

神经采样

萧子豪 朱 军

(清华大学计算机科学与技术系 北京 100084)

(清华大学智能技术与系统国家重点实验室 北京 100084)

(清华大学类脑计算研究中心 北京 100084)

摘 要 神经采样是国际上最近提出的一种基于脉冲神经网络动力学的吉布斯采样算法,是一种有希望在类脑硬件上实现贝叶斯概率推断的算法.神经采样的仿生特点包括考虑神经元间通过发放脉冲来传递信息、突触后膜电压和迟滞效应等.该文首先会介绍国际上在神经采样方面已有的工作,分三部分:第1部分涉及神经采样的抽象模型,包括其原理和在任意贝叶斯网络中采样的具体模型;第2部分涉及硬件实现,包括用累积发放(LIF)模型近似连续时间神经采样动力学的方法;第3部分通过结合前两部分,涉及用脉冲神经网络动力学训练传统的机器学习模型,并在经典计算机上模拟这个训练过程.第3部分具体包括基于脉冲时间的突触可塑性(STDP)的受限玻尔兹曼机(RBM)的事件驱动相对散度训练算法.最后,我们在训练RBM的相对散度和持续相对散度算法中,用神经采样替代传统的吉布斯采样.该工作首先分析了神经采样对初始化状态敏感和混合速度慢的采样特点,然后提出方法消除了这两个采样特点对训练的负面影响.在MNIST数据集上的实验初步显示,基于修正后的神经采样的训练算法能恢复跟传统基于吉布斯采样的算法相似的重构效果.目前在机器学习领域,基于概率的学习算法已发展成主流.神经采样方面的工作为在类脑硬件上实现低能耗的概率模型计算提供方法,未来有希望被用于提高移动设备的智能水平.

关键词 脉冲神经网络;神经采样;马尔可夫链蒙特卡洛;受限玻尔兹曼机;类脑硬件;人工智能

中图法分类号 TP18 DOI号 10.11897/SP.J.1016.2017.01379

Neural Sampling

XIAO Zi-Hao ZHU Jun

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

(State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing 100084)

(Center for Brain-Inspired Computing Research, Tsinghua University, Beijing 100084)

Abstract Neural sampling is a recently proposed Gibbs sampling algorithm implemented by spiking neural network dynamics, and is hopeful to be used as an algorithm to do Bayesian inference on neuromorphic platforms. The biological plausibility of neural sampling is on that neurons use spikes to transmit information, post-synaptic potential and refractory effect, etc. This paper first introduces some existing works on neural sampling in three sub-parts. The first sub-part is about the abstract models of neural sampling, including its theory and a specific model applied to sample from arbitrary Bayesian networks. The second sub-part is about hardware implementations, including using a leaky integrate & fire(LIF) model to approximate continuous time neural sampling dynamics. The third sub-part is about using spiking neural network dynamics to train a regular machine learning model, and the corresponding simulation on sequential

computers, by combining the materials mentioned in the first two sub-parts. The third sub-part includes an event-driven contrastive divergence (CD) to train restricted Boltzmann machines (RBMs) based on spike-timing dependent plasticity (STDP). Finally, our work is to use neural sampling to replace the Gibbs sampling in the regular CD and persistent CD (PCD) algorithm to train RBMs. The authors first analyze that neural sampling is sensitive to initialization and has slow mixing rate. And then we propose methods to fix these two disadvantages. Our preliminary experiments on MNIST dataset show that our training algorithms based on the revised neural sampling have similar reconstruction performance with those based on regular Gibbs sampling. Today, probabilistic methods have become a mainstream in machine learning. The emerging works on neural sampling provide approaches for implementing low-energy probabilistic computations on neuromorphic platforms. This is hopeful to enrich the intelligence on mobile devices in the near future.

Keywords spiking neural network; neural sampling; Markov Chain Monte Carlo; restricted Boltzmann machine; neuromorphic hardware; artificial intelligence

1 引 言

由于真实世界的复杂性和不确定性,人工智能系统需要有效进行不确定性推理.其中,以概率统计为基础的人工智能方法已发展为主流方向^[1].

为了实现高效准确的概率推理,学者们深入研究了变分推理方法^[2]和马尔可夫链蒙特卡洛(Markov Chain Monte Carlo, MCMC)随机采样算法^[3,4].相比于变分方法需要对问题进行简化近似,MCMC方法可以在计算时间足够长时保证收敛到目标分布,因此具有更好的推理精度.本文将主要关注 MCMC 方法,其中,特别关注近期发展的基于神经采样(neural sampling)的算法.

许多经典的 MCMC 采样算法所构造的马尔可夫链是可逆的,如 Metropolis-Hastings 算法和吉布斯采样.但生理实验观察到脉冲神经网络的活动是时序不可逆过程.比如神经元在发放完脉冲后的一段时间里,再发放脉冲的概率会降低,这种现象称为迟滞现象(refractoriness).神经元传递出去的脉冲被接收以后,接收到脉冲的神经元或树突的膜电压会被持续改变一段时间,这称为突触后膜效应(post-synaptic effect).神经采样用不可逆马尔可夫过程描述这些现象,解决了用脉冲神经网络动力学来做 MCMC 随机采样的问题.神经采样算法既有严格的理论证明马尔可夫链会收敛于目标分布^[5],同时,目前已实现在任意贝叶斯网络下采样^[6].文献

[7]用带噪声的 I&F(Integrate&Fire 的缩写)脉冲神经网络模型来近似神经采样,从而有希望在类脑硬件上实现神经采样算法.

类脑硬件受神经系统用脉冲神经网络计算的启发,采用事件驱动(event-driven)的方式传递信息,优点在于比传统的计算机体系消耗更少能量^[8].目前大规模的机器学习算法在传统计算机上训练和计算往往功耗很大,限制了机器学习在移动设备上的应用.我们的目标是在类脑硬件上实现大规模基于概率推理的机器学习算法的推理和训练,并将这样的系统应用于移动设备,而神经采样就是一个有希望在类脑硬件上实现概率推理的算法.除了能用神经采样来做推理,文献[7]提出一个结合 STDP(Spike-Timing-Dependent Plasticity)和神经采样的算法来训练受限玻尔兹曼机(Restricted Boltzmann Machine, RBM),称为事件驱动的相对散度算法(event-driven contrastive divergence).这个算法不借助经典计算机计算,是可以完全依赖脉冲神经网络活动来训练机器的学习模型,是一种可能在类脑硬件上低能耗地训练深度学习网络的方法.

本文会先分节介绍上面提到的神经采样的最近进展,最后介绍我们的工作:通过用神经采样代替相对散度(Contrastive Divergence, CD^[9])和持续相对散度(Persistent Contrastive Divergence, PCD^[10,11])中的吉布斯采样,在理论上分析和实验上展示神经采样的样本在状态空间中的转移特点.

本文第 2 节简单介绍 MCMC 随机采样,并以

RBM 的训练算法为例介绍 MCMC 方法在机器学习问题中的应用;第 3 节介绍离散时间的神经采样算法,论述它契合了生理实验所观察到脉冲神经网络活动的时序不可逆性,并且采样收敛到目标分布;第 4 节将离散时间的神经采样中的网络状态更新法则推广成连续时间的动力学;第 5 节介绍用神经采样对任意贝叶斯网络做随机推理的方法;第 6 节介绍如何用随机 I&F 模型近似连续时间的神经采样,这为用目前的类脑硬件来实现神经采样提供了可能;第 7 节介绍事件驱动的 CD 算法,这是一种只依赖脉冲神经网络动力学来训练机器学习模型的算法;第 8 节我们以 CD 和 PCD 为例理论分析和实验展示神经采样的采样特点,具体包括采样过程对初始化条件敏感和混合速度低;第 9 节总结本文的内容及提出待解决的问题.

2 MCMC 方法及其在 RBM 训练算法中的应用

马尔可夫链蒙特卡洛方法(MCMC)是一种采样算法. 设状态空间为 S , 马尔可夫转移算符 T 定义了从任意一个态 $s \in S$ 到任意一个态 $s' \in S$ 的转移概率:

$$T(s' | s) = p(s' | s) \quad (1)$$

如果转移算符 T 存在稳态分布 p , 即

$$\sum_s T(s' | s) p(s) = p(s') \quad (2)$$

且转移算符 T 是非退化和非周期的, 那么从任意概率分布抽取的起始样本, 通过马尔可夫算符 T 不断地更新样本状态, 经过足够长的计算时间这些样本会转化为稳态分布 p 的样本. 这种采样方法称为 MCMC 方法.

以概率推理为基础的人工智能模型广泛使用 MCMC 采样方法, 训练受限玻尔兹曼机(RBM^[12]) 是一个典型例子. RBM 是由可见层和隐层组成的两层神经网络, 只有不同层的神经元间有相互连接, 同层神经元间没有连接, 每个神经元是 0/1 二值单元. RBM 表示的模型分布是

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp\left(\sum_{i,j} w_{ij} v_i h_j\right) \quad (3)$$

其中 \mathbf{v} 和 \mathbf{h} 分别表示可见层和隐层的状态; v_i 和 h_j 表示单个可见或隐单元的状态; w_{ij} 称为相互作用强度; Z 是归一化系数. 为了简化描述, 我们省略了偏置(bias). RBM 的特点是在观察到可见层状态后隐

单元之间相互独立:

$$p(h_j = 1 | \mathbf{v}, \mathbf{h}_{\setminus j}) = \sigma\left(\sum_i w_{ij} v_i\right) \quad (4)$$

$$p(h_j = 0 | \mathbf{v}, \mathbf{h}_{\setminus j}) = 1 - \sigma\left(\sum_i w_{ij} v_i\right) \quad (5)$$

$j=1, \dots$, 其中 $\mathbf{h}_{\setminus j}$ 表示隐层中除了第 j 个隐单元外其它隐单元的状态, $\sigma(x) = (1 + e^{-x})^{-1}$ 称为 Sigmoid 函数. 已知隐层状态后可见单元之间也相互独立:

$$p(v_i = 1 | \mathbf{h}, \mathbf{v}_{\setminus i}) = \sigma\left(\sum_j w_{ij} h_j\right) \quad (6)$$

$$p(v_i = 0 | \mathbf{h}, \mathbf{v}_{\setminus i}) = 1 - \sigma\left(\sum_j w_{ij} h_j\right) \quad (7)$$

$i=1, \dots$, 其中 $\mathbf{v}_{\setminus i}$ 表示可见层中除了第 i 个可见单元外其它可见单元的状态. 在将可见层状态固定为训练数据后, 可并行地对每个隐单元做吉布斯采样^[13]; 在得到隐层样本后可以反过来再对每个可见单元做吉布斯采样:

$$\text{采样 } h_j \sim p(h_j | \mathbf{v}), j = 1, \dots,$$

$$\text{采样 } v_i \sim p(v_i | \mathbf{h}), i = 1, \dots,$$

$z \sim p$ 表示变量 z 以概率分布 p 采样. 这样在可见层和隐层之间来回做一轮采样推理称为一次重构(reconstruction).

在早期的 RBM 训练算法中, 待优化的目标函数是未知的环境分布 $q(\mathbf{v})$ 和模型表示的环境 $p(\mathbf{v}) = \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h})$ 分布之间的 KL 散度^[14]. 用梯度下降法训练时, 目标函数对参数的偏导数为

$$\frac{\partial \text{KL}(q(\mathbf{v}) \| p(\mathbf{v}))}{\partial w_{ij}} = \langle v_i h_j \rangle_{p(\mathbf{v}, \mathbf{h})} - \langle v_i h_j \rangle_{q^+(\mathbf{v}, \mathbf{h})} \quad (8)$$

其中 $\langle x \rangle_p$ 表示随机变量 x 在概率分布 p 下的期望, $q^+(\mathbf{v}, \mathbf{h}) = q(\mathbf{v})p(\mathbf{h} | \mathbf{v})$. 但模型分布的统计量 $\langle v_i h_j \rangle_{p(\mathbf{v}, \mathbf{h})}$ 无法有效计算, 而用样本估计会因为方差较大而使训练收敛很慢. 一种解决方法是构造一个有意义的中间分布 $r(\mathbf{v})$, 并将目标函数修改为

$$G = \text{KL}(q(\mathbf{v}) \| p(\mathbf{v})) - \text{KL}(r(\mathbf{v}) \| p(\mathbf{v})) \quad (9)$$

这个目标函数对参数的偏导数为

$$\frac{\partial G}{\partial w_{ij}} = \langle v_i h_j \rangle_{r^+(\mathbf{v}, \mathbf{h})} - \langle v_i h_j \rangle_{q^+(\mathbf{v}, \mathbf{h})} \quad (10)$$

其中 $r^+(\mathbf{v}, \mathbf{h}) = r(\mathbf{v})p(\mathbf{h} | \mathbf{v})$. 式(10)不用计算模型分布的统计量, 代之以在中间分布 $r(\mathbf{v})$ 中采样. $r(\mathbf{v})$ 往往选有特定意义的分布来使式(10)和式(8)表示的梯度在方向上相近, CD 算法^[9] 和 PCD 算法^[10,11] 就代表了其中两种选取方法. CD 算法里用训练数据经若干次重构后的样本表示 $r(\mathbf{v})$. 这样一条截断

的马尔可夫链解决了样本完全混合成模型分布后采样噪声过大的问题. PCD 算法中 $r(\mathbf{v})$ 在以任意分布的样本做一次初始化后,就在更新模型参数的同时,用更新后的模型对这些样本粒子继续做重构. 我们在第 7 节“事件驱动的 CD 算法”和第 8 节“神经采样的采样特点”中还会具体介绍这两种算法.

3 神经采样原理

神经采样用马尔可夫过程定量地描述脉冲神经网络的演化过程,并实现从目标分布中采样. 下面首先介绍生理神经网络演化中的不可逆现象,包括迟滞效应(refractoriness)和突触后膜效应(post-synaptic effect);然后引入离散时间神经采样方法里用神经元活动标记随机变量的方法,和马尔可夫转移算符的设计.

生理神经网络的许多活动都跟电现象有关,最常见的两种是迟滞效应和突触后膜效应. 神经元之间通过电脉冲来传递信息. 发放了一个脉冲后,神经元需要一段时间才能恢复正常发放脉冲的能力,这个现象称为迟滞效应. 而接收到电脉冲的神经元,它的膜电压会被改变一段时间,这个现象称为突触后膜效应. 相比于脉冲信号只持续大概 1ms,迟滞效应和突触后膜效应持续更长的时间(5ms~100ms);而且这两个效应在时间的正反两个方向上是不对称的,属于不可逆过程. 文献[5]解决了如何用脉冲来标记样本,和如何用不可逆马尔可夫链来描述脉冲神经网络演化的问题. 我们会在下面先介绍离散时间的神经采样方法,在第 4 节中会推广到更适合描述生物过程和类脑硬件运作的连续时间的神经采样动力学.

将迟滞效应和突触后膜效应持续的时间都设为整数 τ ,如果在当前时间 t ,神经元 k 在 $\{t-(\tau-1), \dots, t\}$ 中任一时间点发放了脉冲,那么这个神经元代表的一个二值随机变量 z_k 编码为 1,否则编码为 0,即

$$z_k(t) = \begin{cases} 1, & t - t_f^k \in \{0, 1, \dots, \tau - 1\} \\ 0, & \text{其它情况} \end{cases} \quad (11)$$

其中 t_f^k 为神经元 k 最近一次放电脉冲的时间. 离散时间神经采样的目标是实现整个脉冲神经网络在稳定时,每一时刻编码的向量 $(z_1(t), \dots, z_N(t))$ 是从目标分布 $p(z_1, \dots, z_N)$ 中采集的一个样本,其

中 N 是采样神经元的数目. 为了在马尔可夫过程中体现生理脉冲神经网络的迟滞效应,每个神经元额外有一个随机变量 ζ_k 标记它距离自身迟滞效应结束的时间:

$$\zeta_k(t) = \begin{cases} \tau - (t - t_f^k), & t - t_f^k \in \{0, 1, \dots, \tau - 1\} \\ 0, & \text{其它情况} \end{cases} \quad (12)$$

我们称 ζ_k 为神经元 k 的内部状态. 如果神经元仍然受到自身迟滞效应影响($\zeta_k > 0, z_k = 1$),那么它再发放脉冲的概率会较低;否则神经元处于静息状态($\zeta_k = 0, z_k = 0$). 在如上述用随机变量标记好神经网络的状态后,神经采样的基本思路是定义关于这些随机变量的分布 $p(z_1, \dots, z_N, \zeta_1, \dots, \zeta_N)$,使它的边缘分布是待采样的目标分布

$$p(z_1, \dots, z_N) = \sum_{\zeta_1, \dots, \zeta_N} p(z_1, \dots, z_N, \zeta_1, \dots, \zeta_N) \quad (13)$$

同时构造以这个分布为不变分布、而且符合生理神经网络迟滞性与突触后膜效应的马尔可夫转移算符.

已知一个神经元编码的 z_k ,它的内部状态均匀分布于所有可能的态,而且独立于其他神经元:

$$p(\zeta_k | z_k, z_{\setminus k}, \zeta_{\setminus k}) = \begin{cases} \frac{1}{\tau}, & \zeta_k = 1, \dots, \tau, z_k = 1 \\ 1, & \zeta_k = 0, z_k = 0 \\ 0, & \text{其它情况} \end{cases} \quad (14)$$

式中 $z_{\setminus k}$ 和 $\zeta_{\setminus k}$ 表示除第 k 个神经元以外其它神经元编码的随机变量和所处的内部状态. 图 1 是一个神经元的不同内部状态相互间转移的图示,箭头表示概率非零的马尔可夫链转移方向.

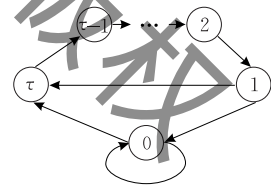


图 1 单个神经元内部状态之间的转移(其中,每个圈代表一种神经元状态 $\zeta_k, \zeta_k = 1, \dots, \tau$ 决定了 $z_k = 1, \zeta_k = 0$ 决定了 $z_k = 0$)

第 k 个神经元的演化状态用 ζ_k 表示, T^k 代表第 k 个神经元的状态转移算符. 在发放电脉冲($\zeta_k = \tau$)以后,神经元受迟滞效应影响在 $\tau-1$ 步内不能再发放脉冲,而只能确定地衰减到下一个内部状态

$$T^k(s-1 | s) = 1, \quad \forall s \in \{2, \dots, \tau\} \quad (15)$$

这相当于一个计数的过程. 神经元转移到发放脉冲的概率为

$$T^k(\tau | 1) = T^k(\tau | 0) = \sigma(u_k - \log \tau) \quad (16)$$

$$T^k(\tau | s) = 0, \forall s \in \{2, \dots, \tau\} \quad (17)$$

同时, 转移到(或维持在)静息状态($\zeta_k = 0$)的概率为

$$T^k(0 | 1) = T^k(0 | 0) = 1 - \sigma(u_k - \log \tau) \quad (18)$$

$$T^k(0 | s) = 0, \forall s \in \{2, \dots, \tau\} \quad (19)$$

其中

$$u_k = \log \frac{p(z_k = 1 | z_{\setminus k})}{p(z_k = 0 | z_{\setminus k})} \quad (20)$$

在神经采样中称为膜电压. 每个神经元 k 按照这样的马尔可夫转移算符 T^k 更新自己的状态, 就得到让整个神经网络的 $(\zeta_1, \dots, \zeta_N)$ 更新一次的马尔可夫转移算符

$$T = T^1 \circ T^2 \circ \dots \circ T^N \quad (21)$$

而 z_k 的值可由 ζ_k 唯一确定. 式(21)中 \circ 表示算符的串联. 文献[5]证明, $p(\zeta_1, \dots, \zeta_N)$ 是 T 的不变分布. 在神经采样中, 采样神经元正确地在目标分布中采样的充分条件是每个采样神经元的膜电压满足式(20), 这个条件称为 NCC 条件(Neural Computable Condition).

除了能在联合概率分布下采样, 神经采样也能做后验概率推断. 假如观察到变量 z_1, \dots, z_l 的取值, 那么计算分布

$$p(z_{l+1}, \dots, z_N | z_1, \dots, z_l) \quad (22)$$

就是一个后验概率推断问题. 如果观察到 $z_i = 1$, 那么将对应采样神经元的膜电压 u_i 保持为一个相当大的数, 使它以最高频率放电; 如果观察到 $z_i = 0$, 则将 u_i 保持为一个相当小的负数, 使这个采样神经元始终不放电. 然后整个网络仍然根据转移算符 T 演化, MCMC 原理保证经过足够长计算时间后, 未观察变量的采样神经元即在后验概率下采样.

4 连续时间的神经采样动力学

离散时间的神经采样适用于计算机模拟, 而连续时间的动力学演化更符合生物神经网络的演化和类脑硬件的运作. 文献[5]将离散时间神经采样的状态编码方式和马尔可夫过程构造方法推广到连续时间.

连续时间神经采样的状态编码方式跟离散时间中相似: 如果在过去的 τ 时间内, 神经元 k 发放了脉冲, 那么 $z_k = 1$; 如果没有, 则 $z_k = 0$. 第 k 个神经元的编码 $z_k(t)$ 与内部状态 $\zeta_k(t)$ 分别为

$$z_k(t) = \begin{cases} 1, & t - t_f^k \leq \tau \\ 0, & t - t_f^k > \tau \end{cases} \quad (23)$$

$$\zeta_k(t) = \begin{cases} \frac{t - t_f^k}{\tau}, & t - t_f^k \leq \tau \\ \text{处于 } I_\epsilon \text{ 中一个随机值}, & t - t_f^k > \tau \end{cases} \quad (24)$$

其中 $I_\epsilon := [-2\epsilon, -\epsilon]$. 当神经元处于迟滞过程, 即 $z_k(t) = 1$, 状态 $\zeta_k(t)$ 表示归一化后距离最近一次发放脉冲的时间. 观察到 z_k 后 ζ_k 仍然是独立于 $z_{\setminus k}$ 和 $\zeta_{\setminus k}$, 且处于均匀分布:

$$p(\zeta_k | z_k, z_{\setminus k}, \zeta_{\setminus k}) = \begin{cases} 1, & \zeta_k \in [0, 1], z_k = 1 \\ \frac{1}{\epsilon}, & \zeta_k \in I_\epsilon, z_k = 0 \\ 0, & \text{其它情况} \end{cases} \quad (25)$$

神经元 k 的马尔可夫转移算符设计为

$$T^k(\zeta_k) := \frac{1}{\tau} \left\{ \frac{\partial [p(\zeta_k | \zeta_{\setminus k}) \chi_{R^+}(\zeta_k)]}{\partial \zeta_k} - \delta(\zeta_k) F(p(\zeta_k | \zeta_{\setminus k})) + \exp(u_k) \delta(\zeta_k - 1) \int_{I_\epsilon} p(\zeta'_k | \zeta_{\setminus k}) d\zeta'_k + \chi_{I_\epsilon}(\zeta_k) [\epsilon^{-1} F(p(\zeta_k | \zeta_{\setminus k})) - \exp(x) p(\zeta_k | \zeta_{\setminus k})] \right\} \quad (26)$$

其中 $\chi_I(x)$ 是集合 I 的示性函数, 当 $x \in I$, $\chi_I(x) = 1$, 否则 $\chi_I(x) = 0$; $\delta(x)$ 是 Kronecker delta 函数, 当 $x = 0$, $\delta(x) = 1$, 否则 $\delta(x) = 0$; F 是个泛函, $F(q) = \lim_{x \rightarrow 0^+} q(x)$; $p(\zeta_k | \zeta_{\setminus k}) = \sigma(u_k) \chi_{[0, 1]}(\zeta_k) + (1 - \sigma(u_k)) \epsilon^{-1} \chi_{I_\epsilon}(\zeta_k)$ 可由式(25)计算得到; u_k 满足 NCC 条件(20). 于是整个神经网络并行更新的马尔可夫转移算符为

$$T = \prod_{k=1}^N T^k \quad (27)$$

文献[5]证明这样设计的状态空间分布和马尔可夫转移算符满足稳定的 Fokker-Plank 方程:

$$\frac{\partial p(\zeta_1, \dots, \zeta_N)}{\partial t} = T p(\zeta_1, \dots, \zeta_N) = 0 \quad (28)$$

即 $p(\zeta_1, \dots, \zeta_N)$ 是 T 的不变分布. 演化过程中 z_k 的值由 ζ_k 唯一确定. 所以以算符 T 演化的脉冲神经网络的状态最终会在 $p(z_1, \dots, z_N)$ 中采样.

5 用神经采样在任意贝叶斯网络中做推理

第 3 节“神经采样原理”已经证明, 当采样神经

元的膜电压满足 NCC 条件(20)时,这个脉冲神经网络能在目标分布中采样.文献[5]发现在玻尔兹曼分布下 NCC 条件有简单的解析表达式.假设一个玻尔兹曼分布

$$p(z_1, \dots, z_N) = \frac{1}{Z} \exp\left(\sum_{i \neq j} w_{ij} z_i z_j + \sum_i b_i z_i\right) \quad (29)$$

那么编码 z_i 的采样神经元 pn_i 的 NCC 条件是

$$u_i = \log \frac{p(z_i = 1 | \mathbf{z}_{\setminus i})}{p(z_i = 0 | \mathbf{z}_{\setminus i})} = \sum_{j \neq i} w_{ij} z_j + b_i \quad (30)$$

其中 u_i 为 pn_i 的膜电压, $\sum_{j \neq i} w_{ij} z_j$ 为其它采样神经元在 pn_i 胞体产生的突触后膜电压(Post-Synaptic Potential, PSP), b_i 称为 pn_i 的自激发电压. i 的膜电压是 PSP 和自激发电压的和.

然而一般概率分布的 NCC 条件没有解析表达式,如果用数值方法枚举不同 z_i 取值下 pn_i NCC 条件的取值,那么枚举的复杂度正比于 $2^{\max_i |z_i|}$, $|x|$ 表示向量 x 的维度.文献[6]发现,在贝叶斯网络中,变量间的相互独立性可以降低 NCC 条件取值的枚举复杂度.贝叶斯网络是指有向无环的概率图模型.这一节我们介绍文献[6]的工作,包括两种在贝叶斯网络中降低 NCC 条件取值的枚举复杂度的数学原理(马尔可夫覆盖性质和因子分解方法)及如何使用辅助神经元和树突结构在脉冲神经网络中实现 NCC 条件的枚举.

在一个能用贝叶斯网络表示的概率分布中,如果观察到随机变量 z_i 的马尔可夫覆盖(Markov blanket) \mathbf{z}^{B_i} 中随机变量的状态,那么 z_i 的后验概率分布与马尔可夫覆盖外的随机变量相互独立:

$$p(z_i | \mathbf{z}_{\setminus i}) = p(z_i | \mathbf{z}^{B_i}) \quad (31)$$

\mathbf{z}^{B_i} 由 z_i 在网络中的父母和孩子组成,用马尔可夫覆盖来表示采样神经元 pn_i 的 NCC 条件:

$$u_i = \log \frac{p(z_i = 1 | \mathbf{z}^{B_i})}{p(z_i = 0 | \mathbf{z}^{B_i})} = \sum_{\mathbf{v} \in Z^{B_i}} M_v^i[\mathbf{z}^{B_i} = \mathbf{v}] \quad (32)$$

其中 Z^{B_i} 表示 \mathbf{z}^{B_i} 所有可能取值的集合; $M_v^i = \log[p(z_i = 1 | \mathbf{z}^{B_i} = \mathbf{v}) / p(z_i = 0 | \mathbf{z}^{B_i} = \mathbf{v})]$ 表示 $\mathbf{z}^{B_i} = \mathbf{v}$ 时 pn_i NCC 条件的取值; $[\mathbf{z}^{B_i} = \mathbf{v}]$ 在 $\mathbf{z}^{B_i} = \mathbf{v}$ 时取 1, 否则取 0. 利用马尔可夫覆盖的性质能将枚举 NCC 条件取值的复杂度降低到与网络中最大的马尔可夫覆盖的大小呈指数关系,即正比于 $2^{\max_i |z^{B_i}|}$.

贝叶斯网络的另一个性质是能对概率分布做因子分解:

$$p(\mathbf{z}) = \frac{1}{Z} \prod_{c=1}^C \phi_c(\mathbf{z}^c) \quad (33)$$

其中 Z 为归一化系数, \mathbf{z}^c 表示因子 ϕ_c 依赖的向量, C 表示因子的总数. 因子分解也降低了枚举 NCC 条件取值的复杂度:

$$\begin{aligned} \log \frac{p(z_i = 1 | \mathbf{z}_{\setminus i})}{p(z_i = 0 | \mathbf{z}_{\setminus i})} &= \log \prod_{c \in C^i} \frac{\phi_c(z_i = 1, \mathbf{z}_{\setminus i}^c)}{\phi_c(z_i = 0, \mathbf{z}_{\setminus i}^c)} \\ &= \sum_{c \in C^i} \left(\sum_{\mathbf{v} \in Z_{\setminus i}^c} M_v^{c,i}[\mathbf{z}_{\setminus i} = \mathbf{v}] \right) \end{aligned} \quad (34)$$

其中 C^i 表示依赖 z_i 的所有因子, $Z_{\setminus i}^c$ 表示在因子 c 中除了 z_i 外其它随机变量 $\mathbf{z}_{\setminus i}^c$ 所有可能的取值, $M_v^{c,i} = \log[\phi_c(z_i = 1, \mathbf{z}_{\setminus i}^c = \mathbf{v}) / \phi_c(z_i = 0, \mathbf{z}_{\setminus i}^c = \mathbf{v})]$ 表示 $\mathbf{z}_{\setminus i}^c = \mathbf{v}$ 对 pn_i NCC 条件贡献的大小. 因子分解方法的枚举复杂度与网络中最大因子的维度呈指数关系, 正比于 $2^{\max_{c \in C^i} |z_{\setminus i}^c|}$, 小于用马尔可夫覆盖性质的枚举复杂度.

其实,无论如何用数学原理降低 NCC 条件的枚举复杂度,实现枚举功能的网络都会对采样网络中的具体放电模式有选择性响应.文献[6]用辅助神经元和树突实现选择性响应.辅助神经元的选择性放电用神经采样的单神经元模型实现.假设当某些随机变量组成的向量 \mathbf{z}^s 取特定值 \mathbf{v} 时辅助神经元 an^s 才响应,那么如果 $\mathbf{v}(i) = 0$, 让 pn_i^s 到 an^s 有很强的抑制性突触连接; 如果 $\mathbf{v}(i) = 1$, 让 pn_i^s 到 an^s 有很强的兴奋性突触连接,从而只有当 $\mathbf{z}^s = \mathbf{v}$ 时, an^s 有非零的几率发放脉冲.这里 i 标记 \mathbf{z}^s 的分量, $\mathbf{v}(i)$ 表示 \mathbf{v} 的 i 分量, i 也标记对应采样神经元组 pn^s 里的 i 神经元.

除了辅助神经元,树突也可以实现选择性响应.在带树突的神经元模型中,其他神经元传来的突触会先连接在树突上.当这些突触在树突内产生的 PSP 高于一个阈值时,树突就会向胞体发放树突脉冲改变胞体的膜电压.这个过程的数学描述为:设这个阈值电压为 θ , 其他神经元在树突 i 内产生的 PSP 为 $a_i(t)$, 那么树突 i 在胞体产生的膜电压为

$$\hat{a}_i(t) = \beta_i H(a_i(t) - \theta) \quad (35)$$

其中 β_i 是个正数,表示第 i 个树突的树突强度; $H(x)$ 为 Heaviside step 函数: 当 $x > 0$ 时 $H(x) = 1$, 当 $x < 0$ 时 $H(x) = 0$, 当 $x = 0$ 时 $H(x) = 1/2$. 胞体的膜电压等于自激发电压 b 与所有树突贡献的膜电压的和:

$$u(t) = b + \sum_i \hat{a}_i(t) \quad (36)$$

通过调节其它神经元连接到树突的强度,可以实现只有当输入神经元以某种模式放电时,树突 i 内的 PSP 才高于阈值,从而实现通过树突有选择性地改变胞体的膜电压. 下面我们具体讲文献[6]里 3 个不同模型中如何设置参数的数值使采样神经元通过枚举方法分别满足用马尔可夫覆盖方法和因子分解方法简化的 NCC 条件. 这些参数包括突触强度 ω 、辅助神经元和采样神经元的自激发电压 b 、树突的阈值 θ 和突触强度 β . 为了简化问题并且不失一般性,下面讲述中我们只考虑编码随机变量 z_i 的采样神经元 pn_i 的 NCC 条件.

第 1 个模型只使用辅助神经元来实现马尔可夫覆盖简化的 NCC 条件. 模型参数设置的目的在于满足下面 3 个部分:(1) 每个辅助神经元 an_i^v 代表马尔可夫覆盖 z^{B_i} 的一种取值 v , an_i^v 能放电当且仅当 $z^{B_i} = v$; (2) 当 $z^{B_i} = v$ 时, an_i^v 的膜电压满足 pn_i 在 $z^{B_i} = v$ 的 NCC 条件; (3) 当任意辅助神经元 an_i^v 放电后, pn_i 马上放电, 否则 pn_i 保持静息. 这 3 个部分共同作用相当于每个辅助神经元 an_i^v 以 $z^{B_i} = v$ 时的 NCC 条件来代替 pn_i 放电, $v \in Z^{B_i}$. 图 2 是一个根据这个方法设计的简单采样神经网络的示意图.

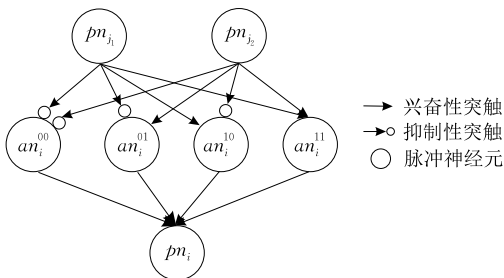


图 2 使用辅助神经元实现根据马尔可夫覆盖简化的 NCC 条件设计的简采样神经网络. 随机变量 z_i 的马尔可夫覆盖只包含变量 z_{j_1} 和 z_{j_2} , 这 3 个变量分别对应采样神经元 pn_i , pn_{j_1} 和 pn_{j_2} . 4 个辅助神经元 an_i^v ($v=00, 01, 10, 11$) 不但接收来自 pn_{j_1} 和 pn_{j_2} 很强的兴奋性或抑制性连接, 而且各自有不同的自激发电压(参数设置方法见正文部分), 同时有很强兴奋性突触连接到 pn_i .

第 1 和第 2 部分通过准确设置 $pn_j^{B_i}$ 到 an_i^v 的突触强度及 an_i^v 的自激发电压来实现, 其中 $pn_j^{B_i}$ 表示编码马尔可夫覆盖 z^{B_i} 里第 j 个随机变量的采样神经元. 设一个正数 $M_i \gg M_i^0 = \log[p(z_i=1 | z^{B_i}=v) / p(z_i=0 | z^{B_i}=v)]$, 当 $v(j)=1$ 时, $pn_j^{B_i}$ 到 an_i^v 的连接强度设置为 M_i ; 当 $v(j)=0$, $pn_j^{B_i}$ 到的 an_i^v 连接强度

设置为 $-M_i$; an_i^v 的自激发电压设置为 $M_i^0 - \eta(v)M_i$, $\eta(v)$ 表示 v 中取值为 1 的分量个数. 神经元的膜电压是 PSP 和自激发电压的和. 这样, 当 $z^{B_i} \neq v$ 时, an_i^v 要么没有获得足够的兴奋性 PSP, 要么获得过多的抑制性 PSP, 两种情况下 an_i^v 的放电概率都接近 0; 只有当 $z^{B_i} = v$ 时, an_i^v 膜电压为 M_i^0 , 满足 pn_i 在 $z^{B_i} = v$ 时的 NCC 条件. 然后将 pn_i 的自激发设为较低负值(比如 $-M_i$), an_i^v 到 pn_i 的突触连接为远远大于这个自激发幅度的一个正数(比如 $2M_i$)就能实现第 3 部分: 静息时 pn_i 的放电概率接近 0, 而 an_i^v 放电后 pn_i 会马上放电.

上一个模型采用辅助神经元实现马尔可夫覆盖简化的 NCC 条件, 在第 2 个模型中用树突来实现. 这个模型具体实现以下两个部分:(1) 每个树突 δ_i^v 代表马尔可夫覆盖 z^{B_i} 的一种取值 v , δ_i^v 对胞体有影响当且仅当 $z^{B_i} = v$; (2) δ_i^v 对胞体膜电压的影响使 pn_i 满足在 $z^{B_i} = v$ 时的 NCC 条件. 图 3 是一个根据这个方法设计的简单采样神经网络的示意图.

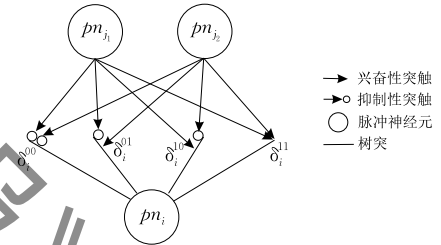


图 3 使用树突来实现根据马尔可夫覆盖简化的 NCC 条件设计的简采样神经网络. 随机变量 z_i 的马尔可夫覆盖只包含变量 z_{j_1} 和 z_{j_2} , 这 3 个变量分别对应采样神经元 pn_i , pn_{j_1} 和 pn_{j_2} . 属于 pn_i 的 4 个树突 δ_i^v ($v=00, 01, 10, 11$) 不但接收来自 pn_{j_1} 和 pn_{j_2} 很强的兴奋性或抑制性连接, 而且各自有不同的阈值和连接到 pn_i 的树突强度(参数设置方法见正文部分).

第 1 部分通过设置 $pn_j^{B_i}$ 到 δ_i^v 的突触强度及树突 δ_i^v 的阈值来实现. 其中一种参数设计的方法是: 设一个正数 $M \gg 0$, 与模型 1 类似, 当 $v(j)=1$ 时, $pn_j^{B_i}$ 到 δ_i^v 的连接强度设置为 M ; 当 $v(j)=0$, $pn_j^{B_i}$ 到 δ_i^v 的连接强度设置为 $-M$, δ_i^v 的阈值设置为 $(\eta(v) - \frac{1}{2})M$. 这样, 当 $z^{B_i} \neq v$ 时, 树突的 PSP 都低于阈值, 树突不会影响胞体的膜电压; 当 $z^{B_i} = v$ 时, PSP 高于阈值, 树突改变胞体的膜电压. 第 2 部分通过设置 δ_i^v 到 pn_i 胞体的树突强度和 pn_i 的自激发电压来实现. 树

突强度设置为 $\beta_i^c = \lambda_i + \log[p(z_i = 1 | \mathbf{z}^{B_i} = \mathbf{v}) / p(z_i = 0 | \mathbf{z}^{B_i} = \mathbf{v})]$, λ_i 是能将 β_i^c 修正成正数的任意数, 一般设 $\lambda_i = -\min_{v \in z^{B_i}} \log[p(z_i = 1 | \mathbf{z}^{B_i} = \mathbf{v}) / p(z_i = 0 | \mathbf{z}^{B_i} = \mathbf{v})]$, 即采样神经元 pn_i 所有 NCC 条件里最小值的相反数; pn_i 的自激发设为 $-\lambda_i$. 这样就实现当树突 $\delta_{i,c}^v$ 响应时, 使胞体的膜电压满足 $\mathbf{z}^{B_i} = \mathbf{v}$ 时的 NCC 条件.

第 3 个模型只使用树突来实现因子分解方法简化的 NCC 条件. 模型参数实现下面两个部分: (1) 每个树突 $\delta_{i,c}^v$ 代表因子 c 里向量 $\mathbf{z}_{i,c}^v$ 的不同取值 \mathbf{v} , $\delta_{i,c}^v$ 对胞体有影响当且仅当 $\mathbf{z}_{i,c}^v = \mathbf{v}$; (2) $\delta_{i,c}^v$ 对胞体膜电压的影响等于 $\mathbf{z}_{i,c}^v = \mathbf{v}$ 为 pn_i 的 NCC 条件贡献的分量. 图 4 是一个根据这个方法设计的简单采样神经网络的示意图.

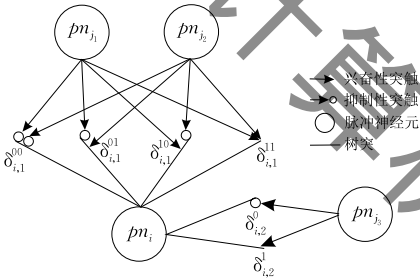


图 4 使用树突来实现根据因子分解简化的 NCC 条件设计的简易采样神经网络. 示例中包含随机变量 z_i 的因子有两个, 其中一个因子还包含变量 z_{j_1} 和 z_{j_2} , 另一个因子还包含变量 z_{j_3} , 这 4 个变量分别对应采样神经元 pn_i, pn_{j_1}, pn_{j_2} 和 pn_{j_3} . 树突小分支 $\delta_{i,1}^v (v=00,01,10,11)$ 接收来自 pn_{j_1} 和 pn_{j_2} 的连接, 小分支 $\delta_{i,2}^v (v=0,1)$ 接收来自 pn_{j_3} 的连接. 这些树突小分支各自有不同的阈值和连接到 pn_i 的树突强度(参数设置方法见正文部分).

模型 3 和模型 2 的一个主要不同点是, 每一时刻在模型 2 中, pn_i 上有且只有一个树突对膜电压有影响; 而模型 3 中, 每个因子中都有且只有一个树突对膜电压有影响, 而来自不同因子的影响在胞体线性叠加. 为了实现第 1 部分, 在因子 c 中用和模型 2 相似的方法, 如果 $v(j) = 1$, pn_{j,ν_i}^c 到 $\delta_{i,c}^v$ 的连接强度设置为 M ; 如果 $v(j) = 0$, pn_{j,ν_i}^c 到 $\delta_{i,c}^v$ 的连接强度设置为 $-M$, 其中 pn_{j,ν_i}^c 表示编码因子 c 里第 j 个随机变量的采样神经元(不包括当前的采样神经元 pn_i). 第 2 部分通过将 $\delta_{i,c}^v$ 的树突强度设置为 $\beta_{i,c}^v = \lambda_{i,c} + \log[\phi_c(z_i = 1, \mathbf{z}_{i,c}^v = \mathbf{v}) / \phi_c(z_i = 0, \mathbf{z}_{i,c}^v = \mathbf{v})]$, 其中 $\lambda_{i,c}$ 是能使因子 c 中所有树突的强度为正数的任意

数, 一般设 $\lambda_{i,c} = -\min_{v \in z_{i,c}^v} \log[\phi_c(z_i = 1, \mathbf{z}_{i,c}^v = \mathbf{v}) / \phi_c(z_i = 0, \mathbf{z}_{i,c}^v = \mathbf{v})]$ 为因子 c 中所有 NCC 条件的分量里最小值的相反数; pn_i 的自激发电压设为 $-\sum_{c \in c^i} \lambda_{i,c}$. 此时, $\delta_{i,c}^v$ 对胞体膜电压的影响即等于 $\mathbf{z}_{i,c}^v = \mathbf{v}$ 为 pn_i 的 NCC 条件贡献的分量, 每个因子中树突影响的总和使 pn_i 满足 NCC 条件.

结合神经采样动力学和采样网络的参数设计方法, 就可以用神经采样对玻尔兹曼分布和任意贝叶斯网络做推理. 因为网络中最大因子的维度总是不大于最大马尔可夫覆盖的维度, 所以第 3 个模型会比第 1 个、第 2 个模型使用更少数量的辅助单元(辅助神经元或树突). 而模型 1 和 2 的主要差别仅在于使用不同的辅助单元. 在第 7 节“事件驱动的 CD 算法”和第 8 节“用 RBM 的训练算法分析和展示神经采样的采样特点”中将看到神经采样在 RBM 上做推理和训练的应用. 在这之前, 我们先讨论如何在类脑硬件上实现神经采样.

6 用随机 I&F 模型近似连续时间的神经采样

虽然神经采样能让脉冲神经网络在演化过程中采样, 但目前没有类脑硬件能直接实现它. 现在类脑硬件用 I&F 神经元模型实现脉冲神经网络. 接下来将介绍文献[7, 15]在用随机 I&F 神经元模型近似神经采样方面的工作, 它们的工作为在目前的类脑硬件上实现神经采样提供了思路. 用随机 I&F 神经元模型近似神经采样会分成两部分近似: 一是神经元胞体对输入的响应; 二是其他神经元的输入.

在连续时间的神经采样模型(见第 4 节“连续时间的神经采样动力学”)中, 神经元平均放电频率 ρ 是关于膜电压 u 的 Sigmoid 函数:

$$\rho(u) = \tau^{-1} \left(1 + \frac{\exp(-u)}{\tau} \right)^{-1} \quad (37)$$

而随机 I&F 神经元模型的特点正是神经元的平均放电频率随注入神经元胞体的电流 I 的变化可以近似于一个 Sigmoid 函数. 在随机 I&F 神经元模型中, 设注入胞体的电流为 $I(t)$, 膜电压 $u(t)$ 在阈值 θ 以下时, $u(t)$ 的演化满足:

$$C \frac{d}{dt} u(t) = -g_L u(t) + I(t) + \epsilon \xi(t),$$

$$u(t) \in (-\infty, \theta) \quad (38)$$

其中 C 称为膜电容, g_L 称为漏电电导率, $\epsilon \xi(t)$ 是幅

度为 ι 的白噪声, 当膜电压达到阈值 $u(t) = \theta$ 时, 胞体马上发放脉冲, 然后在一个低电压值 $u(t) = u_{\text{reset}}$ 维持 τ 时间. 在这个模型中, 参数 $(\theta - u_{\text{reset}}) / \iota$ 越小, 神经元的平均放电频率随电流的变化越接近 Sigmoid 函数, 而且能被

$$\nu(I | \beta, \gamma) = \tau^{-1} \left(1 + \frac{\exp(-\beta I)}{\gamma \tau} \right)^{-1} \quad (39)$$

形式的 Sigmoid 函数很好地拟合, 其中 β 和 γ 是待定参数.

第 2 部分近似是在随机 I&F 模型中用树突来将脉冲转换成随时间指数衰减的电流注入到胞体, 来近似神经采样中采样神经元短时内大小不变的 PSP 带来的影响. 在 I&F 模型中电脉冲会先被树突接收转化, 再以电流的方式传递到胞体. 设从突触 j 输入的脉冲序列为

$$s_j = \sum_{k \in S_j} \delta(t - t_{jk}) \quad (40)$$

其中 S_j 是从突触 j 输入脉冲的集合, $\delta(\cdot)$ 是狄拉克 delta 函数, t_{jk} 为突触 j 的脉冲 k 发放的时间. 那么树突的电流 $I(t)$ 满足:

$$\tau_{\text{syn}} \frac{d}{dt} I(t) = -I(t) + \sum_j q_j s_j \quad (41)$$

其中 τ_{syn} 为突触时间常数, q_j 为突触 j 的强度. 树突被建模为一阶滤波器. 这样的树突和胞体结构组成一个多组分的神经元模型, 突触传递来的电脉冲会被转化为指数衰减的电流, 来近似矩形 PSP 的影响.

综上所述, 可以通过校准随机 I&F 模型网络的参数用近似神经采样, 步骤如下:

假设现在校准第 i 个 I&F 神经元 IF_i , 它近似采样神经元 pn_i .

(1) 给 IF_i 胞体恒定地注入很大的电流使放电频率接近最大值 ν_{max} , 得到这个 I&F 神经元的迟滞时间 $\tau = 1/\nu_{\text{max}}$.

(2) 给 IF_i 胞体恒定地注入不同的电流 i , 测量它的平均放电频率 $\nu(i)$, 然后用 $-\beta i + \log(\gamma)$ 的形式来线性拟合 $\log(\nu(i)^{-1} - \tau)$, 得到参数 β 和 γ .

(3) 校准参数, $q_{ij} = \omega_{ij} / \beta$, $I_i = b_i - \log \gamma$, 其中 ω_{ij} 和 b_i 分别是神经采样网络中的突触强度和自激发电压, q_{ij} 是 IF_j 到 IF_i 的突触强度.

经校准后的 I&F 神经元网络以 I&F 模型的动力学演化, 能近似神经采样. 如果想做后验概率推理, 可以给对应于观察变量的 IF_i 注入足够大的正

电流或足够小的负电流 (分别对应用神经采样做后验推理时给 pn_i 的胞体注入足够大的正电压或足够小的负电压), 然后让网络以 I&F 动力学演化, 那么对应未观察变量的神经元在后验概率做推理. 文献 [7] 用实验表明, 这样用随机 I&F 模型近似神经采样的采样误差主要来自于用指数衰减的电流近似短时内大小不变的 PSP.

7 事件驱动的 CD 算法

神经采样的一个应用前景是用脉冲神经网络动力学在类脑硬件上训练机器学习模型和完成智能任务. 文献 [7] 提出一个用连续时间神经采样和基于脉冲时间的突触可塑性 (Spike-Timing-Dependent Plasticity, STDP) 来训练 RBM 的 CD 算法. 在这个算法里突触强度以事件驱动的方式更新, 而且此算法有在线学习和异步运行的特点.

传统的 CD 算法 [9] 在训练 RBM 时参数更新的法则是 (见第 2 节 “MCMC 方法及其在 RBM 训练算法中的应用”):

$$\Delta w_{ij} = \epsilon (\langle v_i h_j \rangle_{q^+} - \langle v_i h_j \rangle_{q_{\text{recon}}^+}) \quad (42)$$

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$$

其中 ϵ 为学习率, $q_{\text{recon}}^+(\mathbf{v}, \mathbf{h}) = q_{\text{recon}}(\mathbf{v}) p(\mathbf{h} | \mathbf{v})$, $q_{\text{recon}}(\mathbf{v})$ 是环境分布 $q(\mathbf{v})$ 的重构. 经典计算机计算参数更新的传统方法是: 首先用训练样本及根据训练样本推断出隐层的后验分布来近似 $\langle v_i h_j \rangle_{q^+}$ (称为数据项); 然后对训练样本做若干步重构, 并推断重构后隐层的后验分布来近似 $\langle v_i h_j \rangle_{q_{\text{recon}}^+}$ (称为重构项). 相应将这两项代入式 (42) 即称为完成训练的一次迭代. 通常为了减少计算量, 每次迭代中只会用训练集中的部分训练样本. 这种每次迭代用一个子集 (mini-batch) 而非整个数据集的近似方法称为随机梯度下降方法. 在实际应用中, 经过足够多次迭代后 CD 算法通常会收敛到很好的解. 但这个方法并不能直接用脉冲神经网络来实现, 所以文献 [7] 设计出事件驱动的 CD, 实现直接用脉冲神经网络训练 RBM.

文献 [7] 首先设计了能对 RBM 做概率推断的脉冲神经网络. 如果用神经采样中的神经元 pn_i^{vis} 和 pn_j^{hid} 来编码 v_i 和 h_j , 那么神经采样的 NCC 条件是

$$u_i^{\text{vis}} = \sum_j \omega_{ij} h_j \quad (43)$$

$$u_j^{\text{hid}} = \sum_i \omega_{ij} v_i \quad (44)$$

其中 u_i^{vis} 和 u_j^{hid} 分别表示 pn_i^{vis} 和 pn_j^{hid} 的膜电压. 这样 w_{ij} 不但是 RBM 中的突触强度, 也是采样神经网络中的突触强度.

事件驱动 CD 算法体现了生理上一种称为 STDP 的基于脉冲的突触可塑性: 假设一个突触连接神经元 i 和神经元 j , 它的突触强度的改变受两个神经元发放脉冲的时间间隔影响. 据此设计的事件驱动 CD 算法参数更新的法则是

$$\frac{dw_{ij}}{dt} = g(t) \text{STDP}(s_i^{\text{vis}}(t), s_j^{\text{hid}}(t)) \quad (45)$$

其中 $s_i^{\text{vis}}(t)$ 和 $s_j^{\text{hid}}(t)$ 分别表示 pn_i^{vis} 和 pn_j^{hid} 的脉冲序列, 它的定义方法和式(40)一致. $g(t)$ 是个全局调节信号, 区分当前时间应该增强突触强度还是减弱突触强度. STDP($s_i^{\text{vis}}(t), s_j^{\text{hid}}(t)$) 根据两个脉冲神经元 pn_i^{vis} 和 pn_j^{hid} 的脉冲序列来改变突触强度的变化速率:

$$\text{STDP}(s_i^{\text{vis}}(t), s_j^{\text{hid}}(t)) = s_i^{\text{vis}}(t) A_i^{\text{hid}}(t) + s_j^{\text{hid}}(t) A_j^{\text{vis}}(t) \quad (46)$$

$$A_j^{\text{hid}}(t) = A \int_{-\infty}^t K(t-s) s_j^{\text{hid}}(s) ds \quad (47)$$

$$A_i^{\text{vis}}(t) = A \int_{-\infty}^t K(s-t) s_i^{\text{vis}}(s) ds \quad (48)$$

其中 $K(t)$ 是个对称的核函数, 即 $\forall t, K(-t) = K(t)$, A 称为突触强度改变的幅度. 式(47)和式(48)表示突触强度 w_{ij} 的改变量和 pn_i^{vis} 与 pn_j^{hid} 之间脉冲的时间间隔有关. 全局调节信号 $g(t)$ 以时长 T 将训练过程均匀地分成连续的数据相和重构相, 所以一次迭代的周期为 $2T$:

$$g(t) = \begin{cases} 0, & t \bmod T \in [0, \tau_{\text{brn}}] \\ 1, & t \bmod 2T \in [\tau_{\text{brn}}, T] \\ -1, & t \bmod 2T \in [T + \tau_{\text{brn}}, 2T] \end{cases} \quad (49)$$

其中 \bmod 为取模运算, τ_{brn} 是在这个 RBM 中做神经采样推断的预热时间 (burn-in time). 虽然 τ_{brn} 会因为 RBM 规模不同而改变, 但实验表明通常取若干个迟滞时间长度就足够. 设 y 为任意正整数, 时间区间 $[2yT, (2y+1)T]$ 称为数据相, 这时可见层的脉冲神经元表示一个训练样本, 隐层的脉冲神经元做后验推理; 时间区间 $[(2y+1)T, 2(y+1)T]$ 称为重构相, 这时整个网络自由演化. 每一相开始的 τ_{brn} 时间预留来让神经网络做 MCMC 的“预热”, 然后可以更新突触强度. $g(t) = 1$ 和 -1 分别对应 CD 算法中数据项和重构项中的增强和减弱突触强度的效果.

连续时间的神经采样模型和随机的 I&F 模型都能用事件驱动的 CD 算法来训练 RBM. 文献[7]用随机 I&F 模型训练出的 RBM 在 MNIST 数据集上的分类准确率达到 91.9%. 准确率上升的瓶颈是随机 I&F 模型在近似神经采样时存在误差.

8 神经采样的采样特点

为了研究神经采样的采样特点, 和这些特点对概率模型推理结果的影响, 我们用神经采样代替吉布斯采样, 在 CD 和 PCD 算法框架下训练 RBM (RBM 的介绍见第 2 节“MCMC 方法及其在 RBM 训练算法中的应用”). 相比吉布斯采样, 神经采样连续两个样本间依赖很强, 而且混合速度更慢. 为了解决这些问题, 我们通过修改 CD 和 PCD 算法, 提出神经采样 CD 和神经采样 PCD 算法. 虽然神经采样 CD 算法里解决问题的思想已经包含在的事件驱动的 CD 算法中 (见第 7 节“事件驱动的 CD 算法”), 但我们这里会加强对如何用神经采样做正确的重构进行理论分析和实验展示. 下面我们先理论分析神经采样的采样特点和介绍神经采样 CD 和神经采样 PCD 算法, 然后展示在 MNIST 数据集上初步的重构实验结果.

8.1 神经采样 CD 算法

将式(42)传统 CD 算法的参数更新法则重写在这里:

$$\begin{aligned} \Delta w_{ij} &= \epsilon (\langle v_i h_j \rangle_{q_{\text{recon}}^+(v, h)} - \langle v_i h_j \rangle_{q_{\text{recon}}^+(v, h)}) \\ w_{ij} &\leftarrow w_{ij} + \Delta w_{ij} \end{aligned} \quad (50)$$

式中需要用训练样本的重构来估计 $\langle v_i h_j \rangle_{q_{\text{recon}}^+(v, h)}$. 传统方法交替固定可见层、隐层并对另一层做 MCMC 推理. 在可见层和隐层间来回推理一次称为一步重构. 通常, 一步吉布斯更新重构的 CD 算法就能达到很好的训练效果. 在 RBM 中推理的神经采样网络只是将 RBM 中 Sigmoid 神经元模型换成神经采样神经元模型, 网络动力学将吉布斯采样换成神经采样动力学, 而突触强度保持不变 (见第 7 节“事件驱动的 CD 算法”). 将吉布斯采样换成神经采样后, 隐层采样神经元状态的初始化对于训练效果很关键, 不正确地初始化隐层会使训练效果很差 (见 8.3.1 节的实验结果). 神经采样 CD 算法主要解决了如何在每次重构前正确初始化隐层采样神经元状态的问题.

传统地用吉布斯更新的重构隐含了一个恰当初始化隐层的过程,从而保证 CD 算法正确性. 回忆式(9),CD 算法的目标函数是

$$G_{CD} = KL(q(\mathbf{v}) \parallel p(\mathbf{v})) - KL(q_{\text{recon}}(\mathbf{v}) \parallel p(\mathbf{v})) \quad (51)$$

其中 $q_{\text{recon}}(\mathbf{v})$ 是环境分布 $q(\mathbf{v})$ 的重构. 重构的本质是通过训练样本在一个包含隐层的扩大空间里做若干步 MCMC 后再将隐层抛弃,从而实现

$$KL(q(\mathbf{v}) \parallel p(\mathbf{v})) \geq KL(q_{\text{recon}}(\mathbf{v}) \parallel p(\mathbf{v})) \quad (52)$$

等号成立当且仅当 $q(\mathbf{v}) = p(\mathbf{v})$. 式(52)的直观理解是,通过在扩大的空间里做 MCMC,保证重构分布比环境分布更接近模型分布. 在附录中我们证明不等式成立的充分条件是当从训练样本扩大到包含隐层的样本时,隐层以后验概率 $p(\mathbf{h}|\mathbf{v})$ 初始化. 在吉布斯更新中,对隐层做一次吉布斯采样即可自动满足此初始化条件;但在神经采样中,迟滞效应使隐层的新样本依赖于隐层的初始状态. 因此,我们的神经采样 CD 算法在用训练样本初始化可见层后,对隐层用神经采样推理进行初始化,目的是将隐层分布混合成后验概率. 恰当初始化隐层后,我们的神经采样 CD 算法在可见层和隐层间来回做神经采样实现重构,参数更新的方法和传统方法^[16]一致.

事件驱动的 CD 算法采用类似初始化隐层采样神经元状态的方法,他们的做法是在可见层用新样本初始化后,先不更新参数而让隐层随神经采样动力学自由地混合成后验分布. 我们的神经采样 CD 算法和事件驱动的 CD 算法有以下不同点:(1)神经采样 CD 算法用离散时间的神经采样,而事件驱动 CD 算法用连续时间的神经采样(2)神经采样 CD 算法在获得样本后用传统方法更新参数,参数更新需要用传统计算机,每次参数更新可以使用一个子集的训练样本;而事件驱动的 CD 算法用事件驱动的方式更新参数,完全使用脉冲神经网络的特点,但每次只能观察一个训练样本. 我们在这里研究神经采样的采样特点,特别在理论和实验分析上指出在做后验概率推理时预热混合隐层采样神经元状态的重要性.

8.2 神经采样 PCD 算法

传统的 PCD 算法^[10]也是通过从中间分布 $r(\mathbf{v})$ 采样来代替从模型分布中采样,目的是估计参数更新的梯度方向. 留意到一次参数更新前后模型分布变化往往不大,PCD 算法提出用持续的马尔可夫链来混合 $r(\mathbf{v})$. 每次参数更新时对 $r(\mathbf{v})$ 的样本做重

构,而在下次参数更新时,用前一次重构的样本继续做重构,所以 $r(\mathbf{v})$ 的采样过程称为持续的马尔可夫链. 我们记这条持续马尔可夫链的分布为 $q_{\text{per}}(\mathbf{v})$, PCD 算法的参数更新法则为

$$\begin{aligned} \Delta w_{ij} &= \epsilon(\langle v_i h_j \rangle_{q^+(\mathbf{v}, \mathbf{h})} - \langle v_i h_j \rangle_{q_{\text{per}}^+(\mathbf{v}, \mathbf{h})}) \\ w_{ij} &\leftarrow w_{ij} + \Delta w_{ij} \end{aligned} \quad (53)$$

其中 $q_{\text{per}}^+(\mathbf{v}, \mathbf{h}) = q_{\text{per}}(\mathbf{v})p(\mathbf{h}|\mathbf{v})$. 当每次参数更新不大而且马尔可夫链混合速度足够快, $q_{\text{per}}(\mathbf{v})$ 能以有效的方式接近模型分布,从而比一般的 CD 方法更准确地估计梯度方向. 传统 PCD 算法用吉布斯采样在持续马尔可夫链中做重构. 在用神经采样替代吉布斯采样时,需考虑到在神经采样的脉冲神经网络动力学中存在迟滞过程,降低了神经采样作为一种 MCMC 采样算法的混合速度. 我们的解决方法是增加每次参数更新时神经采样重构的步数来提高混合度. 在 8.3.2 节的实验结果中能看到,增加重构步数显著提高神经采样 PCD 算法训练过程的稳定性.

和 CD 算法一样,PCD 算法中持续马尔可夫链的重构过程也需要满足重构后分布比重构前分布更接近模型分布,即

$$KL(q_{\text{per}}(\mathbf{v}) \parallel p(\mathbf{v})) \geq KL(q_{\text{recon}}(\mathbf{v}) \parallel p(\mathbf{v})) \quad (54)$$

其中 $q_{\text{recon}}(\mathbf{v})$ 为 $q_{\text{per}}(\mathbf{v})$ 的重构. 所以在以神经采样作为持续马尔可夫链时,在马尔可夫链的开端以任意分布初始化可见单元后,同样需用 8.1 节中提出的初始化方式正确初始化隐层.

8.3 实验结果

我们已经分析了神经采样在做推理时对初始化状态敏感和混合速度慢的特点,现在通过实验展示我们用神经采样 CD 和神经采样 PCD 算法对这些推理缺陷的有效解决. 特别需要注意的是,由于神经采样和吉布斯采样很相似,所以我们假设用神经采样代替吉布斯采样训练出的 RBM 在任务中最终的重构误差或准确率并不会更好. 神经采样的真正优势在于将来能在类脑硬件上实现概率推理. 实验只用 MNIST 数据集来小规模训练 RBM,和通过重构误差指标展示我们的算法有效消除神经采样的推理缺陷对 RBM 训练的负面影响. MNIST 数据集为 28×28 像素的手写数字图片. 将每个像素映射到 0 和 1 之间,比如 s_i ,那么对应的可见单元取 1 的概率为 s_i ,取 0 的概率为 $1 - s_i$. 实验中我们用训练集中第 1~20 张图片作为重构检验的检验集,第 21~10020 张图片作为训练样本.

在实验中我们的参数设置如下:所有实验中,一个子集包含 10 张图片,学习率都是从 0.1 线性下降到 0. 在神经采样的神经元模型中,迟滞时间设置为 $\tau=20$, τ 的值越大神经采样的对初始化状态敏感和混合速度慢的现象越明显. 神经采样 CD 算法的实验用 700 个隐单元,神经采样 PCD 算法中用 1000 个隐单元.

8.3.1 神经采样 CD 算法

我们比较在用神经采样替代吉布斯采样后,不恰当初始化 CD 算法的隐层状态以及采用 8.1 节提出的恰当初始化方式对 RBM 训练过程中重构误差曲线变化的影响.“不恰当初始化”指在重构前用均匀分布初始化隐层采样神经元的内部状态.“恰当初始化”指的是在均匀初始化隐层后,对隐层用神经采样做 50 步推理,每步都遍历隐层一次. 训练中重构误差(取 5 次实验的均值)的变化和方差如图 5 所示.

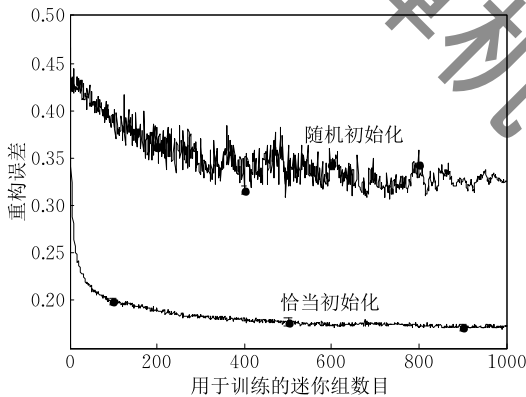


图 5 用神经采样代替吉布斯采样后,恰当初始化和随机初始化隐层状态的 CD 算法在训练过程中的重构曲线(表示方差的误差棒很小)

重构误差取检验集中原图片和吉布斯重构图片间的像素值均方差. 最后用神经采样 CD 算法训练出来的 RBM 在测试集上做吉布斯重构,从而评估其重构性能. 图中明显观察到,不恰当初始化隐层状态无法有效训练 RBM,训练过程中重构误差远远大于经过恰当初始化(即神经采样 CD 算法)时的误差. 不恰当的初始化隐含地增加了在扩大空间中环境分布和模型分布之间的距离,所以少数几步重构无法马上使重构分布比环境分布离模型分布更近. 此时即使当 $p(v) = q(v)$, 一步重构后的分布 $q_{\text{recon}}(v) \neq q(v)$ ——不恰当的初始化会将隐空间状态取一个不相关的值. 图 6 显示用神经采样 CD 算法训练出来的 RBM 在 MNIST 测试集上重构效果.



图 6 用神经采样 CD 算法训练出来的 RBM 在测试集上做吉布斯重构. 第 1 行为重构结果,第 2 行为 MNIST 测试集中的原图

8.3.2 神经采样 PCD 算法

我们通过改变每次参数更新时用神经采样做 MCMC 的步数,观察到如下现象:脉冲神经网络动力学中的迟滞效应导致了神经采样混合速度较慢,在步数很少的时候,训练过程不稳定;随着步数的增加,训练过程趋于稳定,并恢复与用吉布斯采样做 MCMC 相似的效果. 设每次参数更新前持续马尔可夫链的样本做 m 步 MCMC,我们比较了用 $m=1, 5$ 的神经采样 PCD 算法和 $m=1$ 的吉布斯采样 PCD 算法训练 RBM 训练过程中的重构误差. 每种情况都做了 5 次实验来计算重构误差的均值与方差. 3 种 PCD 算法均在持续马尔可夫链中取了 40 个样本粒子,初始状态是内部状态的均匀分布. 如图 7 所示,在神经采样 PCD 算法中, m 越小我们能观察到越严重的波动,训练过程不稳定. 和单步吉布斯采样相比,单步神经采样的混合速度明显更慢. 但通过增加 MCMC 步数, m 步神经采样恢复和单步吉布斯采样的混合度,训练过程的稳定性相近.

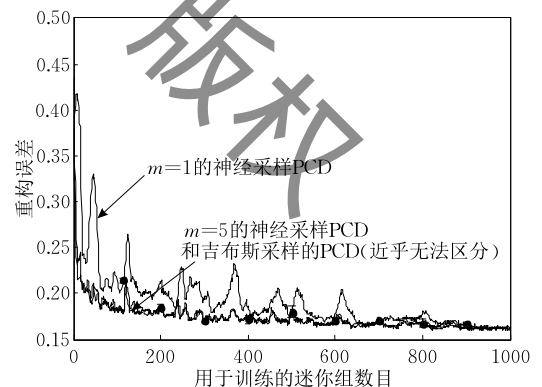


图 7 神经采样 PCD 训练过程中重构误差曲线随 MCMC 步数增加而趋于稳定(表示方差的误差棒很小)

但即便如此, $m=1$ 时神经采样的混合速度仍足够使学习越来越好,而且随着迭代次数的增多, $m=1$ 的神经采样和 $m=5$ 的神经采样或吉布斯采样之间的重构误差差距亦越来越小. 图 8 展示了神经采样 PCD 训练出来的 RBM 有很好的重构效果.



图 8 神经采样 PCD 算法训练出来的 RBM 用于在 MNIST 测试集上做吉布斯重构(从上到下分别表示 $m=1,5$ 的神经采样 PCD)

9 结论与展望

在神经采样方法出现以前,脉冲神经网络都是基于信念传递方法(belief propagation)对任意概率图模型做推理^[17-21]. 除文献[17]以外,这些网络都用放电频率编码信息,神经采样的脉冲时刻编码比用放电频率编码更能发挥类脑硬件上事件驱动的计算特点,所以是目前最适合在类脑硬件上实现的推理方法. 而且现在神经采样已经能够对任意贝叶斯网络做推理(见第 5 节“用神经采样在任意贝叶斯网络中做推理”);脉冲时刻编码方式能和 STDP 结合起来以事件驱动的形式训练生成式模型(见第 7 节“事件驱动的 CD 算法”及文献[22]). 这些进展都为在类脑硬件上实现概率模型的推理和学习提供了广阔前景.

但一些概率模型需要对连续随机变量做推理,比如 LDA(Latent Dirichlet Allocation^[23])里主题分布参数是概率单纯形上的样本,因为神经采样只能编码离散随机变量,所以无法直接对连续随机变量采样. 一种在脉冲神经网络中对连续随机变量推理的方法是突触采样^[24]. 突触采样假设突触的演化能实现 SGLD(Stochastic Gradient Langevin Dynamics^[25],一种在连续状态空间中采样的方法)随机概率推理. 这个方法不但能够解释大脑中突触的随机演化现象,而且有望和神经采样结合起来在类脑硬件上实现同时对离散和连续随机变量采样.

移动设备上概率方法的应用往往需要快速做推理,例如无人飞机对自身当前位置的估计. 但神经采样不但需要预热时间,而且需要足够采样时间来准确地估计统计量,在实用中的有效推理速度较慢. 文献[26]用一群独立而相同的脉冲神经元同时对一个随机变量做采样,能够保证在迅速变化的环境中统计量仍然被快速准确地估计. 而且用一群神经元表示一个变量能使类脑硬件有更高容错性. 这种群体

神经元的编码方法仍然有待与神经采样方法结合起来提高脉冲网络推理的速度和容错能力.

总的来讲,神经采样是目前最有希望在类脑硬件上实现的概率推理及训练概率模型方法之一. 但这个方法在实际应用中仍然面对诸如只能对离散随机变量推理和推理速度慢等问题. 如何在具体模型中结合其他已有的脉冲神经网络方法来解决这些问题将会是未来研究工作的重心.

参 考 文 献

- [1] Russell S, Novig P. Artificial Intelligence: A Modern Approach. 3rd Edition. Upper Saddle River: Prentice Hall, 2009
- [2] Wainwright M, Jordan M I. Graphical models, exponential families, and variational Inference. Foundations and Trends in Machine Learning, 2008, 1(1-2): 1-305
- [3] Gelman A, Carlin J, Stern H, et al. Bayesian Data Analysis. 3rd Edition. BocaRaton, FL, USA: Chapman&Hall/CRC, 2013
- [4] Bishop C M. Pattern Recognition and Machine Learning. New York: Springer, 2006
- [5] Buessing L, Bill J, Nessler B, Maass W. Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons. PLoS Computational Biology, 2011, 7(11): e1002211
- [6] Pecevski D, Buessing L, Maass W. Probabilistic inference in general graphical models through sampling in stochastic networks of spiking neurons. PLoS Computational Biology, 2011, 7(12): e1002294
- [7] Neftci E, Das S, Pedroni B, et al. Event-driven contrastive divergence for spiking neuromorphic systems. Frontiers in Neuroscience, 2014, 7: 74-87
- [8] Merolla P A, Arthur J V, Alvarez-Icaza R, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. Science, 2014, 345(6197): 668-73
- [9] Hinton G E. Training products of experts by minimizing contrastive Divergence. Neural Computation, 2002, 14(8): 1771-800
- [10] Tieleman T. Training restricted boltzmann machine using approximation to likelihood gradient//Proceedings of the 25th Annual International Conference on Machine Learning. Helsinki, Finland, 2008: 1064-71
- [11] Tieleman T, Hinton G. Using fast weights to improve persistent contrastive divergence//Proceedings of the 26th Annual International Conference on Machine Learning. New York, USA, 2009: 1033-40

- [12] Smolensky P. Information processing in dynamical systems; Foundations of harmony theory//Rumelhart D E, McClelland J L eds. Parallel Distributed Processing Volume 1. Cambridge: MIT Press, 1986: 194-281
- [13] Geman S, Geman D. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. IEEE Transactions on Pattern Analysis Machine Intelligence, 1984, 6(6): 721-741
- [14] Hinton G E, Sejnowski T J. Learning and relearning in boltzmann machines//Rumelhart D E, McClelland J L eds. Parallel Distributed Processing Volume 1. Cambridge: MIT Press, 1986: 282-317
- [15] Probst D, Petrovici M A, Bytschok I, et al. Probabilistic inference in discrete spaces can be implemented into networks of LIF neurons. Frontiers in Computational Neuroscience, 2015, 9(13): 1-11
- [16] Hinton G E. A practical guide to training restricted boltzmann machines. Department of Computer Science, University of Toronto; Technical Report 2010-003, 2010
- [17] Deneve S. Bayesian spiking neurons I: Inference. Neural Computation, 2008, 20(1): 91-117
- [18] Boerlin M, Deneve S. Spike-based population coding and working memory. PLoS Computational Biology, 2011, 7(2): e1001080
- [19] Steiner A, Mass W, Douglas R. Belief propagation in networks of spiking neurons. Neural Computation, 2009, 21(9): 2502-23
- [20] Litvak S, Ullman S. Cortical circuitry implementing graphical models. Neural Computation, 2009, 21(11): 3010-56
- [21] Bobrowski O, Meir R, Eldar Y C. Bayesian filtering in spiking neural networks; noise, adaptation, and multisensory integration. Neural Computation, 2009, 21(5): 1277-320
- [22] Nessler B, Pfeiffer M, Buesing L, Maass W. Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. PloS Computational Biology, 2013, 9(4): e1003037
- [23] Blei D, Ng A, Jordan M. Latent dirichlet allocation. Journal of Machine Learning Research, 2003, 3: 993-1022
- [24] Kappel D, Habenschuss R, Legenstein R, Maass W. Synaptic sampling: A bayesian approach to neural network plasticity and rewiring//Proceedings of the Advances in Neural Information Processing Systems, Montreal, Canada, 2015: 370-78
- [25] Welling M, Teh Y W. Bayesian learning via stochastic gradient langevin dynamics//Proceedings of the 28th International Conference on Machine Learning. Bellevue, USA, 2011: 681-88
- [26] Legenstein R, Maass W. Ensembles of spiking neurons with noise support optimal probabilistic inference in a dynamically changing environment. PLoS Computational Biology, 2014, 10(10): e1003859

附 录.

证明 $\forall g(s|\mathbf{v}), KL(p(\mathbf{v})g(s|\mathbf{v})\|q(s,\mathbf{v})) \geq KL(p(\mathbf{v})\|q(\mathbf{v}))$, 等号成立当且仅当 $g(s|\mathbf{v})=q(s|\mathbf{v})$.

证明.

$$\begin{aligned} & KL(p(\mathbf{v})g(s|\mathbf{v})\|q(s,\mathbf{v})) \\ &= \sum_{\mathbf{v},s} p(\mathbf{v})g(s|\mathbf{v}) \left[\log \frac{p(\mathbf{v})}{q(\mathbf{v})} + \log \frac{g(s|\mathbf{v})}{q(s|\mathbf{v})} \right] \end{aligned}$$

$$\begin{aligned} &= \sum_{\mathbf{v}} p(\mathbf{v})g(s|\mathbf{v}) \log \frac{p(\mathbf{v})}{q(\mathbf{v})} + \sum_s g(s|\mathbf{v}) \log \frac{g(s|\mathbf{v})}{q(s|\mathbf{v})} \\ &= KL(p(\mathbf{v})\|q(\mathbf{v})) + KL(g(s|\mathbf{v})\|q(s|\mathbf{v})) \\ &\geq KL(p(\mathbf{v})\|q(\mathbf{v})). \end{aligned}$$

等号成立当且仅当

$$g(s|\mathbf{v})=q(s|\mathbf{v}).$$

证毕.



XIAO Zi-Hao, born in 1992, M. S. candidate. His research interests include Bayesian methods and spiking neural networks.

ZHU Jun, born in 1983, Ph. D., associate professor. His research interests include artificial intelligence and machine learning.

Background

Neural sampling is an important new method in answering the research question of using spiking neural network dynamics to do probabilistic inference. This research question is recently

focused by computer scientists, neuroscientists, and hardware engineers. As this review paper introduces, neural sampling not only has a rigorous theory based on MCMC, but also has

been successfully applied to do sampling from Bayesian networks, and model a RBM training process on neuromorphic hardware. As a dynamics to do sampling, we also discuss and show that neural sampling is sensitive to initialization and has slow mixing rate in this paper. As a concrete example, we fix their negative effects on training RBM.

This research topic of understanding how to use neural sampling to do inference in a machine learning model is a first step towards using neuromorphic hardware to solve intelligence problem. Statistical machine learning methods are powerful, but are usually energy consuming on conventional computers, which limits their applications on mobile devices. On the other hand, neuromorphic hardware is very energy friendly but how to use the spiking neural network dynamics inherited in the hardware to do inference for probabilistic models has

remained an open problem. The solutions to this open problem have the potential to greatly change the application of intelligence systems in industry such as mobile phone, automobile and aircraft. The neural sampling method, with its spike-timing encode characteristic, is closer to the solution than many other existing spiking neural networks that do inference. This paper provides a landscape of the existing theory and applications of neural sampling, and a perspective on the problems needed to solve in the future.

The work is supported by the National Basic Research Program (973 Program) of China (No. 2013CB329403), the National Natural Science Foundation of China (Nos. 61322308, 61332007), the Tsinghua National Laboratory for Information Science and Technology Big Data Initiative, and Tsinghua Initiative Scientific Research Program (No. 20141080934).

《计算机学报》
版权