此举将布尔值输入转换为其根据该矢量的取负值

$$[0417] \quad h(\mathbf{a}_j^k, \{i\}, \mathbf{n}^k) = \frac{1}{2} \sum_i a_{ij}^k (1 - n_i^k) + \sum_i (n_i^k a_{ij}^k) x_i$$

[0418] 此外,令系数 $\mathbf{a}_j^k$ 被定义成使得以上函数对由 $\{x_i\}$ 表示的具有真逻辑蕴含(例如,A、B、非C、以及非D)的逻辑变量组合等于 $\theta$ ,从而对于所有 $k$ ,满足以下表达式:

$$[0419] \quad h(\mathbf{a}_j^k, \{i\}, \mathbf{n}^k) = \theta$$

[0420] 这等效于如上所述地为稳态学习设置目标。有效地,以上目标 $\theta$ 可被转译成输出延迟目标。

[0421] 代入以上方程,

$$[0422] \quad \sum_i (n_i^k a_{ij}^k) x_i = \theta - \frac{1}{2} \sum_i a_{ij}^k (1 - n_i^k)$$

[0423] 或

$$[0424] \quad \sum_i (n_i^k a_{ij}^k) \left( x_i - \frac{1}{2} \right) = \theta - \frac{1}{2} \sum_i a_{ij}^k$$

[0425] 或

$$[0426] \quad \sum_i (n_i^k a_{ij}^k) (x_i') = \theta - \frac{1}{2} \sum_i a_{ij}^k$$

[0427] 因此,有了足够的方程,在给定了数个未知数 $x_i$ 的情况下,可识别集合 $\{x_i\}$ 中的逻辑条件,例如,通过使用接收所有整合神经元输出作为输入并在达到(一致输入定时的)阈值之际激发的另一神经元。

[0428] 例如,考虑针对以上示例的一些实例。在没有否定的情况下,第一神经元(标记为 $k=1$ )能学习系数以满足真因果关系。然而,还将满足其它某些条件,

$$f_1(\{A, B, C, D\} \rightarrow E) \\ = (x_A \wedge x_B \wedge \overline{x_C} \wedge \overline{x_D}) \vee (\overline{x_A} \wedge \overline{x_B} \wedge x_C \wedge x_D) \\ \vee (\overline{x_A} \wedge x_B \wedge \overline{x_C} \wedge x_D) \vee (x_A \wedge \overline{x_B} \wedge \overline{x_C} \wedge x_D) \vee (\overline{x_A} \wedge x_B \wedge x_C \wedge \overline{x_D}) \\ \vee (x_A \wedge \overline{x_B} \wedge x_C \wedge \overline{x_D}) \quad \text{这是因}$$

为如果对于所有*i*有 $n_i^1 = +1$ ,则对于以上示例中的每个变量, $\alpha_{i,j}^1 = \theta/2$ 。因此,可无歧义地检测到此真原因。然而,如果添加了另一神经元(标记为*k*=2),则

$$[0429] \quad f_2(\{\overline{A}, B, \overline{C}, D\} \rightarrow E) \\ = (x_A \wedge x_B \wedge \overline{x_C} \wedge \overline{x_D}) \vee (\overline{x_A} \wedge \overline{x_B} \wedge x_C \wedge x_D) \\ \vee (\overline{x_A} \wedge x_B \wedge \overline{x_C} \wedge x_D) \vee (x_A \wedge \overline{x_B} \wedge \overline{x_C} \wedge x_D) \vee (x_A \wedge x_B \wedge x_C \wedge x_D) \\ \vee (x_A \wedge \overline{x_B} \wedge \overline{x_C} \wedge \overline{x_D})$$

[0430] 这里,可看到,在此组合中,这两个神经元可被用于减少歧义性,因为在组合中仅存在四个交迭的逻辑条件(前四个)。有了更多神经元,歧义性可完全被消除。然而,还可以发现对于此特定逻辑条件没有多义性的否定矢量:

$$[0431] \quad f_3(\{A, \overline{B}, \overline{C}, \overline{D}\} \rightarrow E) = (x_A \wedge x_B \wedge \overline{x_C} \wedge \overline{x_D}) \vee (\overline{x_A} \wedge \overline{x_B} \wedge x_C \wedge x_D)$$

[0432] 在这一情形中,不需要多个神经元,因为这一神经元只有在对应于真原因的逻辑条件得到满足的情况下才将在期望延迟激发。

[0433] 然而一般地,可能有动机对于可能产生的任何特定逻辑条件具有解。为此,可利用具有不同否定矢量的神经元的整合。可随后将它们的输出馈送给时间一致性识别神经元,该时间一致性识别神经元检测这些神经元中何时有足够的神经元在同一时间激发,由此无歧义地识别对应于真原因的逻辑条件。具有足够的神经元通常指当神经元集群区分真和假逻辑条件时(即,当这些神经元一起(由一致性激发来衡量达到预定时间精度)能正确预测因果关系、无效果、或其两者到期望程度或准确性时)的状态或条件。

[0434] 图31解说了根据本公开的某些方面的如何可实现否定和整合演绎两者。神经元C3102是一个输入的示例。神经元C抑制表示神经元C的否定的神经元(非C)3104。注意,如果神经元C激发推迟,则神经元(非C)将首先激发。回想短延迟意味着大值(逻辑“真”)而长延迟意味着小值(逻辑“假”)。每一者是至具有不同否定矢量的不同神经元的输入(即,神经元E<sup>1</sup>3106使用非否定C,而神经元E<sup>2</sup>3108使用否定C)。第三神经元E<sup>3</sup>(未示出)若被使用,则可取决于E<sup>3</sup>的否定矢量而使用非否定C或否定C。此外,根据每个学习神经元的否定矢量,学习神经元3106、3108中的每一者可具有其它输入,无论是否定的还是非否定的(例如,神经元A、神经元(非A)、神经元B、神经元(非B)、神经元D、或者神经元(非D))。与每个输入相关联的延迟在学习神经元中被适配以满足目标输出延迟,如上所述的。这些学习神经元3106、3108的输出作为输入被馈送至神经元R3110,神经元R3110能够识别这些学习神经元3106、3108的输出中的时间一致性(即,如果神经元E<sup>1</sup>和E<sup>2</sup>关于逻辑条件匹配达成一致)。

[0435] Hebbian学习

[0436] Hebbian学习是其中输入和输出一起激发的学习形式。Hebbian学习规则的简单形式是STDP规则,

$$[0437] \quad \Delta w_{ik} = A_{\text{sign}(\Delta T)} e^{-|\Delta T|/\tau_{\text{sign}(\Delta T)}}$$

[0438] 其中 $\Delta T = t_{\text{post}} - t_{\text{pre}} = t_k - t_i = \tau_{kr} - \tau_{ir}$ ,其中*r*是提供时间参考的参考。假定乘性改变 $w_{ik}' = w_{ik}(1 + \Delta w_{ik})$ 转换至时间域,

$$[0439] \quad q^{-\Delta\tau_{jk}'} = q^{-\Delta\tau_{jk}}(1 + \Delta w_{ik})$$

[0440] 或即

$$[0441] \quad \Delta \Delta \tau_{jk} = \Delta \tau'_{jk} - \Delta \tau_{jk} = -\log_q(1 + \Delta w_{ik})$$

[0442] 因此,

$$[0443] \quad \Delta \Delta \tau_{jk} = -\log_q(1 + A_{\text{sign}(\Delta T)} e^{-|\Delta T|/\tau_{\text{sign}(\Delta T)}})$$

[0444] 因此,Hebbian学习可被用于时间域中,以取决于输入/输出尖峰定时  $\Delta T$  来调整延迟,而无需任何权重。注意,  $\Delta T$  的值越大,对数参数的值越接近1,并且因此,延迟上的变化越小。根据长期增强(LTP)的权重增大对于正  $\Delta T$  具有正A值,并且因此,对数参数将大于一,从而得到延迟上的负变化(增大)。根据长期抑压(LTD)的权重减小对于负  $\Delta T$  具有负A值,并且因此,对数参数将小于一,从而得到延迟上的正变化(增大)。延迟越长意味着输入越不显著。延迟越短意味着输入越显著。

[0445] 图32是根据本公开的某些方面的用于在尖峰神经网络中学习以发射输出尖峰的示例操作3200的流程图。操作3200可以硬件(例如由一个或多个处理单元)、以软件或以固件来执行。

[0446] 操作3200可在3202通过对与至神经元模型的输入(例如,树突)相关联的当前延迟进行初始化来开始。对于某些方面,当前延迟可被初始化为0。

[0447] 在3204,神经元模型中的输入尖峰可根据当前延迟被延迟。输入尖峰可发生在相对于该神经元模型的参考时间而言的输入尖峰时间处。在3206,可至少部分地基于经延迟的输入尖峰来从该神经元模型发射输出尖峰。在3208,可确定输出尖峰的输出尖峰时间与该神经元模型的参考时间之间的实际时间差。

[0448] 在3210,可基于当前延迟、输入尖峰的输入尖峰、以及目标时间差与实际时间差之差来调整与该输入相关联的当前延迟。如果目标时间差与实际时间差之差大于阈值或迭代轮数尚未达到上限,则3204-3210处的操作可以经调整延迟(作为当前延迟)来重复。3204-3210处的操作可被重复数次,至少直至目标时间差与实际时间差之差小于或等于阈值或者直至已经执行了最大轮数的迭代(即,迭代轮数已经达到上限)。根据某些方面,目标时间差可以是涉及该神经元模型的稳态过程的设定点,如上所述。

[0449] 在3212,可基于经调整延迟来确定标量值。换句话说,该标量值已被该神经元模型学习到。该标量值可被确定为该神经元模型的膜电位的变化系数的指数值的倒数,该指数在取倒数之前被升至经调整延迟次幂。对于某些方面,该标量值可以是线性变换的系数。对于某些方面,该标量值可以是至显示器或任何其他用于指示该标量值的适合装置的输出。

[0450] 图33是根据本公开的某些方面的用于在尖峰神经网络中进行因果性学习的示例操作3300的流程图。操作3300可以硬件(例如由一个或多个处理单元)、以软件或以固件来执行。

[0451] 操作3300可在3302通过在一个或多个学习神经元模型处提供一组逻辑输入来开始,其中在该组逻辑输入上施加有真因果性逻辑关系。在3304,可在每组逻辑输入处接收输入尖峰之间的变化定时。对于这一个或多个学习神经元模型中的每一个神经元模型,可在3306使用接收到的输入尖峰来调整与每个逻辑输入相关联的延迟,从而学习神经元模型发射符合根据对应于该真因果性逻辑关系的一个或多个逻辑条件的目标输出延迟的输出尖峰。

[0452] 对于某些方面,对于这一个或多个学习神经元模型中的每一个神经元模型,在3306,在调整延迟之前可初始化与每个逻辑输入相关联的延迟。

[0453] 根据某些方面,在3302提供该组逻辑输入可包括从包含多个逻辑输入的群中选择每组逻辑输入。对于某些方面,该群还可包括这多个逻辑输入的否定,其中选择每组逻辑输入包括从包含这多个逻辑输入和否定的群中选择每组逻辑输入。

[0454] 操作3300可进一步包括将这多个逻辑输入中的每一个建模为输入神经元模型,并且对于这多个逻辑输入的每一个,如果一个或多个否定矢量中的至少一个具有针对该逻辑输入的否定指示,则提供表示该逻辑输入的否定的否定神经元模型,其中每组逻辑输入是根据其中一个否定矢量来选择的。在这一情形中,每个学习神经元模型可对应于这些否定矢量之一,并且对于这多个逻辑输入的每一个,根据该否定矢量,输入神经元模型的或其对应的否定神经元模型的输出可被耦合至学习神经元模型的输入。对于某些方面,这些输入神经元模型中的每一个可抑制对应的否定神经元模型。对于某些方面,否定指示可包括-1。

[0455] 操作3300可进一步包括基于来自学习神经元模型的输出尖峰的定时来确定这一个或多个学习神经元模型已经学习到对应于真因果性逻辑关系的这一个或多个逻辑条件。对于某些方面,此确定可包括确定这些学习神经元模型间的一致性或激发模式。

[0456] 根据某些方面,可将时间一致性识别神经元模型耦合至来自每个学习神经元模型的输出。该时间一致性识别神经元模型可被配置成在有阈值数目的学习神经元模型在大致相同时间激发的情况下激发。操作3300可进一步包括如果时间一致性识别神经元模型激发,则确定这一个或多个学习神经元模型已经学习到对应于该真因果性逻辑关系的至少一个逻辑条件。

[0457] 根据某些方面,在3304在每组逻辑输入处接收输入尖峰之间的可变定时可包括在该组逻辑输入处接收变化的布尔矢量。对于某些方面,在此变化的布尔矢量中,相对短的延迟表示逻辑真而相对长的延迟表示逻辑假。

[0458] 对于某些方面,经调整的延迟、这一个或多个逻辑条件、和/或此真因果性逻辑关系可以被输出至显示器或任何其他用于指示这些内容的适合装置。对于某些方面,这些学习神经元模型可包括ALIF神经元模型。

[0459] 结论

[0460] 通过将信息编码在尖峰的相对时间中,使用尖峰发放神经元实际上可存在计算上的优势。因此,本文描述了能在尖峰神经网络中高效模拟时间编码到任意性精度的神经元模型(一种模拟器设计)。

[0461] 如上所述,可使用本文公开的、利用向相对时间编码进行的对数变换的尖峰发放神经元模型来计算任何线性系统。任何个体尖峰的信息内容仅受时间分辨率所限定,因此单个神经元模型可计算任意性精度的线性变换,从而在一个尖峰中得到结果。某些方面使用抗泄漏积分激发(ALIF)神经元作为不具有突触权重或突触后滤波器的示例性神经元模型,。计算可使用时间延迟以及自指(SR)尖峰定时与非自指(NSR)尖峰定时之间的转换来发生在对数值域中。以上描述了达成此举的多种手段,包括使用第二神经元模型进行转换或使用传播参考时间帧波的方法(很像生物学振荡)。因为不需要执行权重乘法并且不需要计算突触后滤波器,所以在连续或离散时间中计算神经元输出的高效方法是可能的。

[0462] 此外,可使用包括以下两种事件类型的基于事件的调度用软件或硬件来模拟尖峰

神经网络：(1) 经延迟的突触输入事件和 (2) 预期的将来尖峰时间事件。当突触前神经元激发时，可取决于神经元之间的轴突或树突延迟为每个突触后神经元调度在将来时间的事件。当输入事件发生时，神经元的状态可自先前更新起被直接更新，而非按时间步长逐步更新。该输入可被添加，并且可直接计算将来激发时间。如果神经元在给定了当前状态的情况下将不激发，则此将来激发时间可以是无穷。无论如何，将来激发时间事件可以被重新调度。以此方式，定时上的任意高的精度（甚至连续时间）可被模拟，而无需任何附加成本，由此减少了功耗。

[0463] 以上所描述的方法的各种操作可由能够执行相应功能的任何合适的装置来执行。这些装置可包括各种硬件和/或软件组件和/或模块，包括但不限于电路、专用集成电路 (ASIC)、或处理器。一般而言，在存在附图中解说的操作的场合，那些操作可具有带相似编号的相应配对装置加功能组件。例如，图22中所解说的操作2200对应于图22A中所解说的装置2200A。

[0464] 例如，用于指示的装置可包括显示器（例如，监视器、平面屏幕、触屏等）、打印机、或任何其他用于指示值的合适装置。用于处理的装置、用于接收的装置、用于发射的装置、用于添加的装置、用于输出的装置、用于复位的装置、用于延迟的装置、用于调节的装置、用于重复的装置、用于初始化的装置、用于建模的装置、用于提供的装置、或者用于确定的装置可包括处理系统，其可包括一个或多个处理器。

[0465] 如本文所使用的，术语“确定”涵盖各种各样的动作。例如，“确定”可包括演算、计算、处理、推导、研究、查找（例如，在表、数据库或其他数据结构中查找）、查明、及类似动作。而且，“确定”可包括接收（例如接收信息）、访问（例如访问存储器中的数据）、及类似动作。同样，“确定”还可包括解析、选择、选取、建立、及类似动作。

[0466] 如本文中所使用的，引述一系列项目中的“至少一个”的短语是指这些项目的任何组合，包括单个成员。作为示例，“a、b或c中的至少一个（者）”旨在涵盖：a、b、c、a-b、a-c、b-c、以及a-b-c。

[0467] 结合本公开描述的各种解说性逻辑框、模块、以及电路可用通用处理器、数字信号处理器 (DSP)、专用集成电路 (ASIC)、现场可编程门阵列 (FPGA) 或其他可编程逻辑器件 (PLD)、分立的门或晶体管逻辑、分立的硬件组件、或其设计成执行本文中描述的功能的任何组合来实现或执行。通用处理器可以是微处理器，但在替换方案中，处理器可以是任何可商业购得的处理器、控制器、微控制器、或状态机。处理器还可以被实现为计算设备的组合，例如DSP与微处理器的组合、多个微处理器、与DSP核心协同的一个或多个微处理器或任何其它此类配置。

[0468] 结合本公开描述的方法或算法的步骤可直接在硬件中、在由处理器执行的软件模块中、或在这两者的组合中实施。软件模块可驻留在本领域所知的任何形式的存储介质中。可使用的存储介质的一些示例包括随机存取存储器 (RAM)、只读存储器 (ROM)、闪存、EPROM 存储器、EEPROM 存储器、寄存器、硬盘、可移动盘、CD-ROM，等等。软件模块可包括单条指令、或许多条指令，且可分布在若干不同的代码段上，分布在不同的程序间以及跨多个存储介质分布。存储介质可被耦合到处理器以使得该处理器能从/向该存储介质读写信息。替换地，存储介质可以被整合到处理器。

[0469] 本文所公开的方法包括用于达成所描述的方法的一个或多个步骤或动作。这些方

法步骤和/或动作可以彼此互换而不会脱离权利要求的范围。换言之,除非指定了步骤或动作的特定次序,否则具体步骤和/或动作的次序和/或使用可以改动而不会脱离权利要求的范围。

[0470] 所描述的功能可在硬件、软件、固件或其任何组合中实现。如果以硬件实现,则示例硬件配置可包括设备中的处理系统。处理系统可以用总线架构来实现。取决于处理系统的具体应用和整体设计约束,总线可包括任何数目的互连总线和桥接器。总线可将包括处理器、机器可读介质、以及总线接口的各种电路链接在一起。总线接口可用于尤其将网络适配器等经由总线连接至处理系统。网络适配器可用于实现信号处理功能。对于某些方面,用户接口(例如,按键板、显示器、鼠标、操纵杆等)也可被连接至总线。总线还可链接各种其他电路(诸如定时源、外围设备、稳压器、电源管理电路等),这些电路在本领域中是众所周知的,因此将不再赘述。

[0471] 处理器可负责管理总线和一般处理,包括执行存储在机器可读介质上的软件。处理器可用一个或多个通用和/或专用处理器来实现。示例包括微处理器、微控制器、DSP处理器、以及其他能执行软件的电路系统。软件应当被宽泛地解释成意指指令、数据、或其任何组合,无论是被称作软件、固件、中间件、微代码、硬件描述语言、或其他。作为示例,机器可读介质可以包括RAM(随机存取存储器)、闪存、ROM(只读存储器)、PROM(可编程只读存储器)、EPROM(可擦式可编程只读存储器)、EEPROM(电可擦式可编程只读存储器)、寄存器、磁盘、光盘、硬驱动器、或者任何其他合适的存储介质、或其任何组合。机器可读介质可被实施在计算机程序产品中。该计算机程序产品可以包括包装材料。

[0472] 在硬件实现中,机器可读介质可以是处理系统中与处理器分开的一部分。然而,如本领域技术人员将容易领会的,机器可读介质、或其任何部分可在处理系统外部。作为示例,机器可读介质可包括传输线、由数据调制的载波、和/或与设备分开的计算机产品,所有这些都可由处理器通过总线接口来访问。替换地或补充地,机器可读介质、或其任何部分可被集成到处理器中,诸如高速缓存和/或通用寄存器文件可能就是这种情形。

[0473] 处理系统可以被配置为通用处理系统,该通用处理系统具有一个或多个提供处理器功能性的微处理器、和提供机器可读介质中的至少一部分的外部存储器,它们都通过外部总线架构与其他支持电路系统链接在一起。替换地,处理系统可以用带有集成在单块芯片中的处理器、总线接口、用户接口、支持电路系统、和至少一部分机器可读介质的ASIC(专用集成电路)来实现,或者用一个或多个FPGA(现场可编程门阵列)、PLD(可编程逻辑器件)、控制器、状态机、门控逻辑、分立硬件组件、或者任何其他合适的电路系统、或者能执行本公开通篇所描述的各种功能性的电路的任何组合来实现。取决于具体应用和加诸于整体系统上的总设计约束,本领域技术人员将认识到如何最佳地实现关于处理系统所描述的功能性。

[0474] 机器可读介质可包括数个软件模块。这些软件模块包括当由处理器执行时使处理系统执行各种功能的指令。这些软件模块可包括传送模块和接收模块。每个软件模块可以驻留在单个存储设备中或者跨多个存储设备分布。作为示例,当触发事件发生时,可以从硬驱动器中将软件模块加载到RAM中。在软件模块执行期间,处理器可以将一些指令加载到高速缓存中以提高访问速度。随后可将一个或多个高速缓存行加载到通用寄存器文件中以供处理器执行。在以下谈及软件模块的功能性时,将理解此类功能性是在处理器执行来自该

软件模块的指令时由该处理器来实现的。

[0475] 如果在软件中实现,则各功能可作为一条或多条指令或代码存储在计算机可读介质上或藉其进行传送。计算机可读介质包括计算机存储介质和通信介质两者,这些介质包括促成计算机程序从一地向另一地转移的任何介质。存储介质可以是能被计算机访问的任何可用介质。作为示例而非限定,这样的计算机可读介质可包括RAM、ROM、EEPROM、CD-ROM或其它光盘存储、磁盘存储或其它磁存储设备、或能被用来携带或存储指令或数据结构形式的期望程序代码且能被计算机访问的任何其它介质。任何连接也被正当地称为计算机可读介质。例如,如果软件是使用同轴电缆、光纤电缆、双绞线、数字订户线(DSL)、或无线技术(诸如红外(IR)、无线电、以及微波)从web网站、服务器、或其他远程源传送而来,则该同轴电缆、光纤电缆、双绞线、DSL或无线技术(诸如红外、无线电、以及微波)就被包括在介质的定义之中。如本文中所使用的盘(disk)和碟(disc)包括压缩碟(CD)、激光碟、光碟、数字多用碟(DVD)、软盘、和蓝光<sup>®</sup>碟,其中盘(disk)常常磁性地再现数据,而碟(disc)用激光来光学地再现数据。因此,在一些方面,计算机可读介质可包括非瞬态计算机可读介质(例如,有形介质)。另外,对于其他方面,计算机可读介质可包括瞬态计算机可读介质(例如,信号)。上述组合应当也被包括在计算机可读介质的范围内。

[0476] 因此,某些方面可包括用于执行本文中给出的操作的计算机程序产品。例如,此种计算机程序产品可包括其上存储(和/或编码)有指令的计算机可读介质,这些指令能由一个或多个处理器执行以执行本文中所描述的操作。对于某些方面,计算机程序产品可包括包装材料。

[0477] 另外,应领会,用于执行本文中所描述的方法和技术的模块和/或其它恰适装置能由设备在适用的场合下载和/或以其他方式获得。例如,此类设备能被耦合至服务器以促成用于执行本文中所描述的方法的装置的转移。替换地,本文中所描述的各种方法能经由存储装置(例如,RAM、ROM、诸如压缩碟(CD)或软盘之类的物理存储介质等)来提供,以使得一旦将该存储装置耦合到或提供给设备,该设备就能获得各种方法。此外,能利用适于向设备提供本文中所描述的方法和技术的任何其他合适的技术。

[0478] 应该理解的是,权利要求并不被限定于以上所解说的精确配置和组件。可在以上所描述的方法和设备的布局、操作和细节上作出各种改动、更换和变形而不会脱离权利要求的范围。

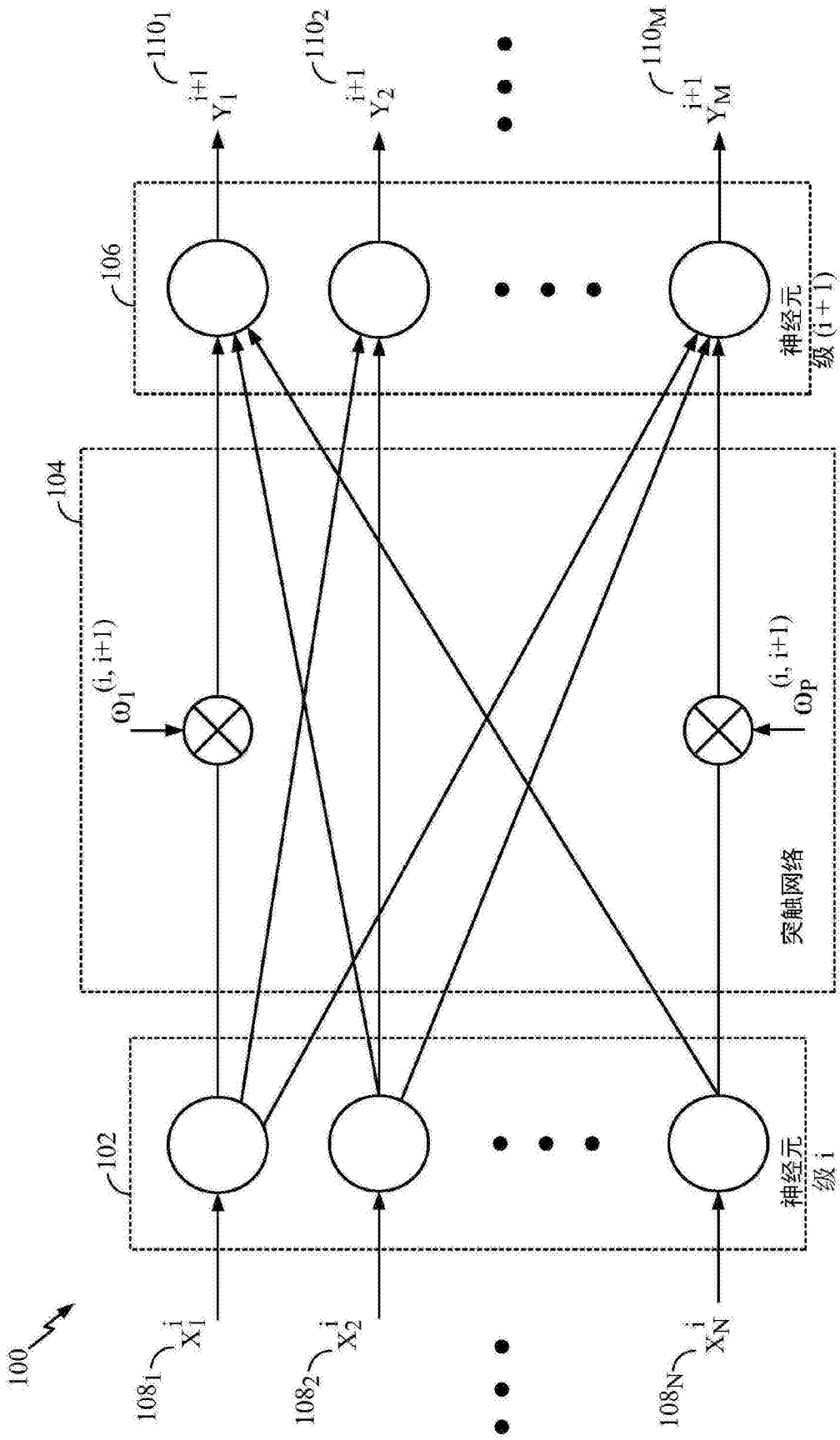


图1

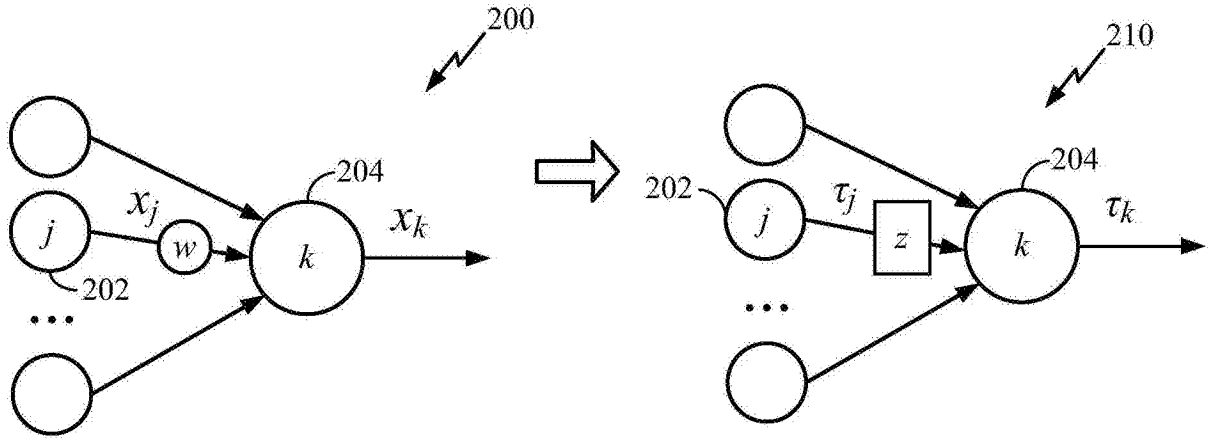


图2

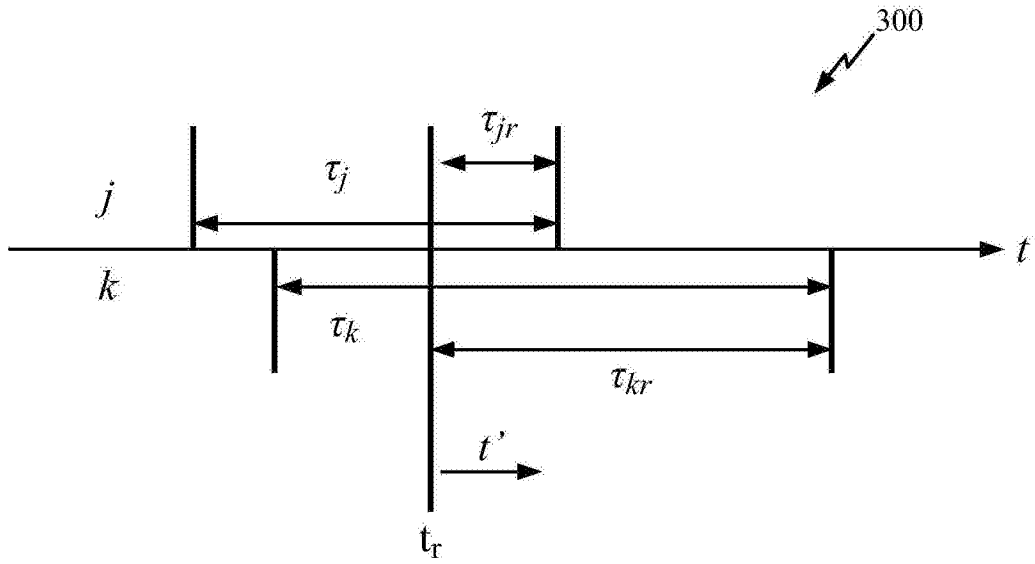


图3

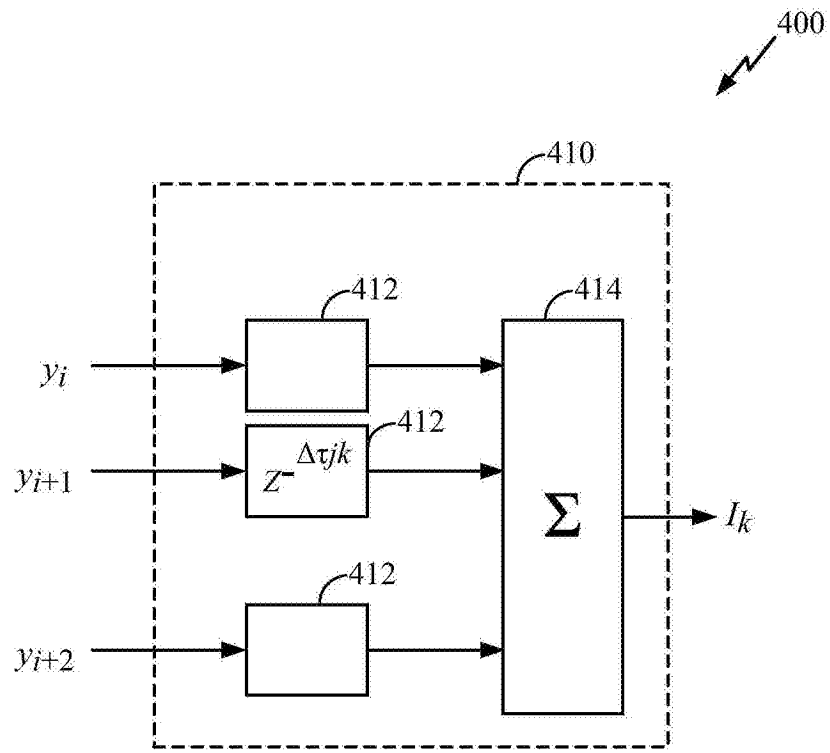


图4

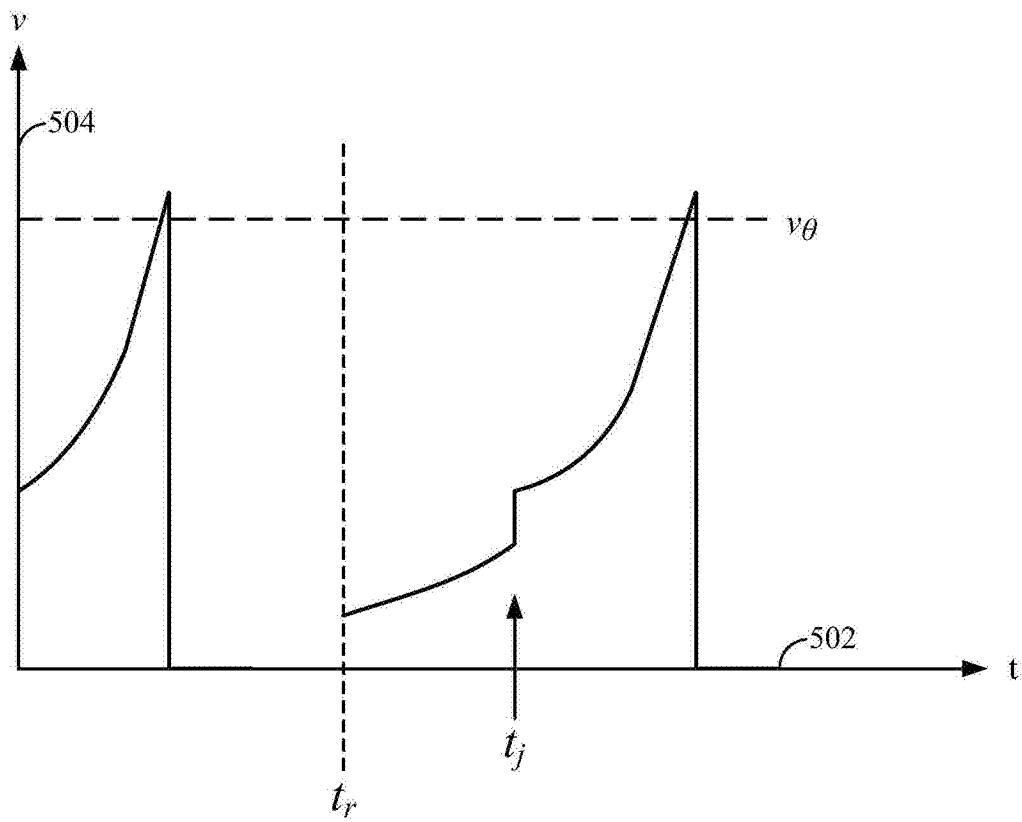


图5

600

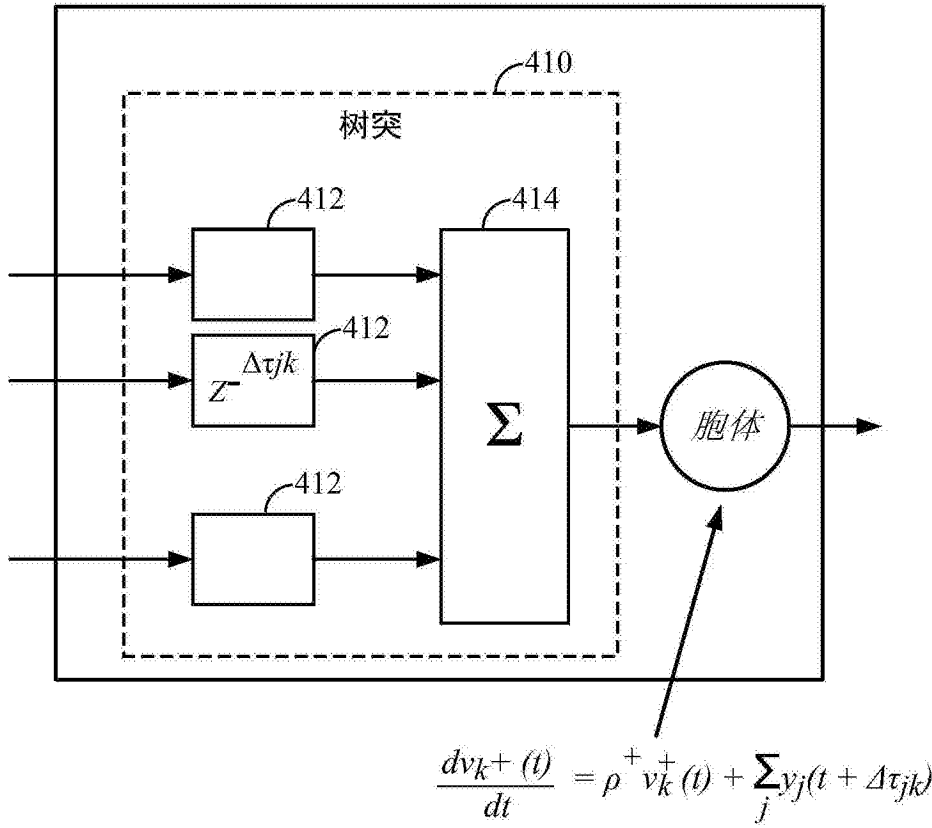


图6

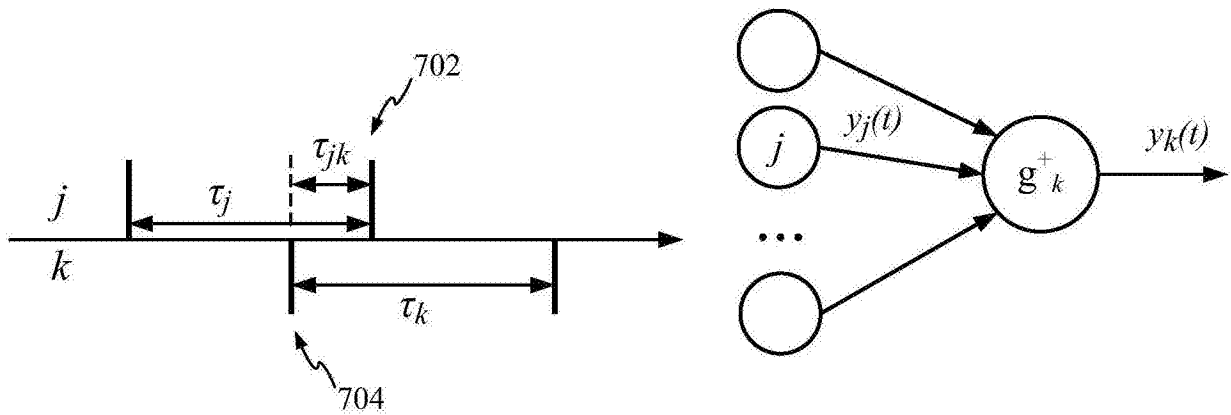


图7

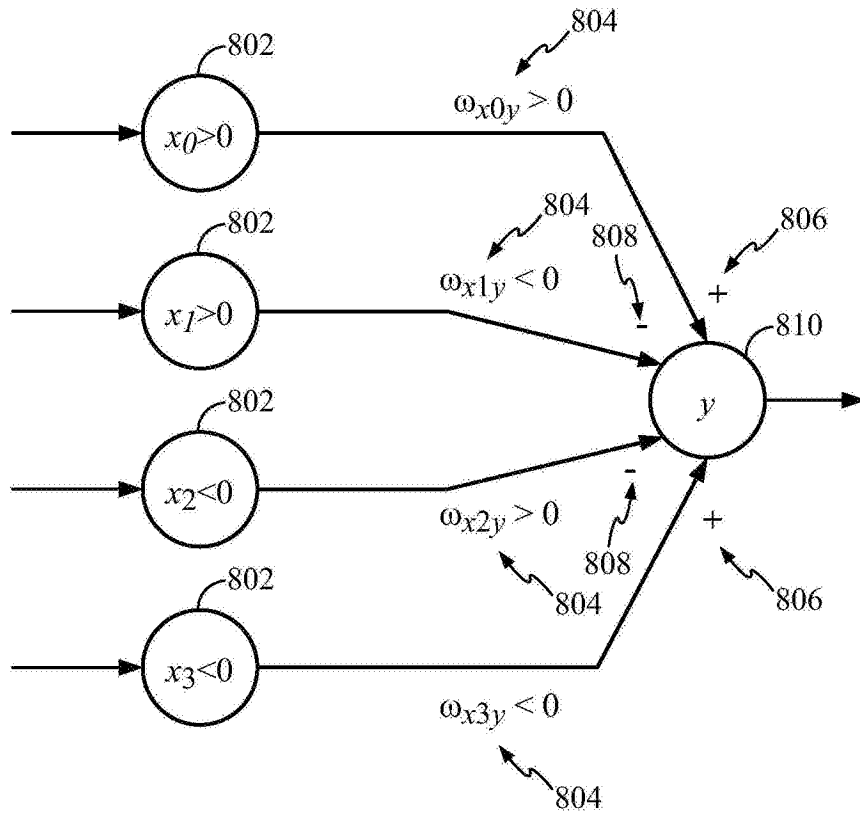


图8

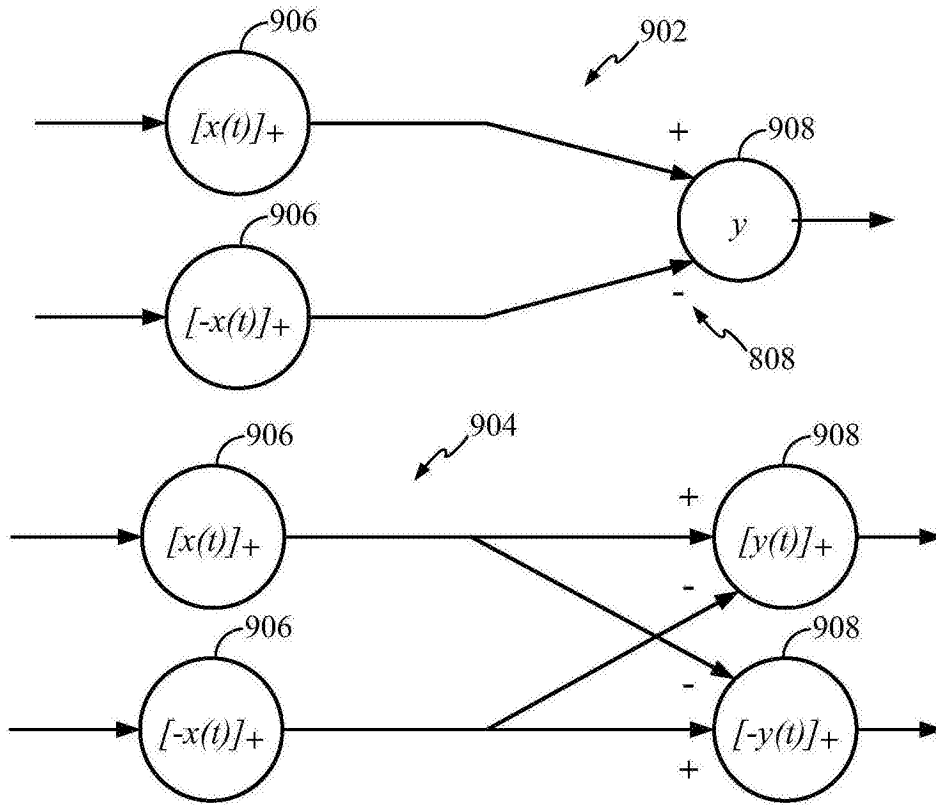


图9

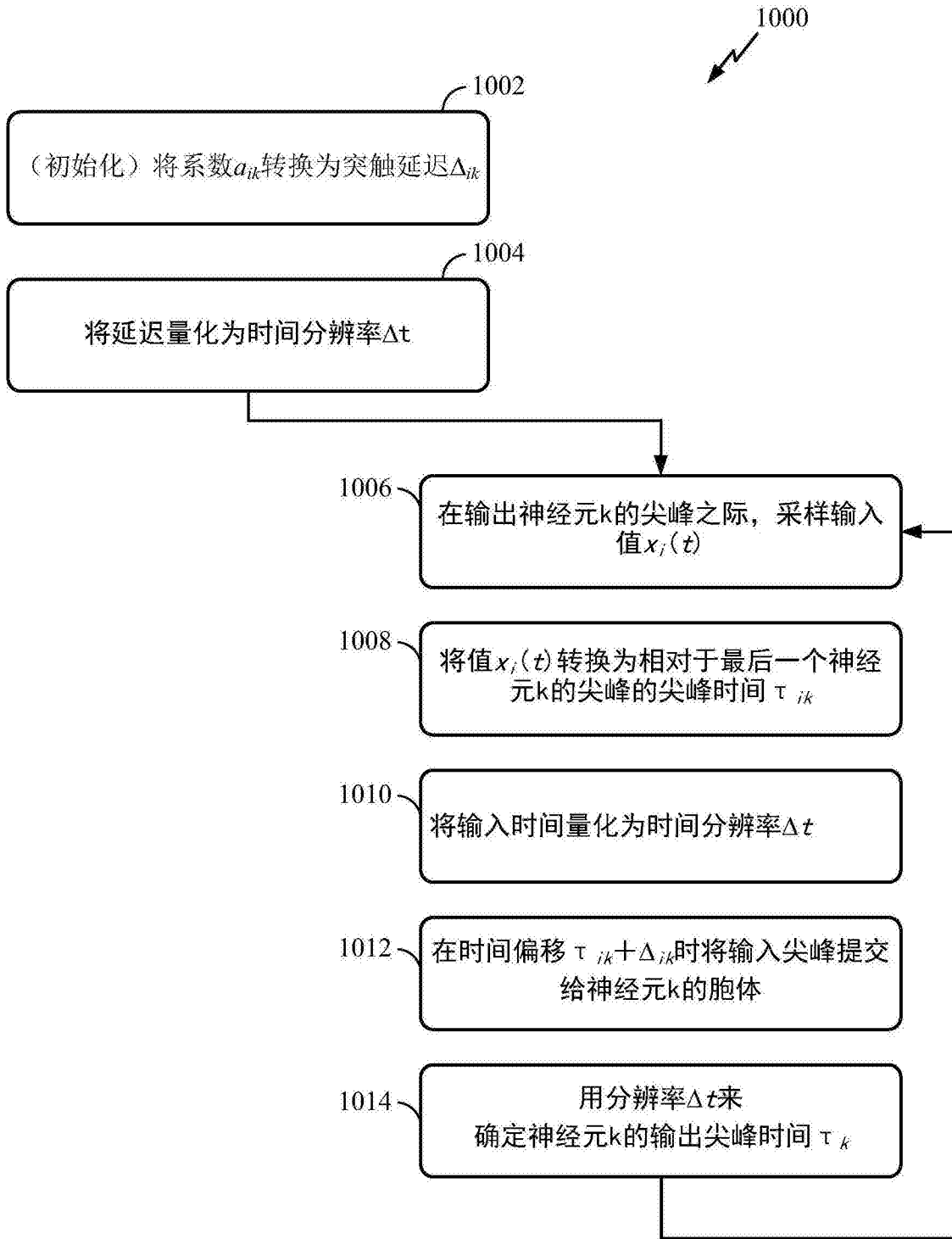


图10

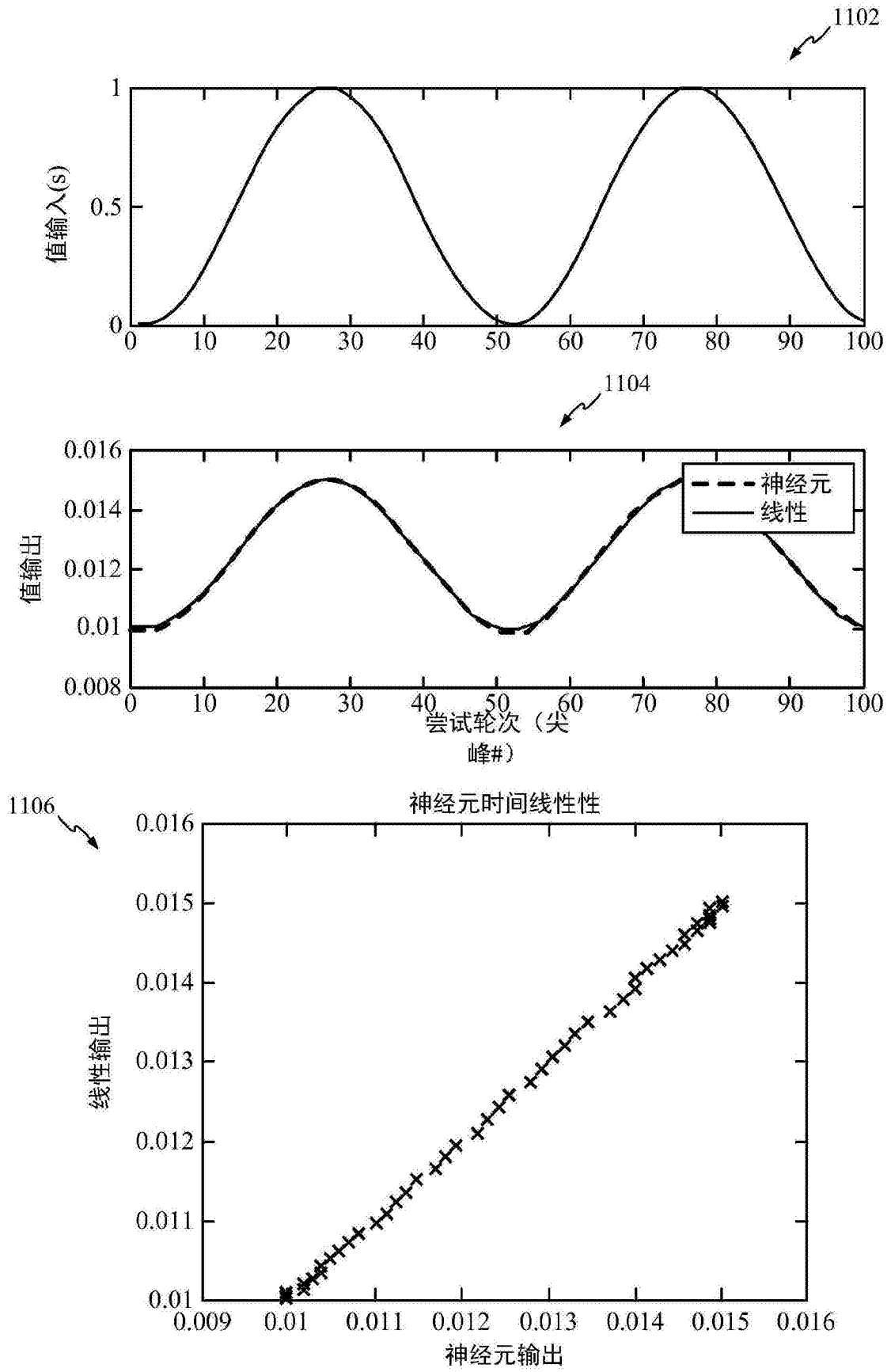


图11A

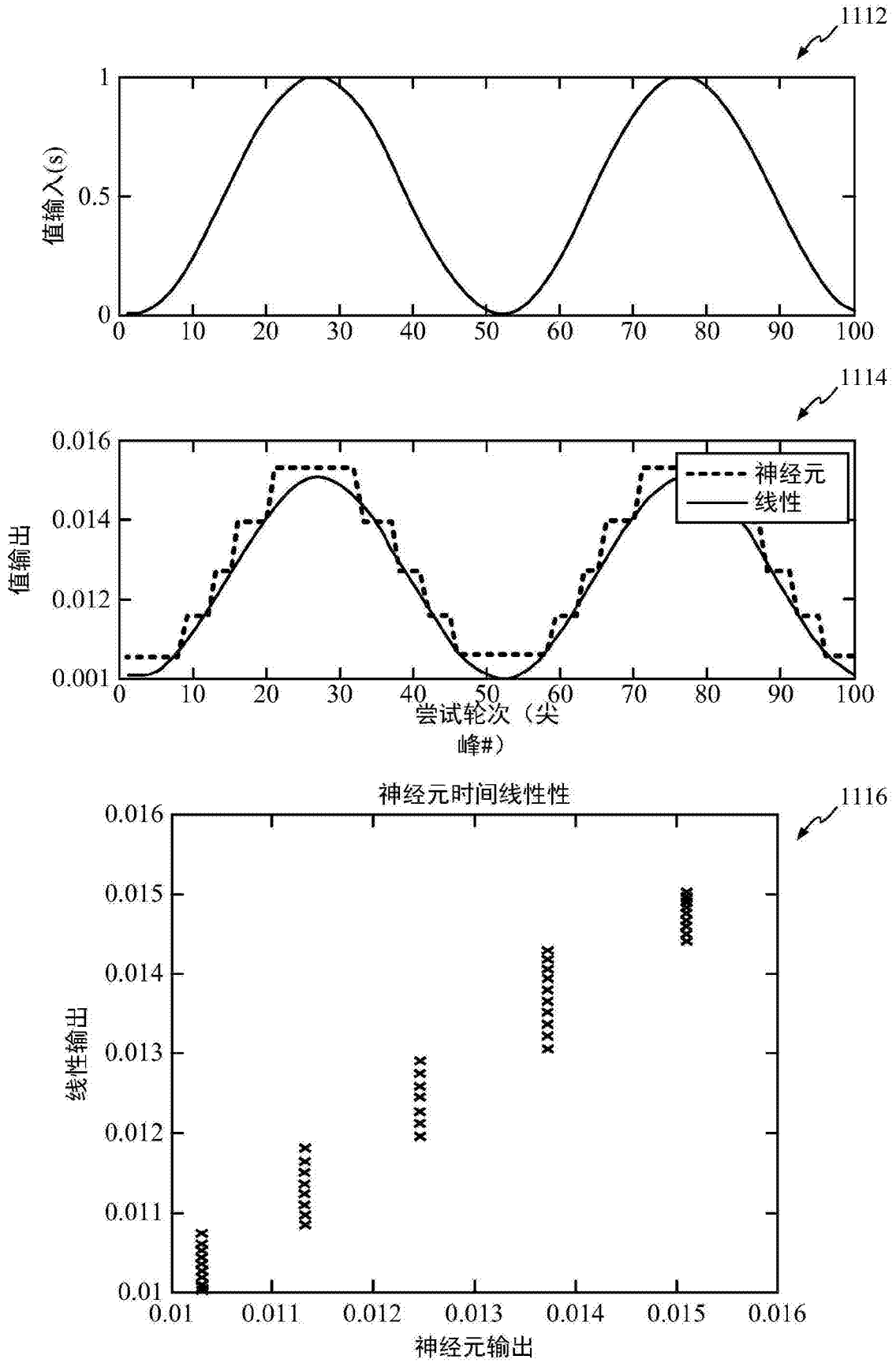


图11B

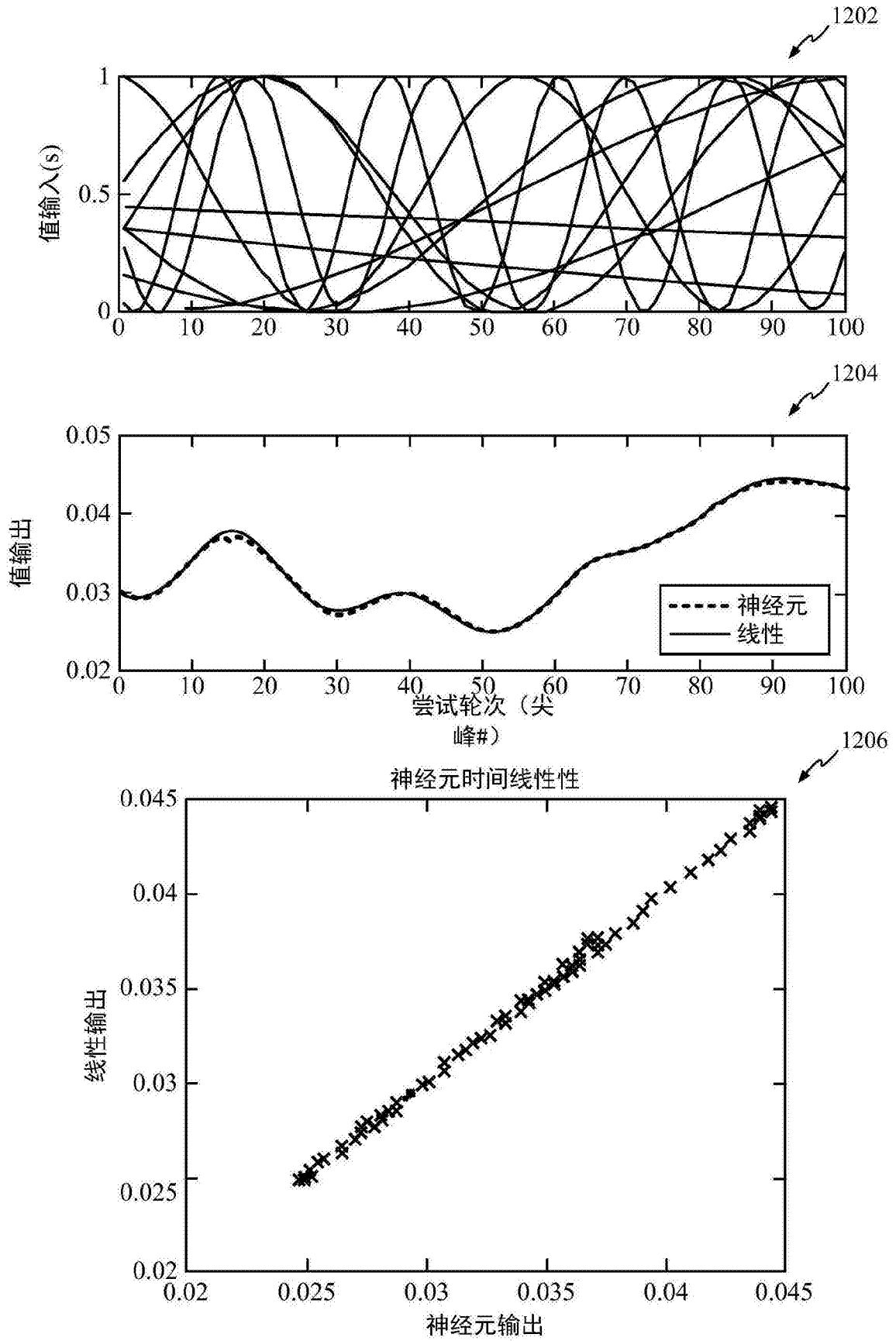


图12A

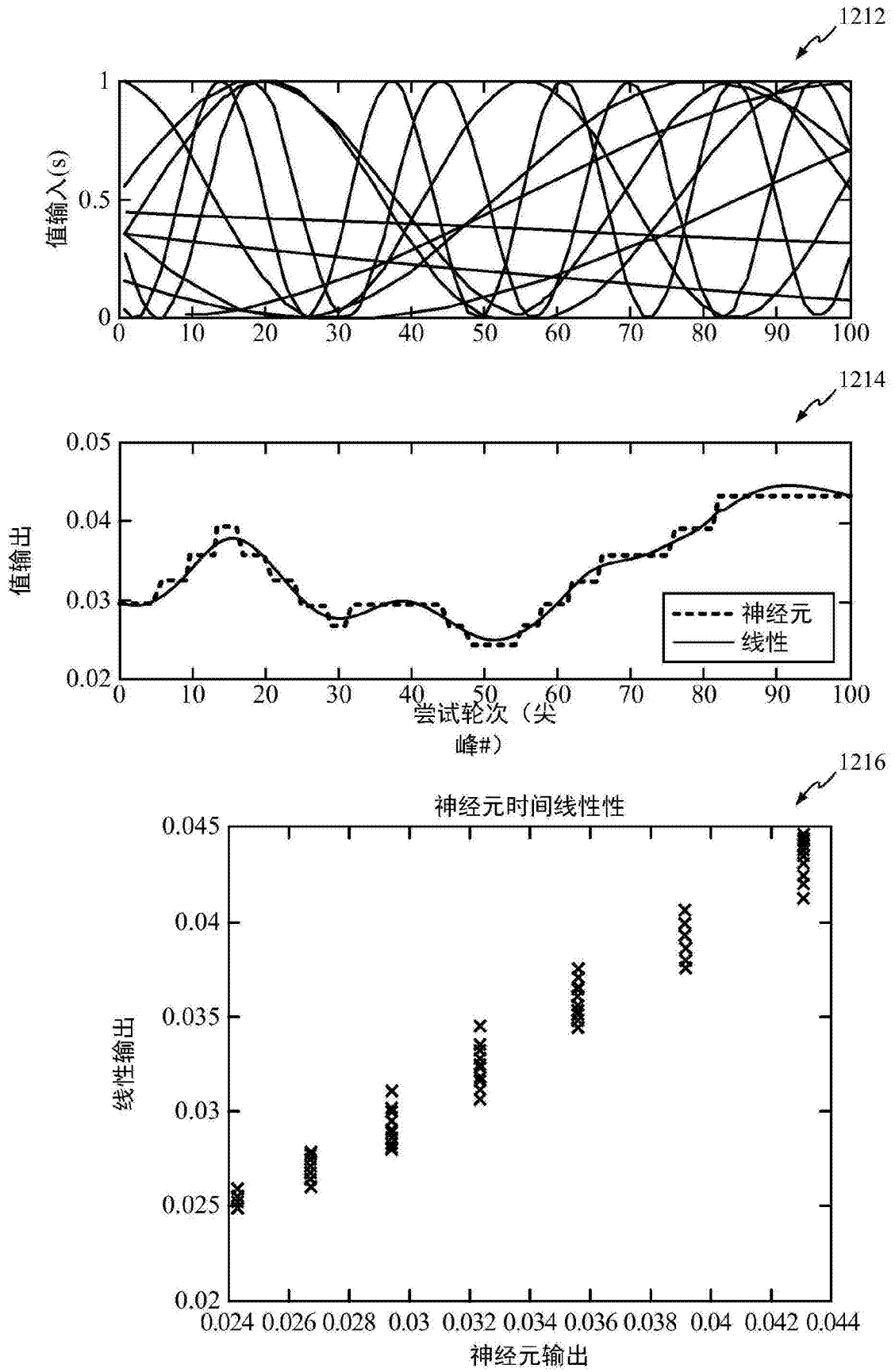


图12B

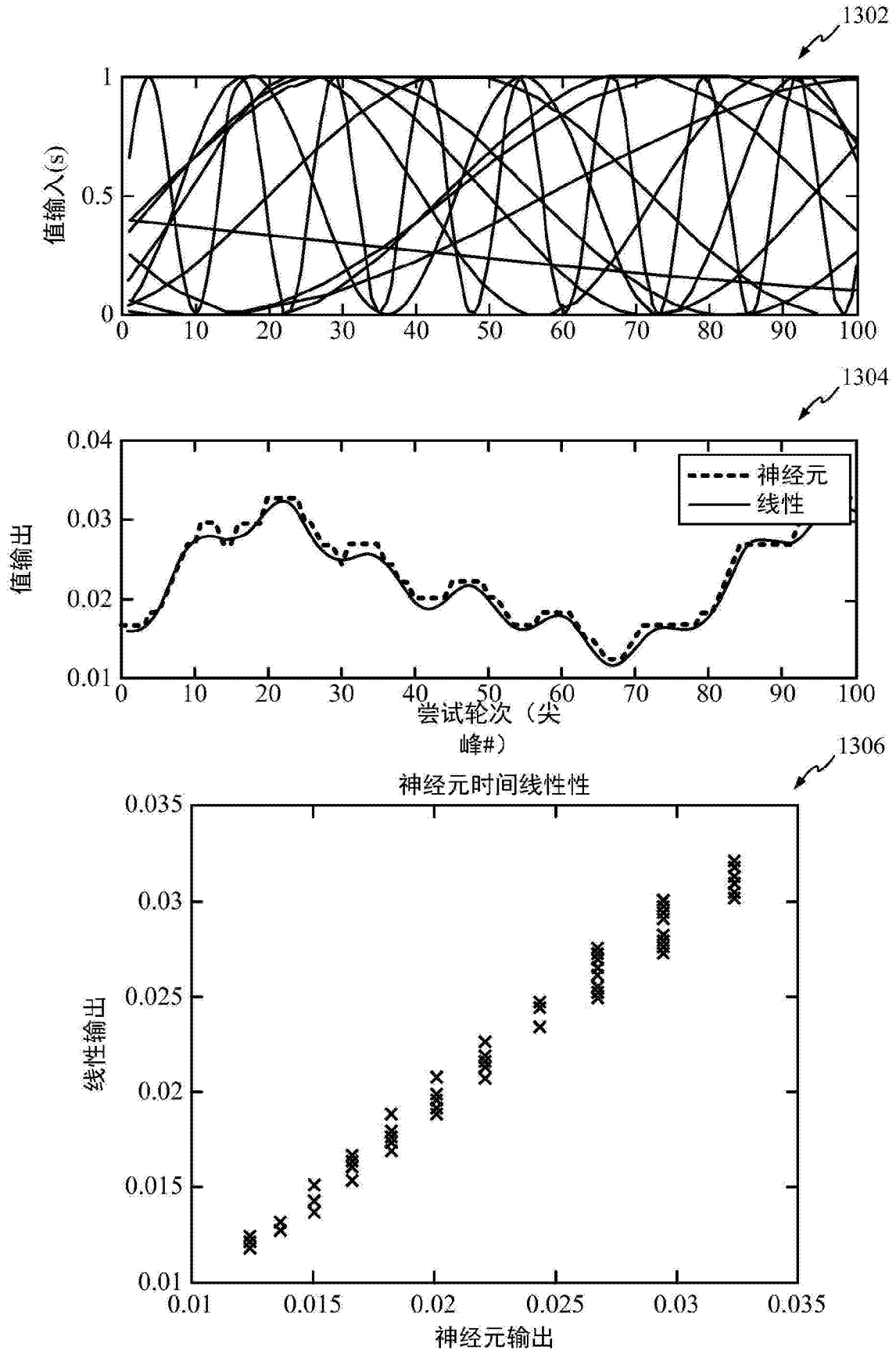


图13A

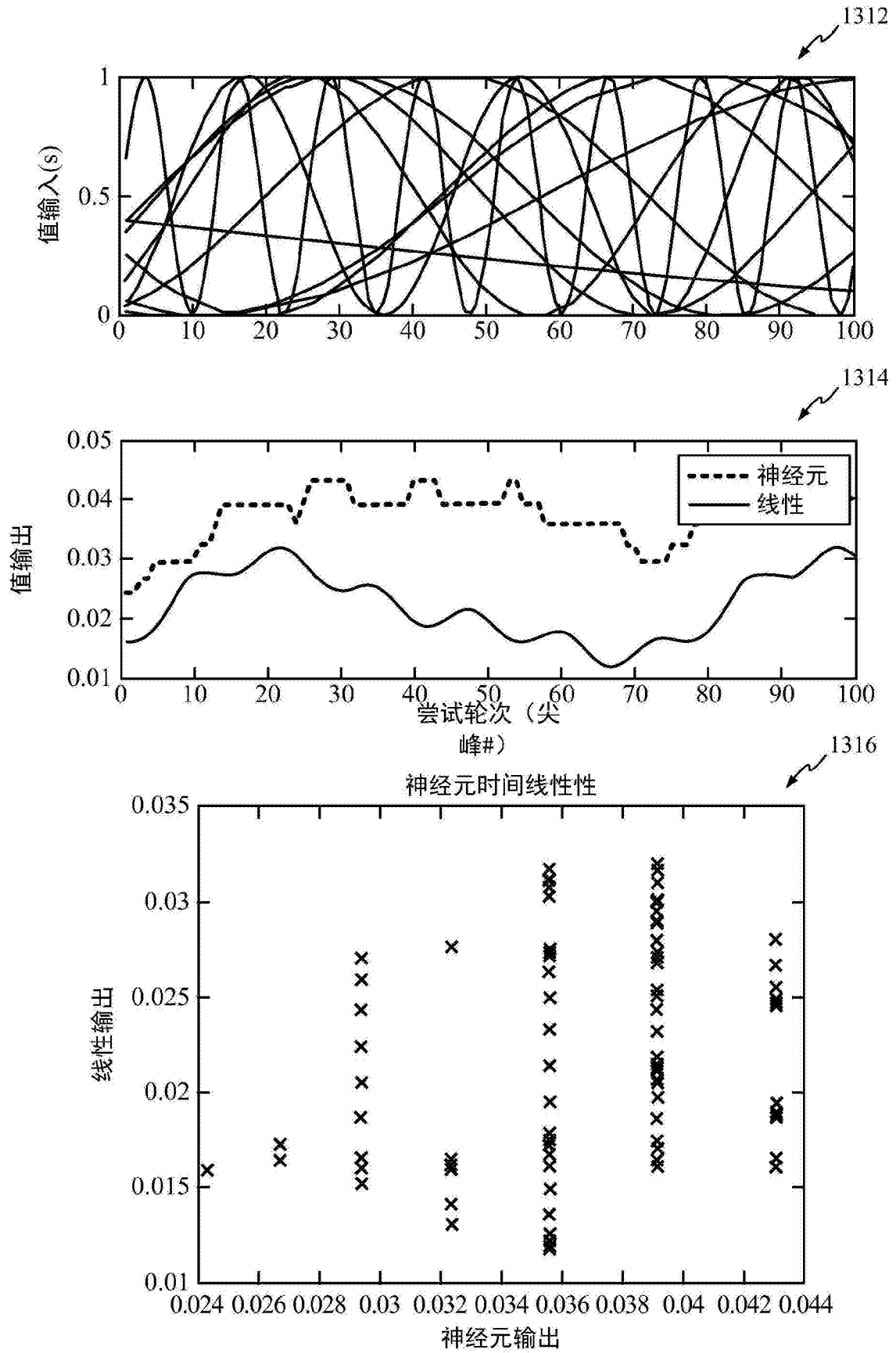


图13B

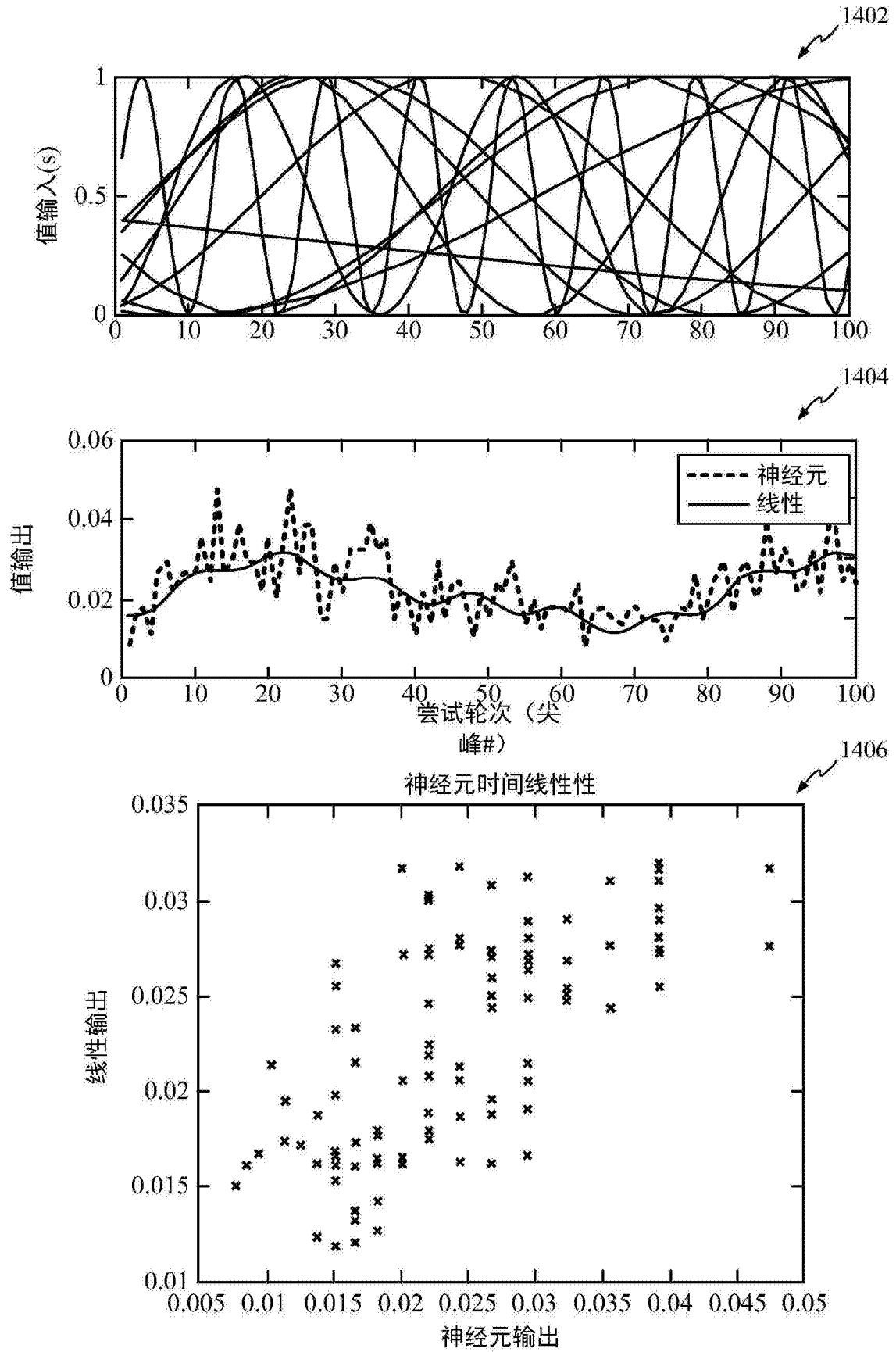


图14

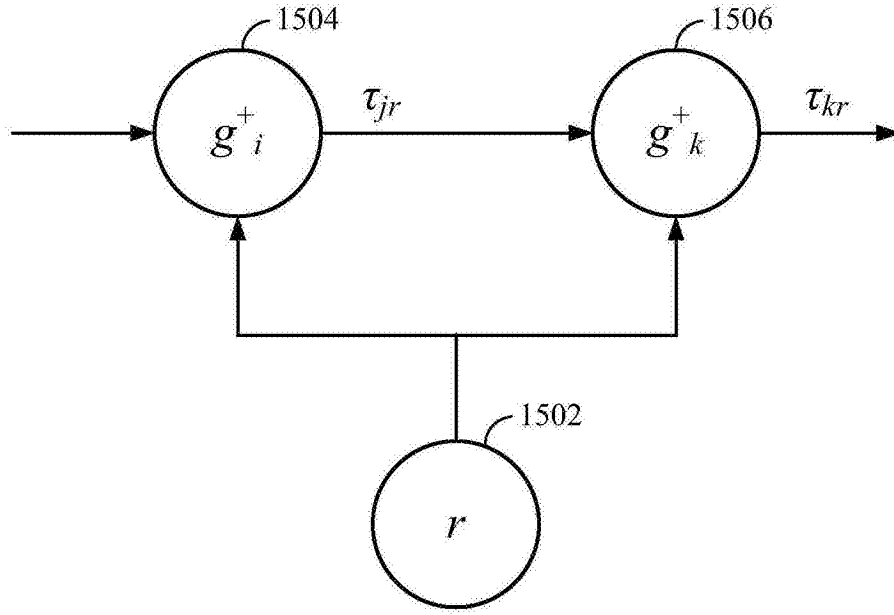


图15

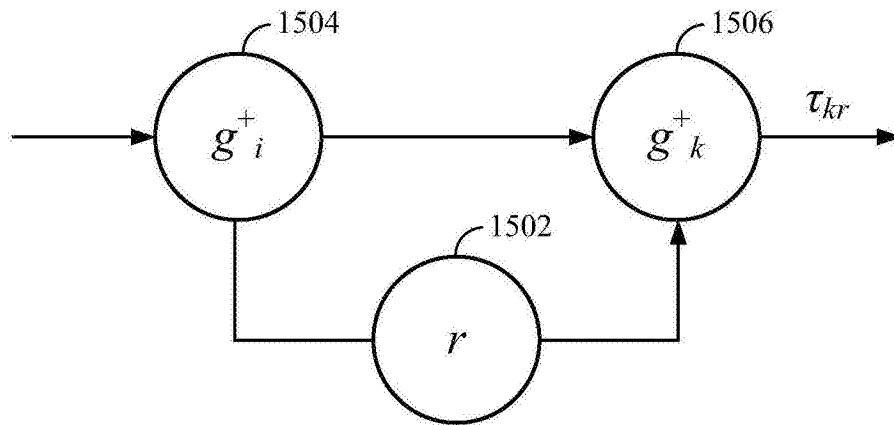


图16

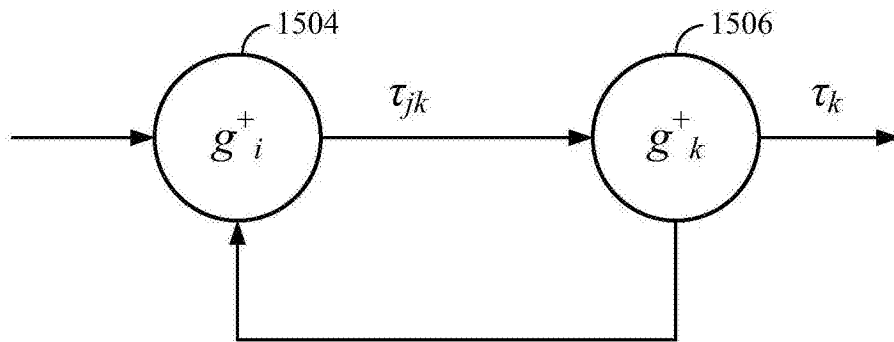


图17

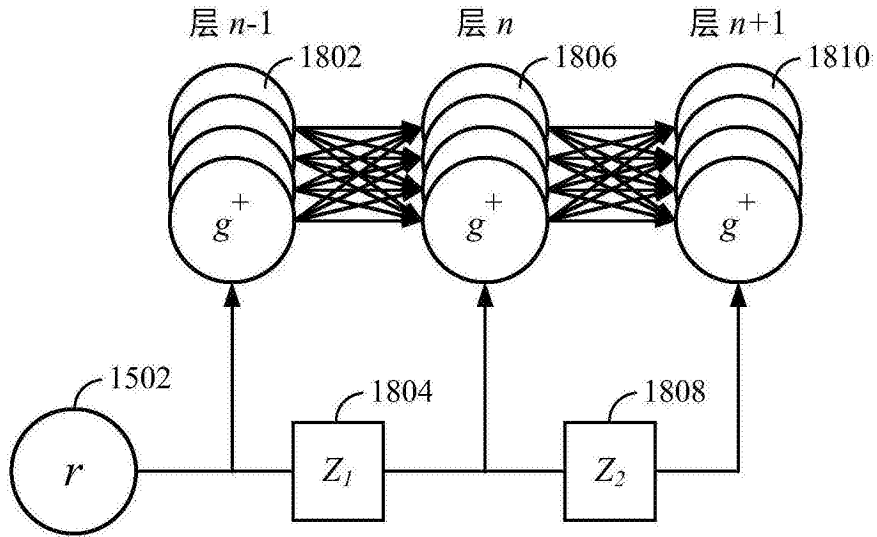


图18

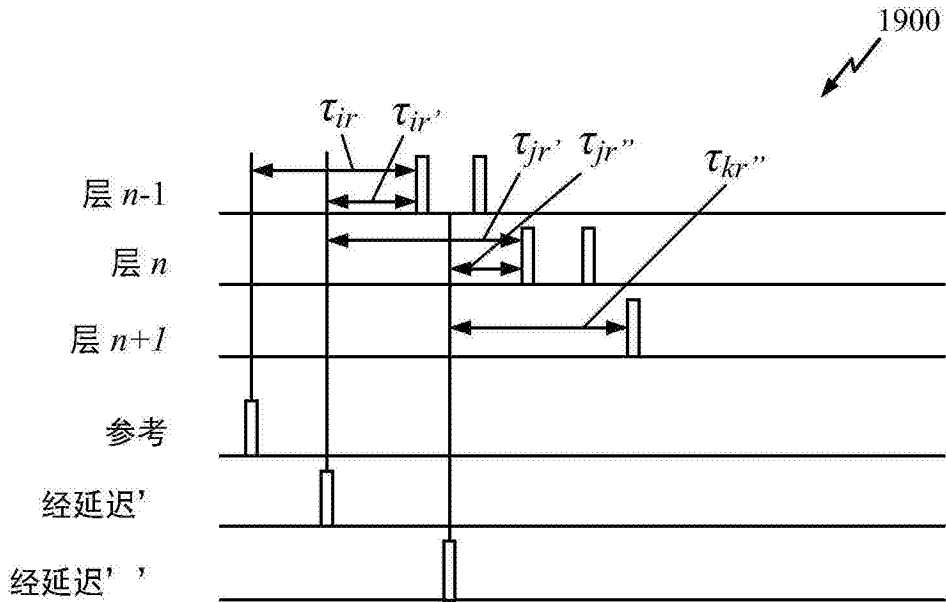


图19

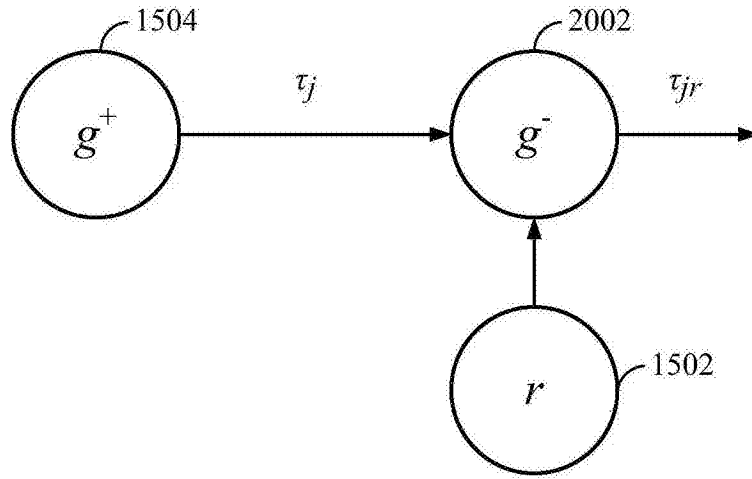


图20

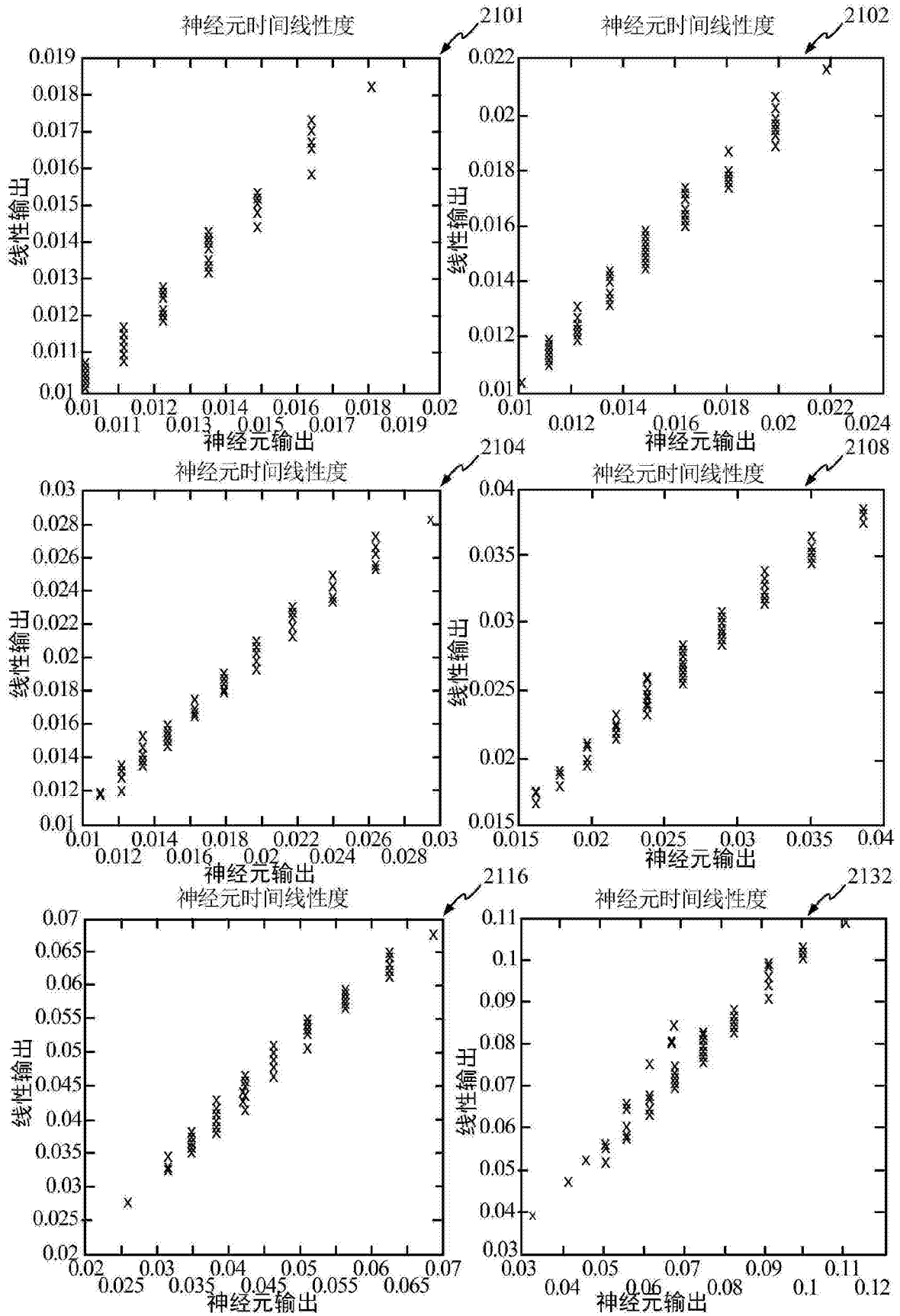


图21

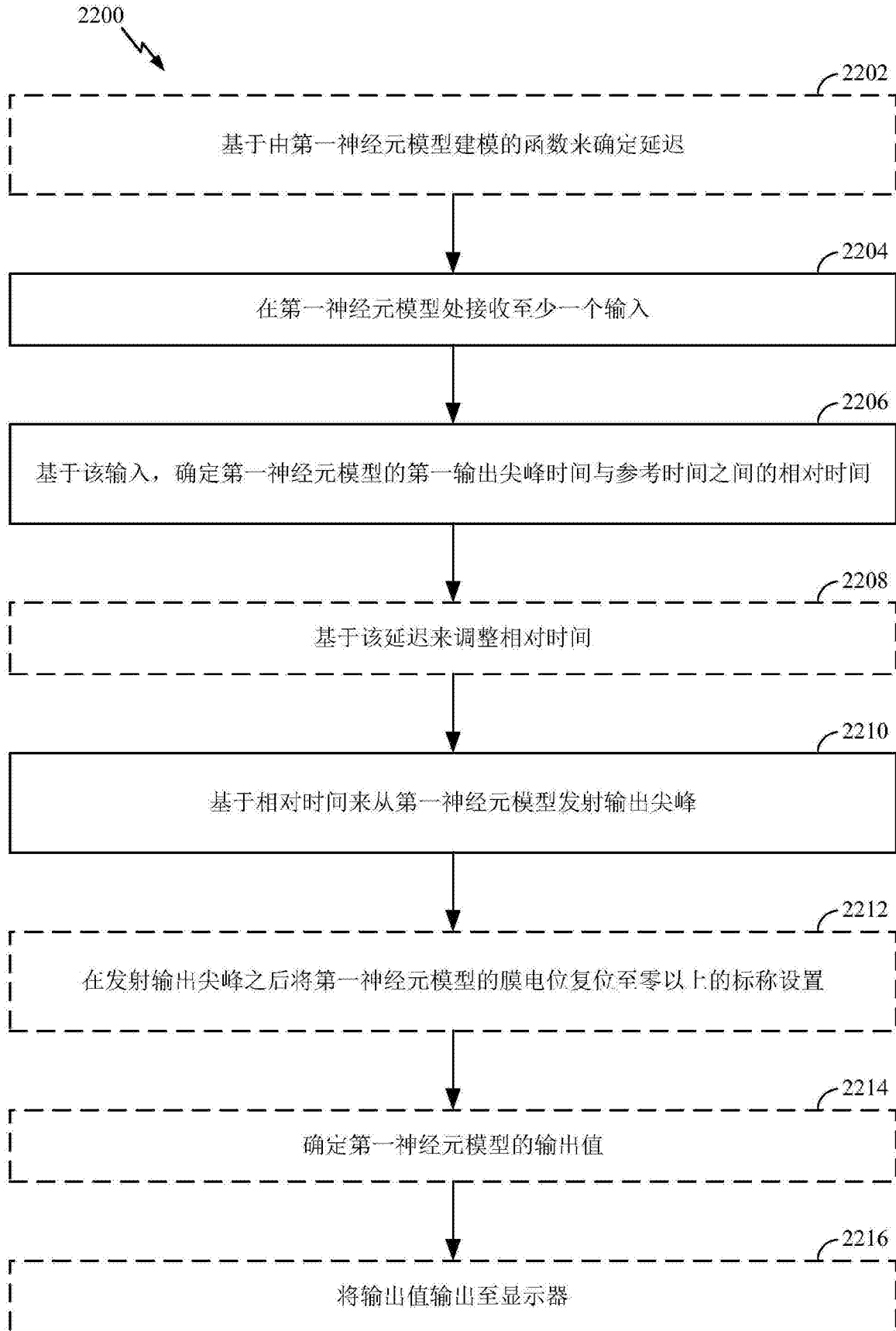


图22

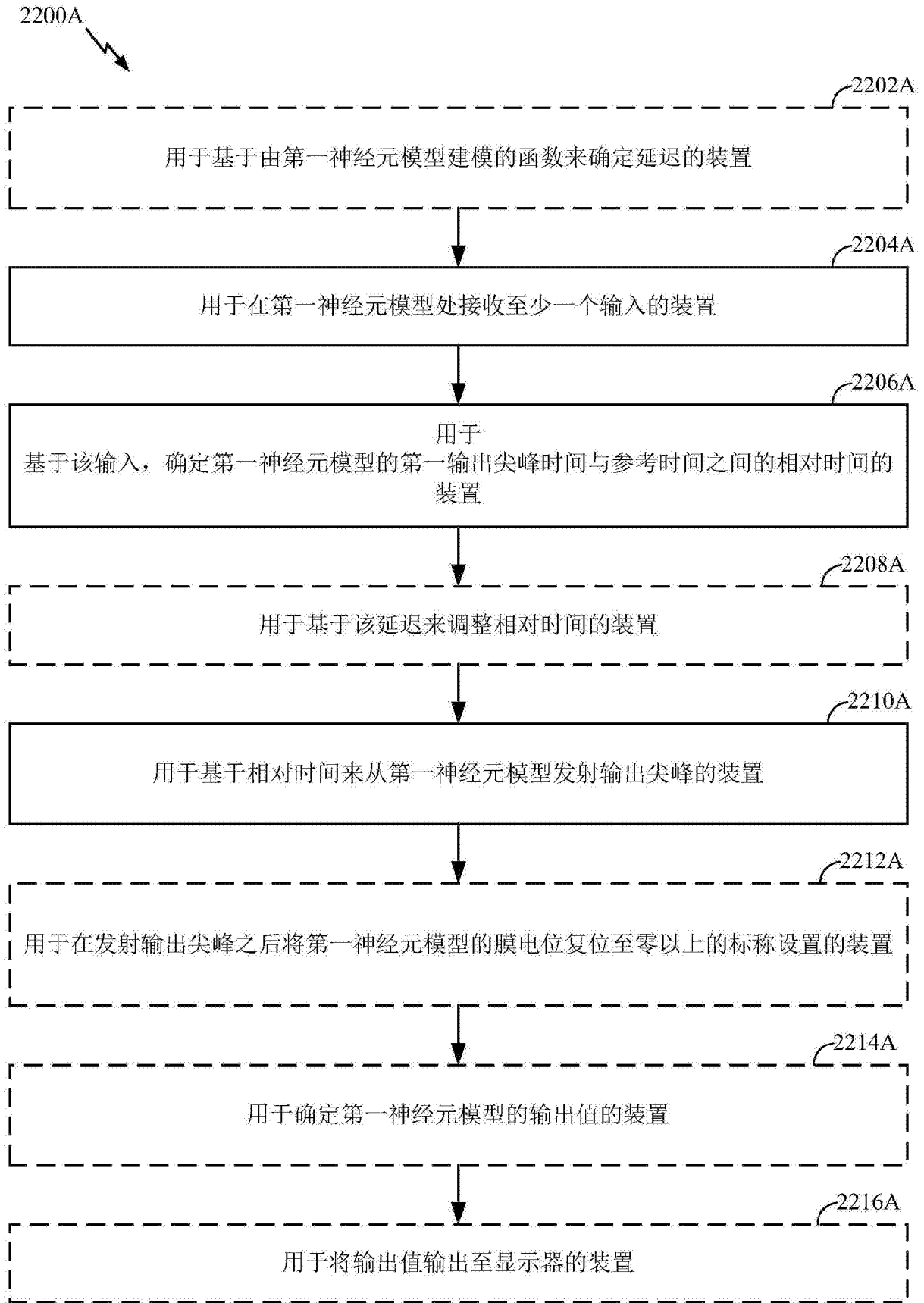


图22A

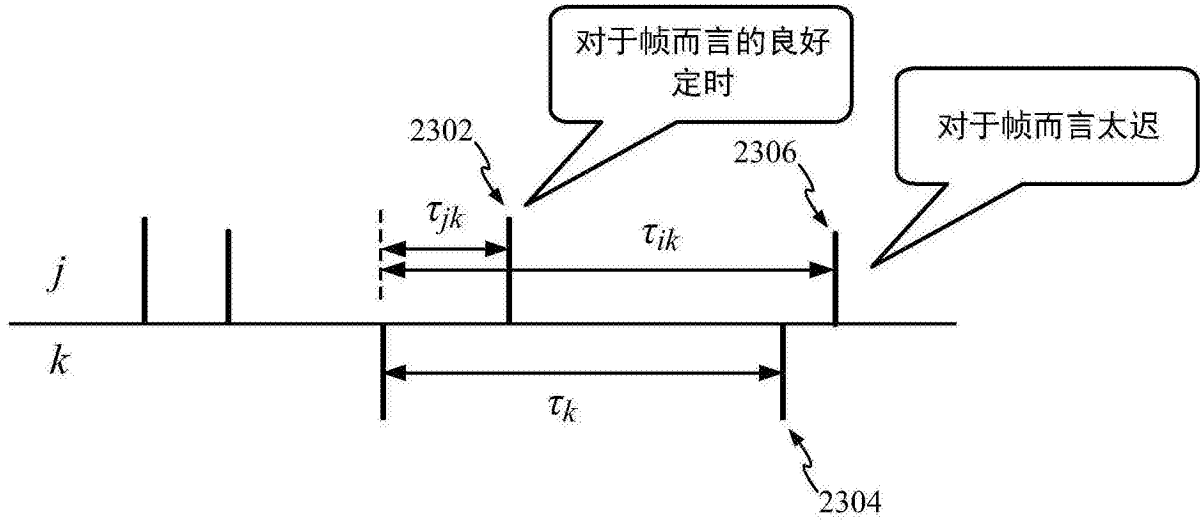


图23

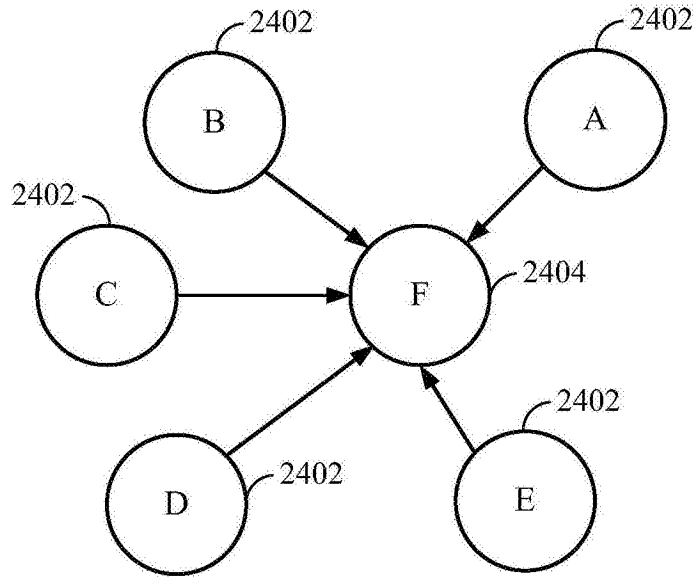


图24

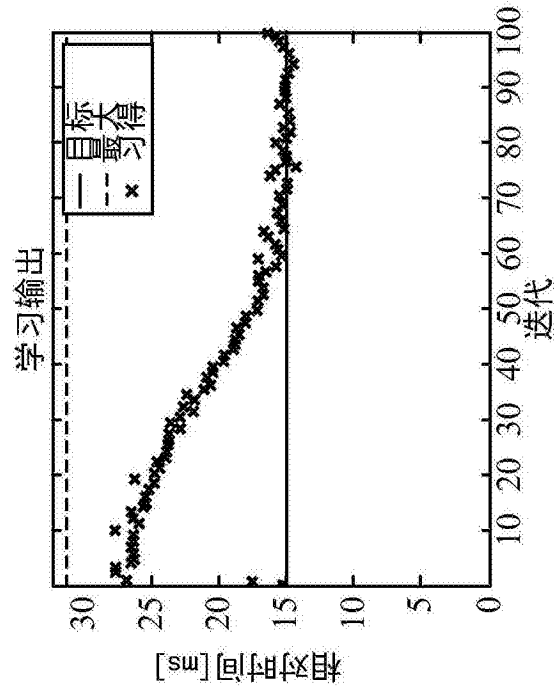


图25A

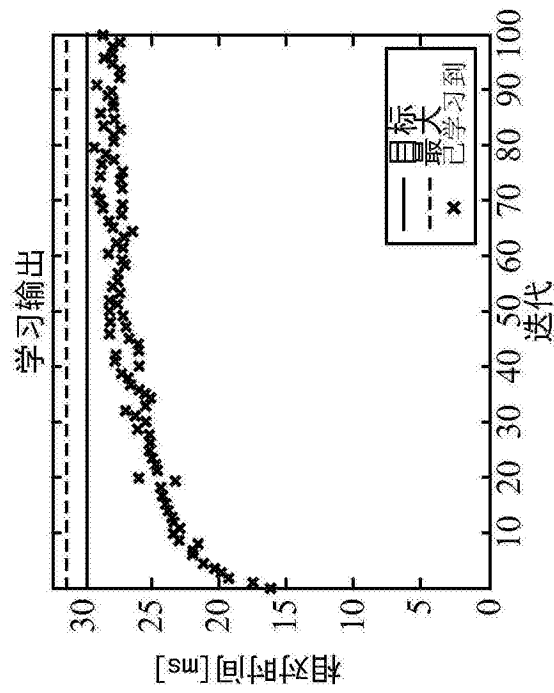


图25B

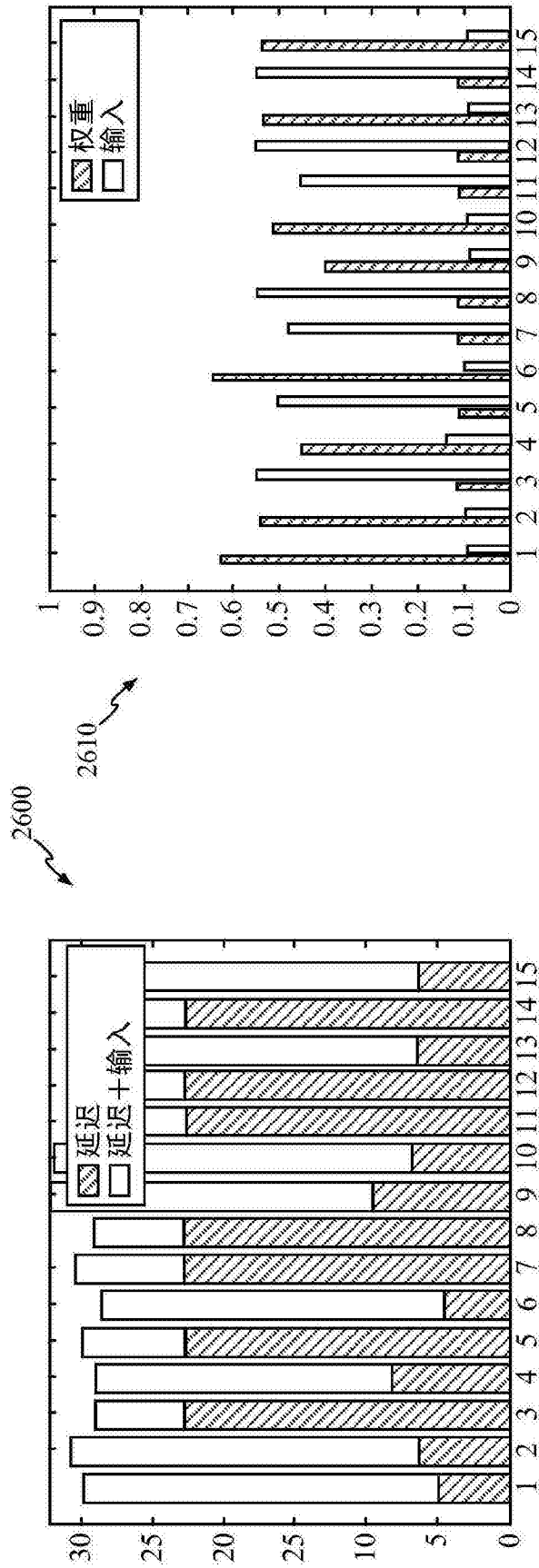


图26

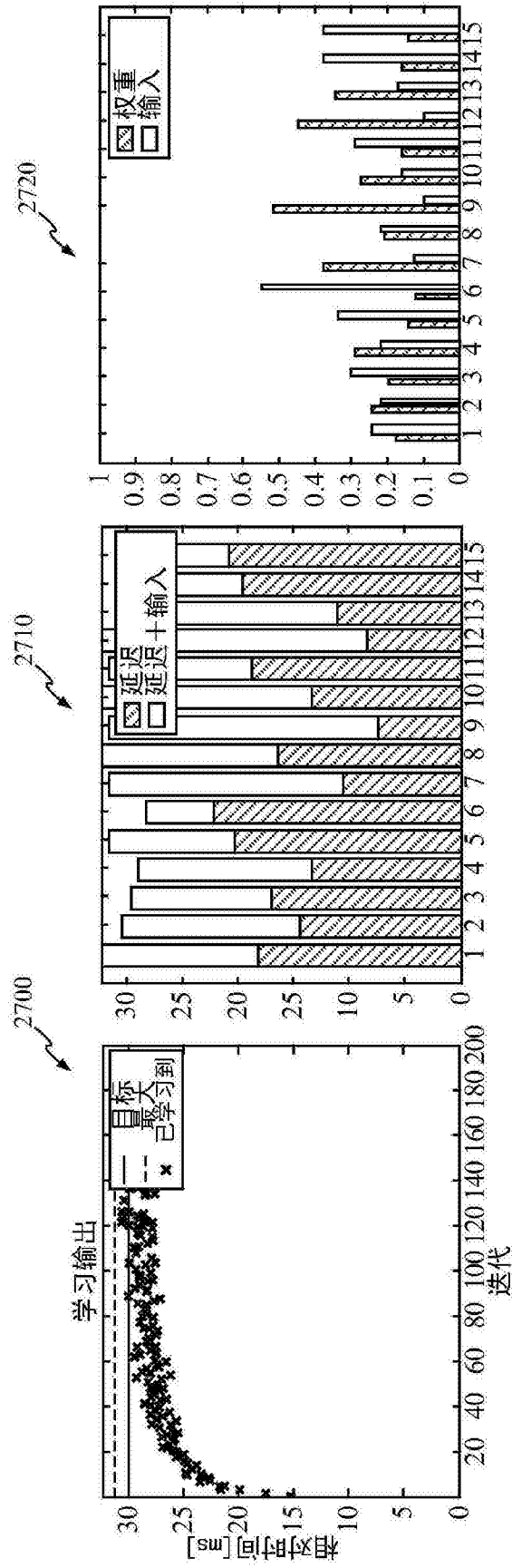
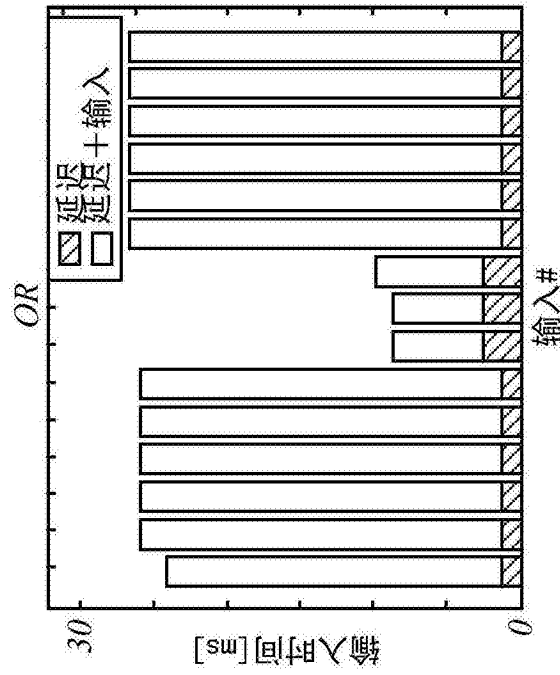


图27



2800 ↗

图28A

2810 ↗

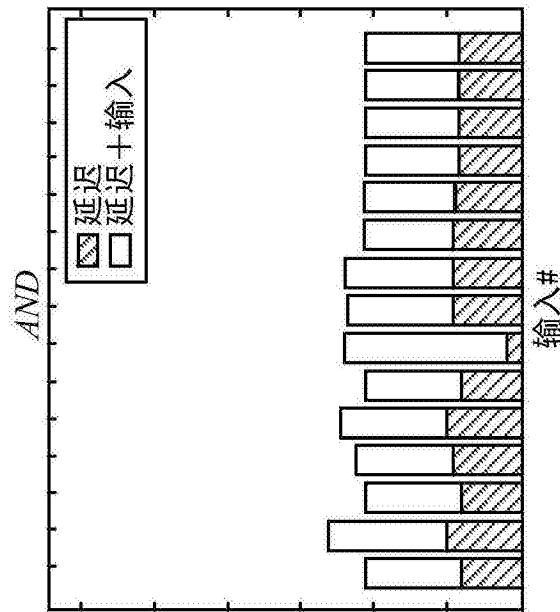


图28B

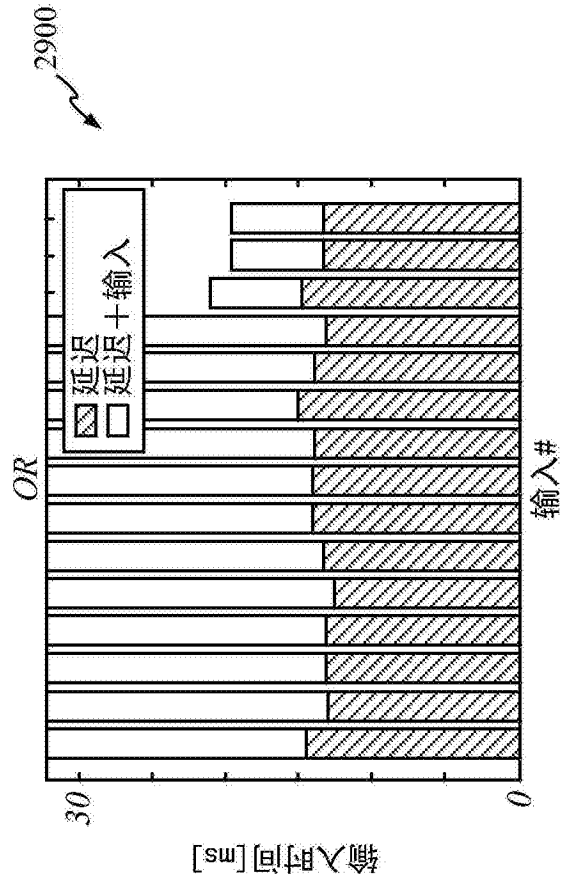


图29A

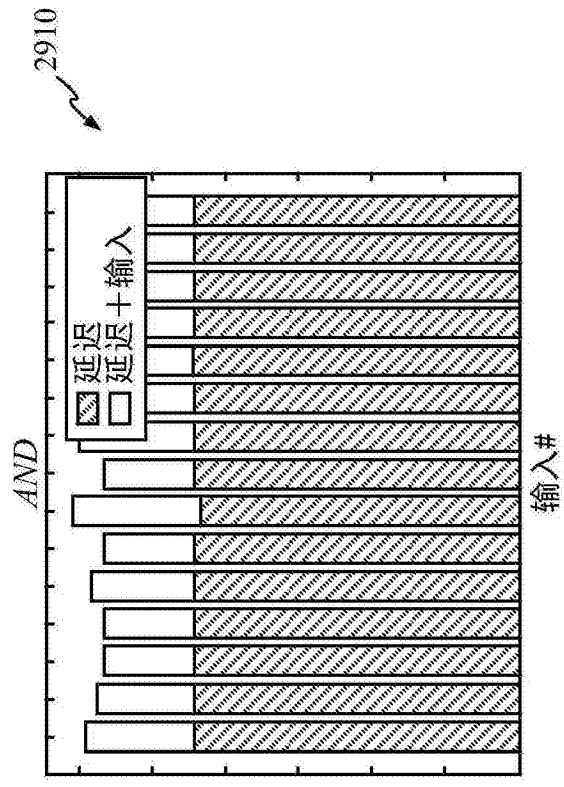


图29B

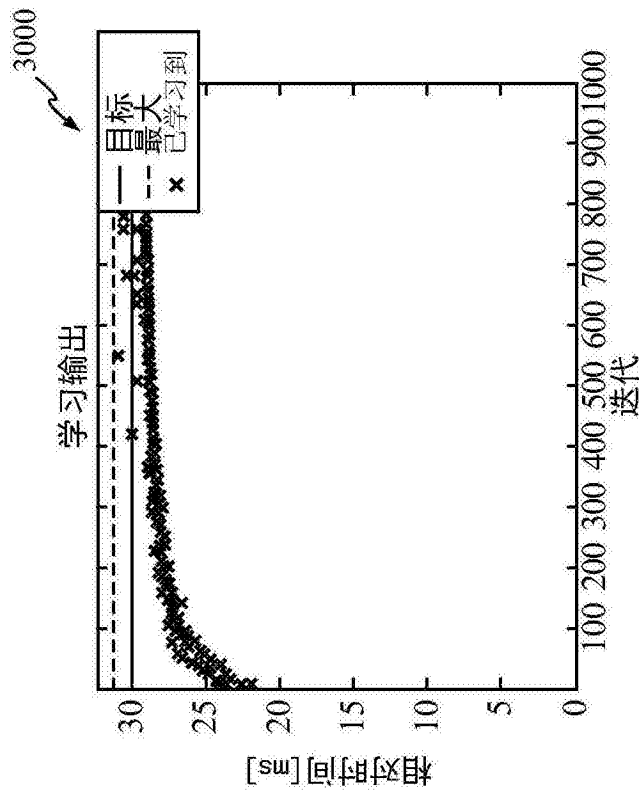
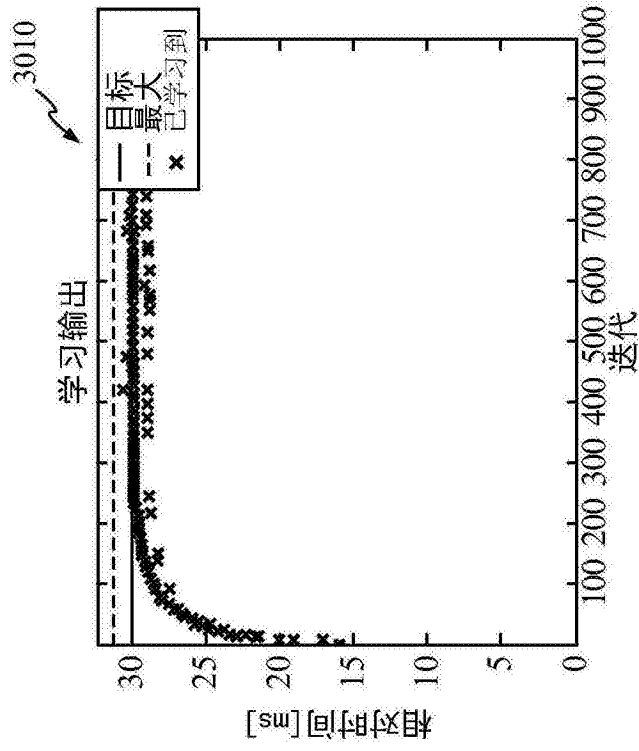


图30

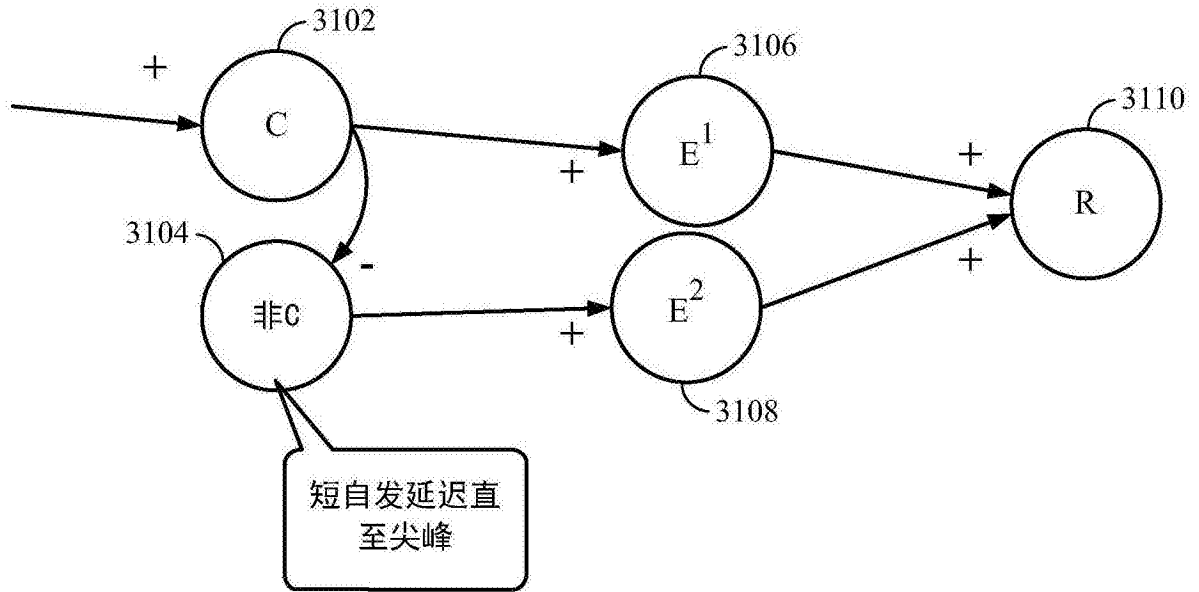


图31

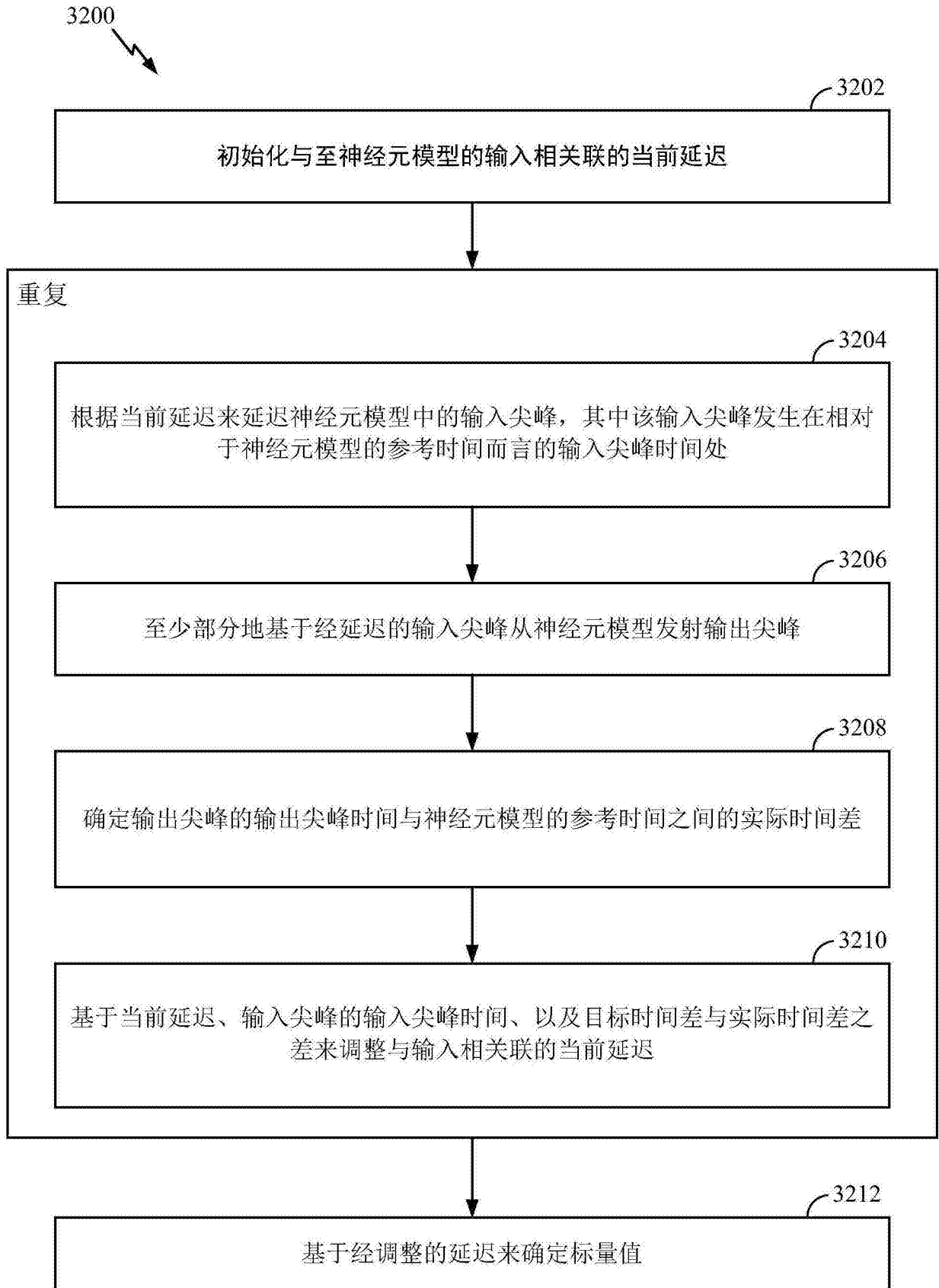


图32

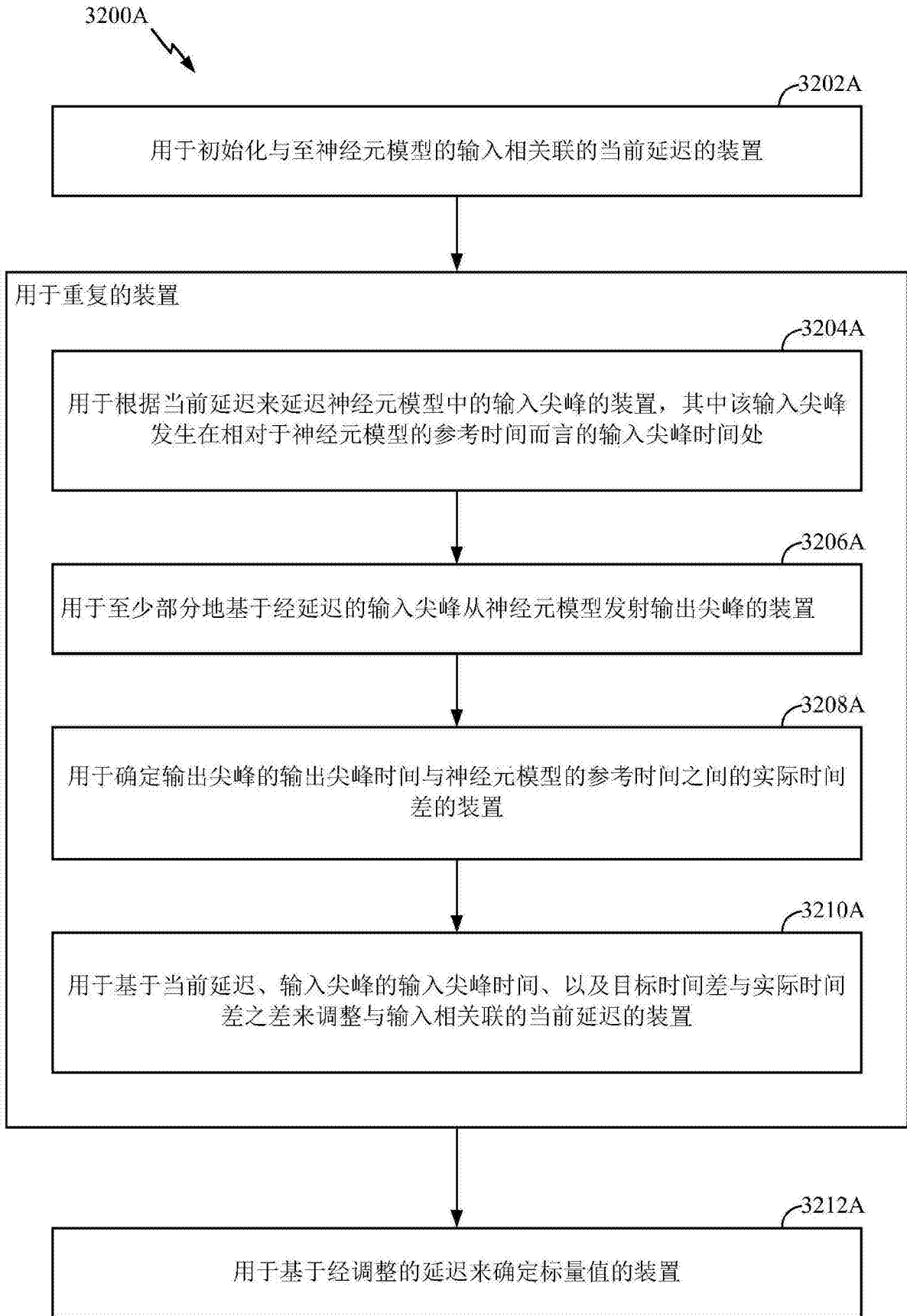


图32A

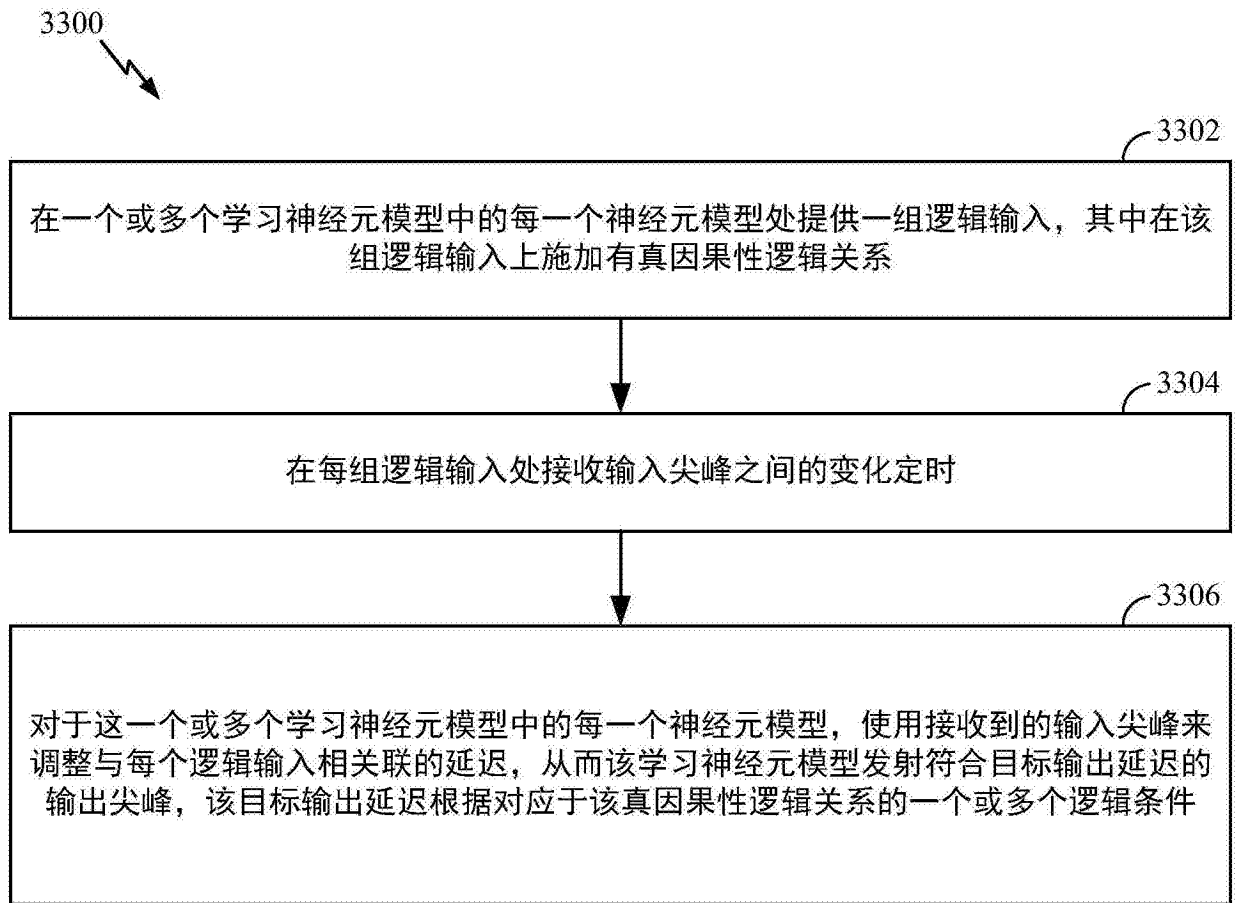


图33

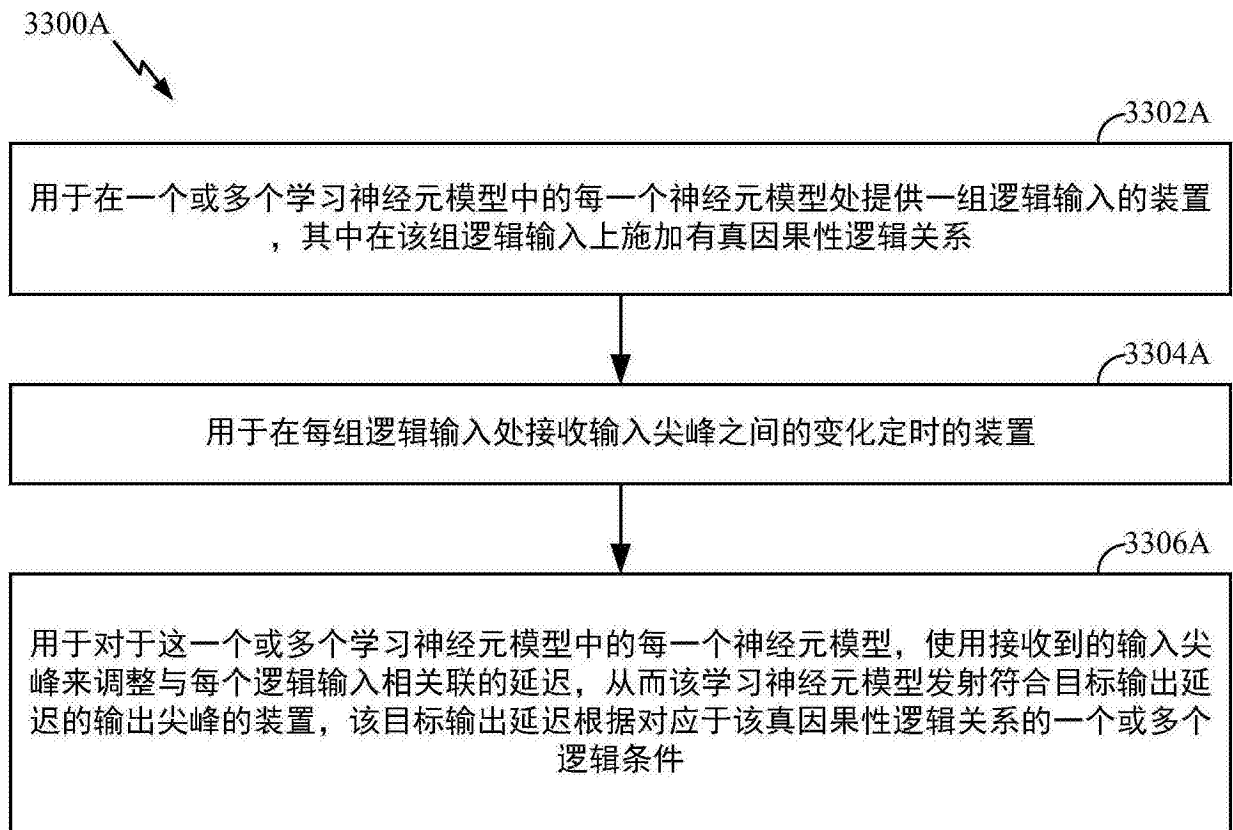


图33A