基于忆阻循环神经网络的层次化状态正则变分自编码器

胡小方* 杨 涛
(西南大学人工智能学院 重庆 400715)
(类脑计算与智能控制重庆市重点实验室 重庆 400715)

摘 要：变分自编码器(VAE)作为一个功能强大的文本生成模型受到越来越多的关注。然而，变分自编码器在优化过程中容易出现后验崩溃，即忽略潜在变量，退化为一个自编码器。针对这个问题，该文提出一种新的变分自编码器模型，通过层次化编码和状态正则方法，可以有效缓解后验崩溃，且相较于基线模型具有更优的文本生成质量。在此基础上，基于纳米级忆阻器，将提出的变分自编码器模型与忆阻循环神经网络(RNN)结合，设计一种基于忆阻循环神经网络的硬件实现方案，即层次化变分自编码忆组神经网络(HVAE-MNN)，探讨模型的硬件加速。计算机仿真实验和结果分析验证了该文模型的有效性与优越性。

关键词：变分自编码器；忆阻器；忆阻循环网络；文本生成
中图分类号：TN918.3; TN601 **文献标识码**：A **文章编号**：1009-5896(2023)02-0689-09
DOI: 10.11999/JEIT211431

Hierarchical State Regularization Variational AutoEncoder Based on Memristor Recurrent Neural Network

HU Xiaofang YANG Tao
(*College of Artificial Intelligence*, *Southwest University*, *Chongqing* 400715, *China*)
(*Brain-inspired Computing & Intelligent Control of Chongqing Key Laboratory*, *Chongqing* 400715, *China*)

Abstract: As a powerful text generation model, the Variational AutoEncoder(VAE) has attracted more and more attention. However, in the process of optimization, the variational auto-encoder tends to ignore the potential variables and degenerates into an auto-encoder, called a posteriori collapse. A new variational auto-encoder model is proposed in this paper, called Hierarchical Status Regularisation Variational AutoEncoder (HSR-VAE), which can effectively alleviate the problem of posterior collapse through hierarchical coding and state regularization and has better model performance than the baseline model. On this basis, based on the nanometer memristor, the model is combined with the memristor Recurrent Neural Network (RNN). A hardware implementation scheme based on a memristor recurrent neural network is proposed to realize the hardware acceleration of the model, which called Hierarchical Variational AutoEncoder Memristor Neural Networks (HVAE-MHN). Computer simulation experiments and result analysis verify the validity and superiority of the proposed model.

Key words: Variational AutoEncoder(VAE); Memristor; Memristor recurrent network; Text generation

1 引言

变分自编码器(Variational AutoEncoder, VAE[1])和其他深度生成模型，如生成对抗网络[2]和自回归模型[3]等，都可以从复杂且高维的未标记数据中学习到相应的信息。其中VAE广泛应用于图像处理[4, 5]和自然语言处理任务[6–9]。

然而，VAE在优化过程中常常会出现后验崩溃，又称为KL散度(Kullback–Leibler Divergence, KLD)消失[10]，即在生成过程中，模型忽略变分自编码器的潜在变量信息，退化为一个自编码模型。由于循环神经网络(Recurrent Neural Network, RNN)自身的强自回归性，使得基于循环神经网络的变分自编码器更容易出现这种现象。针对这一问题，研究人员陆续提出多种解决方案[10–12]。在最近的研究中，Shen等人[13]利用多层卷积神经网络替代

收稿日期：2021-12-06；改回日期：2022-03-03；网络出版：2022-03-12
*通信作者： 胡小方 huxf@swu.edu.cn
基金项目：国家自然科学基金(61976246)，重庆市自然科学基金(cstc2020jcyj-msxmX0385)
Foundation Items: The National Natural Science Foundation of China (61976246), The Natural Science Foundation of Chongqing (cstc2020jcyj-msxmX0385)

编码器并用循环网络作为解码器；Hao等人[14]使用循环模拟退火方法来缓解KL散度消失；He等人[15]提出一个滞后推理网络，在解码器更新之前多次更新编码器，从动力学的角度避免该问题；Zhu等人[16]将批量归一化(Batch Normalization, BN)正则应用于VAE的近似后验概率的参数中，确保KL值为正值；Li等人[17]对编码器中的隐变量施加KL正则，缓解后验崩溃的问题；Pang等人[18]提出一种新的推理方法，在VAE模型的后验分布的指导下运行一定次数的朗之万动力学(Langevin dynamics)算法，从而有效避免模型崩溃的问题。然而，这些模型大多集中于缓解VAE后验崩溃的问题，而忽略了模型预测性能。

RNN是一种广泛研究的具有信息反馈的神经网络模型，与前馈神经网络相比，RNN融合了时间序列的概念，保持了对时间序列的长期依赖性，并且对时间序列场景具有良好的建模能力，然而，在文本生成过程中，当文本序列过长时，RNN模型会发生梯度消失的现象。为解决这个问题，提出长短期记忆神经网络(Long Short-Term Memory, LSTM)，LSTM通过控制模型内部的遗忘门在一定程度上抑制RNN模型的梯度消失的问题，并在较长时间内保持了信息依赖性。随着LSTM模型的发展，其显著增加的复杂度和不断增长的参数量，使得基于互补金属氧化物半导体(Complementary Metal Oxide Semiconductor, CMOS)器件实现的LSTM网络，在计算方面表现出一些不足之处。

忆阻器是一种二端口“记忆电阻”，能够在存储信息的地方进行计算，这种存算一体化的特点减少了存储和计算之间传输数据的需求。与传统的基于CMOS器件的实现方案相比，基于忆阻器的人工神经网络具有体积小、功耗低、集成度高等特点。忆阻器已经被应用于许多人工神经网络硬件部署，包括单层或多层神经网络[19]、卷积神经网络(Convolutional Neural Networks, CNN)[20]和LSTM[21]等。其中，Adam等人[22]提出了一种用于时间序列预测的忆阻LSTM；Gokmen等人[23]将LSTM功能模块映射到忆阻交叉阵列中，并探索了器件缺陷对模型性能的影响；Li等人[24]展示了LSTM网络核心模块的忆阻器硬件实现，并采用两个1T1M的方式来表示正负权值；Liu等人[25]在LSTM的硬件实现上提出一种新的权值更新方案，实现在线训练，并对忆阻器的电导值实现并行更新。

本文针对VAE后验崩溃的问题，提出一种新的变分自编码器模型，称为层次化状态正则变分自编码器(Hierarchical Status Regularisation Variational AutoEncoder, HSR-VAE)。HSR-VAE不但可以有效缓解后验崩溃的问题，且较于基线模型，拥有更好的文本生成质量。与现有的变分自编码器仅在最后的时间步状态下施加KL正则[17]，或者仅仅是通过分层的思想对隐藏状态矩阵进行细化处理[26]不同，HSR-VAE在层次化状态方法的基础上引入时间步状态正则的方法，通过层次化方法对隐藏状态矩阵进行细化处理，并且对各个时间步的隐藏细化状态值施加KL正则，两种方法的结合可以有效缓解VAE的后验崩溃问题，明显提升模型预测能力。同时，为提高HSR-VAE模型的计算效率，本文在忆阻循环网络的基础上，将HSR-VAE部署在忆阻交叉阵列中，提出HSR-VAE的硬件加速方案，即层次化变分自编码忆阻神经网络(Hierarchical Variational AutoEncoder Memristor Neural Networks, HVAE-MNN)。通过忆阻器存算一体的特性，明显提升HSR-VAE模型的计算效率。

为了证明本文方法的有效性，本文加入一些强基线模型进行对比，并基于4个公共数据集，分别在语言模型和对话响应生成任务上进行实验对比。语言模型任务中， HSR-VAE可有效缓解后验崩溃，且在定量分析负对数似然(Negative Log Likelihood, NLL)和困惑度(PerPlexity Loss, PPL)的平均实验结果表明，较于基线模型，NLL值降低6，PPL值降低5.9，KL值提高5.6；对话响应生成任务中，多样性评估指标Intra-dist1和Inter-dist1分别提升5.6%和20.4%。

综上所述，本文贡献如下：

(1) 提出一种新的变分自编码器模型HSR-VAE，有效缓解变分自编码器后验崩溃的问题。

(2) 提出一种层次化状态正则的方法。在层次化状态的基础之上引入时间步状态正则的方法，明显提升模型预测性能。

(3) 设计一种基于忆阻循环神经网络的变分自编码器硬件实现方案HVAE-MNN，为变分自编码器的硬件加速提供一种新的思考。

2 层次化状态正则变分自编码器

2.1 变分自编码器

变分自编码器是一种基于隐空间的生成模型，旨在通过解码隐变量生成相应数据。对于数据集$X=\{x_i\}_{i=1}^N$，VAE生成过程如下：(1)通过先验分布$P_\theta(\boldsymbol{z})$采样出模型隐变量$\boldsymbol{z}$(2)，通过后验分布$P_\theta(x_i|\boldsymbol{z})$生成数据$x_i$。由于生成隐变量$\boldsymbol{z}$需要计算后验分布$P_\theta(x_i|\boldsymbol{z})$，但后验分布$P_\theta(x_i|\boldsymbol{z})$难以直接计算，所以VAE模型构造$Q_\phi(\boldsymbol{z}|x_i)$来近似真实后验分布$P_\theta(x_i|\boldsymbol{z})$。VAE模型的损失函数是带正则项的负对数似然函数，损失函数为

$$l_i(\theta,\phi) = -E_{\boldsymbol{z}\sim p_\theta(\boldsymbol{z}|x_i)}\left[\log_{Q_\phi}(x_i|\boldsymbol{z})\right] + \mathrm{KL}(p_\theta(\boldsymbol{z}|x_i)||p(\boldsymbol{z})) \tag{1}$$

其中，第1项是重构损失，目的是让生成数据和原始数据尽可能相近，第2项KL散度是正则项，它衡量了两个分布的近似程度。基本的VAE-RNN模型遵循式(1)。由于编码器是RNN，所以隐变量$\boldsymbol{z}$是在最后一个时间步的隐藏状态值中采样得到的，将该隐变量作为解码器的输入。VAE-RNN模型的损失函数为

$$l_i(\theta,\phi) = -E_{z\sim p_\theta(\boldsymbol{z}^T|x_i)}\left[\log_{Q_\phi}(x_i|\boldsymbol{z}^T)\right] + \mathrm{KL}(p_\theta(\boldsymbol{z}^T|x_i)||p(\boldsymbol{z}^T)) \tag{2}$$

其中，总时间步长T即为输入句子的长度。对公式分析可知，在优化过程中，当$Q_\phi(\boldsymbol{z}^T|x_i)=P(\boldsymbol{z}^T)$接近全局最小值时，$\boldsymbol{z}^T$和$x_i$即为两个独立变量，因而使得解码器无法从$\boldsymbol{z}^T$中学习到对应信息，VAE退化为一个自编码器模型，即后验崩溃。

2.2 结合层次化和时间步正则的变分自编码器

针对VAE后验崩溃，时间步正则变分自编码器(Time step-Wise Regularisation Variational AutoEncoder, TWR-VAE)[17]对编码器的所有时间步的隐藏状态值施加标准正态分布KL正则。TWR-VAE虽然有效缓解后验崩溃，但与批量归一化变分自编码器(Batch Normalization Variational AutoEncoder, BN-VAE)[16]相比，KL值相对较低，针对这一问题，本文提出层次化状态正则变分自编码器HSR-VAE。HSR-VAE通过层次化方法编码隐藏状态矩阵，并且对编码后的隐藏状态矩阵各个时间步的状态值施加KL正则。

HSR-VAE模型结构如图1所示。HSR-VAE的编码器由两层LSTM网络组成，解码器为单层LSTM网络。在t时刻的编码过程中，本文模型采用两层LSTM网络的方式，进一步处理隐藏状态矩阵$\boldsymbol{h}_1^t$，输出一个新的隐藏状态矩阵$\boldsymbol{h}_2^t$。整个编码过程为

$$\boldsymbol{h}_1^t = f_\theta^{\mathrm{enc1}}(x_i);\boldsymbol{h}_2^t = f_\theta^{\mathrm{enc2}}(\boldsymbol{h}_1^t);\mu^t,\varepsilon^t = \mathrm{MLP}_\theta(\boldsymbol{h}_2^t) \tag{3}$$

HSR-VAE模型的编码流程：输入文本数据$X(x_1,x_2,\cdots,x_T)$，通过第1层LSTM网络输出t时刻第1层LSTM网络隐藏状态矩阵$\boldsymbol{h}_1^t$，将该矩阵作为第2层LSTM网络的输入，输出t时刻第2层LSTM网络隐藏状态矩阵$\boldsymbol{h}_2^t$。为便于计算，采用全连接的方式从$\boldsymbol{h}_2^t$中采样出该状态矩阵在t时刻的均值μ^t和方差ε^t，通过均值μ^t和方差ε^t的重参数化，得到隐变量$\boldsymbol{z}^t$。

在解码过程中，将隐变量$\boldsymbol{z}^t$作为解码器的输入，输出$t+1$时刻的预测值x_{t+1}，如式(4)所示

$$x_{t+1} = f_\phi^{\mathrm{dec}}(\boldsymbol{z}^t) \tag{4}$$

其中，为了对每个时间步t的隐藏状态值施加KL正则，t时刻的KL正则如式(5)所示

$$D_{\mathrm{KL}}^t = -\frac{1}{2}\sum\left(1+\varepsilon^t+(\mu^t)^2-\mathrm{e}^{\varepsilon^t}\right) \tag{5}$$

其中，μ^t和ε^t为t时刻通过编码器计算得到的隐藏状态矩阵的均值和方差。

本文模型通过层次化计算隐藏状态值并采用式(5)的计算方式对每个时间步下隐藏状态值施加KL正则，最终HSR-VAE模型损失函数定义如式(6)所示

$$l_i(\theta,\phi)_{\text{HSR - VAE}} = -E_{\boldsymbol{z}\sim p_\theta(\boldsymbol{z}^T|x_i)}\left[\log_{Q_\phi}(x_i|\boldsymbol{z}^T)\right] + \frac{1}{T}\sum_{t=1}^{T}D_{\mathrm{KL}}(p_\theta(\boldsymbol{z}^T|x_i)||p(\boldsymbol{z}^T)) \tag{6}$$

其中，T为输入序列的长度，θ和ϕ分别是编码器和解码器的参数。

在模型优化过程中，采用重参数法采样隐变量$\boldsymbol{z}^T$，使得隐变量可微，便于梯度回传。在参数θ和ϕ的优化过程中通过蒙特卡罗(Monte Carlo, MC)方法获得θ和ϕ的无偏梯度。模型梯度计算如式(7)所示

$$\nabla_{\theta,\phi}l(\theta,\phi)_{\text{HSR - VAE}} \simeq \frac{1}{M}\sum_{m=1}^{1}\nabla_{\theta,\phi}\left(\log P_\theta\left(x_i|\boldsymbol{z}_m^t - \frac{1}{T}\sum_{t=1}^{T}\log\frac{Q_\phi(\boldsymbol{z}_m^t|x_i^{1:t})}{P(\boldsymbol{z}_m^t)}\right)\right),\quad \boldsymbol{z}_m^t = Q_\phi(\boldsymbol{z}_m^t|x_i^{1:t}) \tag{7}$$

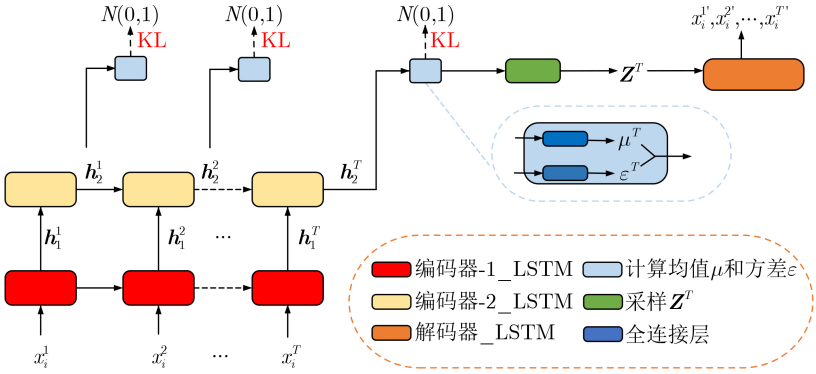


图 1 HSR-VAE模型结构图

其中，M表示从近似后验分布$Q_\phi(\boldsymbol{z}_m^t|x_t^{1:t})$随机抽取$\boldsymbol{z}_m^t(m\in[1:M])$的次数。

3 基于忆阻神经网络的层次化变分自编码器

3.1 忆阻器

1971年，文献[27]在研究电荷、电流、电压和磁通量之间的关系时，定义了磁通量和电荷之间的关系，提出忆阻器的概念。忆阻器是一种有记忆功能的非线性电阻，通电时可以通过改变流过它的电荷数量或磁通量来改变阻值，断电时保持当前阻值不变。2008年，惠普实验室设计出一个能工作的忆阻器物理模型，一个典型的惠普$\mathrm{Pt/TiO_2/Pt}$忆阻器数学模型[28]如式(8)所示

$$v(t)=i(t)\cdot R(t);R(t)=R_{\mathrm{on}}x(t)+R_{\mathrm{off}}(1-x(t));\quad x(t)=\frac{w(t)}{D} \tag{8}$$

其中，$R(t)$表示忆阻器的阻值，R_{on}和R_{off}分别表示忆阻器的最小和最大的阻值。$w(t)$表示掺杂层厚度，$x(t)$表示内部状态变量，D表示为忆阻器的厚度。

本文采用Ag/AgInSbTe/Ta(AIST)忆阻器模型，其内部状态变量描述为

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t}=\begin{cases}u_v\dfrac{R_{\mathrm{on}}}{D^2}\dfrac{\mathrm{i_{off}}}{i(t)-\mathrm{i_0}}f(x(t)), & 0<V_{\mathrm{th}}^{+}<v(t)\\ 0, & V_{\mathrm{on}}\le v(t)\le V_{\mathrm{off}}\\ u_v\dfrac{R_{\mathrm{on}}}{D^2}\dfrac{i(t)}{\mathrm{i_{on}}}f(x(t)), & v(t)<V_{\mathrm{th}}^{-}<0\end{cases};\ f(x(t))=1-(2x(t)-1)^{2p} \tag{9}$$

其中，u_v是平均离子迁移率，V_{th}^{+}和V_{th}^{-}分别为正阈值电压和负阈值电压，$\mathrm{i_{on}},\mathrm{i_{off}},\mathrm{i_0}$都为电流常数。$f(x(t))$是窗函数，$P$为正整数。

3.2 HSR-VAE硬件部署设计

本文模型HSR-VAE的硬件部署设计方案HVAE-MNN通过忆阻交叉阵列实现。本模型由3层LSTM网络组成，所以重点介绍基于忆阻LSTM的HSR-VAE硬件实现方案。

LSTM网络的关键组成为3个门控单元，即输入门、输出门和遗忘门。LSTM利用独特的门控单元对序列数据进行学习和选择性记忆，保持长距离的时间序列信息相关性，实现高精度预测。其中，输入门主要处理输入数据，遗忘门决定当前神经元对历史信息的记忆程度，输出门代表神经元的输出结果。输入文本序列$(x_1,x_2,\cdots,x_T)$，则t时刻，LSTM网络迭代公式为

$$i_t=S(\boldsymbol{w}_i\cdot[\boldsymbol{h}_{t-1},x_t]+\boldsymbol{b}_i);f_t=S(\boldsymbol{w}_f\cdot[\boldsymbol{h}_{t-1},x_t]+\boldsymbol{b}_f);\quad \boldsymbol{c}_t=f_t\cdot\boldsymbol{c}_{t-1}+i_t\cdot\tanh(\boldsymbol{w}_c\cdot[\boldsymbol{h}_{t-1},x_t]) \tag{10}$$

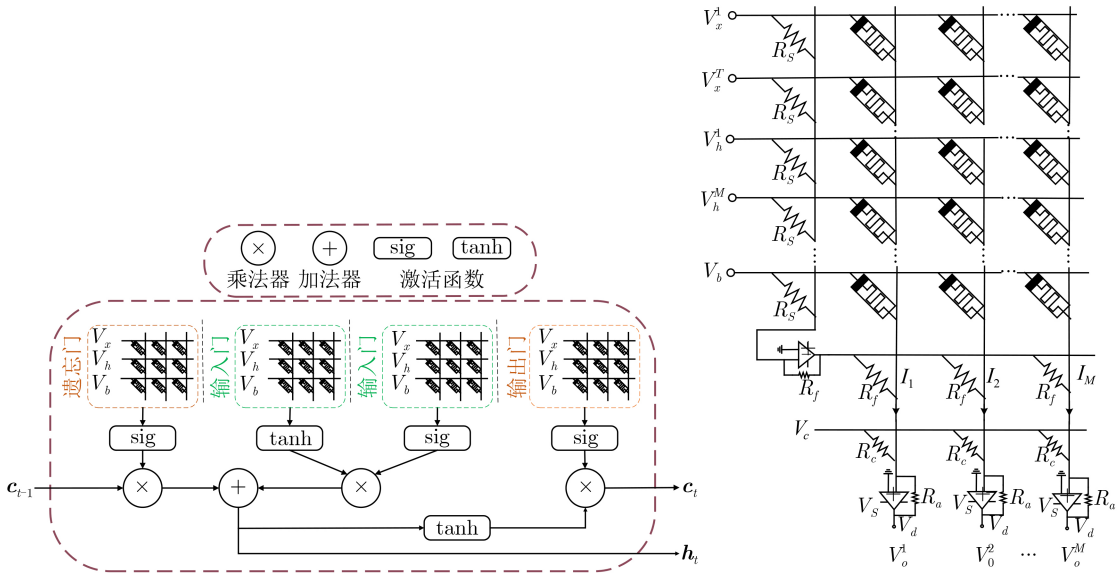
$$o_t=S(\boldsymbol{w}_o\cdot[\boldsymbol{c}_t,\boldsymbol{h}_{t-1},x_t]+\boldsymbol{b}_o);\boldsymbol{h}_t=o_t\cdot\tanh(\boldsymbol{c}_t) \tag{11}$$

其中，i_t, f_t和o_t分别表示t时刻的输入门、输出门和遗忘门的输入；x_t表示t时刻LSTM的输入序列，$\boldsymbol{h}_{t-1}$表示$t-1$时刻的隐藏层输出状态，$\boldsymbol{b}_i$, $\boldsymbol{b}_f$和$\boldsymbol{b}_o$分别是对应的偏移向量，$\boldsymbol{w}_i$, $\boldsymbol{w}_f$和$\boldsymbol{w}_o$表示对应的权重矩阵，$\boldsymbol{c}_t$表示t时刻LSTM网络记忆信息。S表示sigmoid激活函数。

对公式分析可知，在LSTM网络中，其核心计算模块为矩阵的乘累加计算。忆阻器具有可变电阻和记忆电阻状态的能力，是权值矩阵计算的理想器件。因此，在具体应用过程中，将LSTM网络中的权值计算过程映射到忆阻交叉阵列中，通过改变加载幅值相同的电压时间长短的方式完成输入向量与权值向量的乘累加计算，实现LSTM网络的硬件加速，提升计算效率。

LSTM网络的忆阻电路设计如图2(a)所示，假设文本序列的长度为$\boldsymbol{T}$，模型隐藏状态矩阵的维度为M。图2(a)中，将文本序列$X(x_1,x_2,\cdots,x_T)$转换成电压信号$V_x(v_x^1,v_x^2,\cdots,v_x^T)$，隐藏状态矩阵$\boldsymbol{H}(\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_M)$的电压信号为$V_h(v_h^1,v_h^2,\cdots,v_h^M)$，偏移向量$\boldsymbol{b}$转换成电压$V_b$。对图2(a)中LSTM电路结构分析，一个LSTM网络含有4个矩阵计算单元，每个矩阵计算单元将对应权值矩阵映射到忆阻交叉阵列中，变换输入电压v_x^t,v_h^t和v_b^t，计算忆阻交叉阵列输出电压。忆阻交叉阵列将输出电压通过数模转换(Analog-to-Digital Converter, ADC)的方式输入到线上模型HSR-VAE中，计算输出当前时刻的记忆信息值$\boldsymbol{c}_t$和隐藏状态值$\boldsymbol{h}_t$。其中，每个计算单元的参数量为$(T+M+1)M$，一个忆阻LSTM的参数量为$4(T+M+1)M$。

忆阻LSTM中单个忆阻交叉阵列矩阵计算单元如图2(b)所示，以遗忘门为例。模型将输入文本X、隐藏状态值H、偏移向量$\boldsymbol{b}$和前一时刻记忆信息$\boldsymbol{c}_{t-1}$转换成电压信号V_x, V_h, V_b和V_c，并将这些电压信号加载到忆阻交叉阵列矩阵计算单元中，通过忆阻交叉阵列的基尔霍夫电流定理和欧姆定律，以及存算一体的特性，实现矩阵的原位乘累加计算，最终输出电压信号V_o。在映射过程中，由于模型的隐藏状态取值范围是$(-1,1)$，而忆阻交叉阵列中的忆阻器电导都为正值，为完整描述模型权值的正负性，本文采用1个忆阻器和1个固定电阻并联的方式来表示具有正负号的网络权值。计算方法为



(a) 忆阻LSTM电路　　(b) 忆阻LSTM遗忘门电路计算单元

图 2 忆阻LSTM

$$I_k = \sum_j (1/R_s - G_{j,k}) V_{\mathrm{in},j} \tag{12}$$

其中，I_k为忆阻交叉阵列中第k列总的输出电流，$(1/R_s - G_{j,k})$表示模型映射到交叉阵列中的权值的大小。$1/R_s$表示固定电阻的电导，$G_{j,k}$代表第j个输入数据在第K列上忆阻器的电导值。$V_{\mathrm{in},j}$表示第j个输入电压，in表示输入类型是X, H或$\boldsymbol{b}$，其对应的j的取值范围是$(0,T)$, $(T,T+M)$, $(T+M, T+M+1)$, T和M分别是文本序列的长度和隐藏层的维度。

本文模型在忆阻交叉阵列的基础上，提出HSR-VAE硬件加速方案HVAE-MNN。HVAE-MNN忆阻电路由3层忆阻LSTM所组成，其中，两个忆阻LSTM组成模型编码器，单个忆阻LSTM组成模型解码器，每个LSTM硬件电路网络基于图2(a)所示的忆阻交叉阵列。在实际应用场景中，硬件加速计算流程包括：将训练好的模型权值矩阵映射到忆阻交叉阵列中，其输入数据转换为对应的电压信号，经过图2(a)所示的LSTM电路计算隐藏状态矩阵值，将该隐藏状态矩阵使用ADC信号转化器转换为数字信号；在软件层面上，计算该隐藏状态矩阵的均值和方差，重参数化构建$\boldsymbol{z}^t$隐变量矩阵，再将该隐变量矩阵通过DAC信号转换器转化为模拟信号，输入到解码器LSTM网络中，进行LSTM网络硬件加速计算，最后将输出转换为数字信号传给软件，计算预测值，并构建预测文本序列信息，最终，该文模型实现HSR-VAE模型的硬件加速。

4　实验结果分析

本文采用4个公共数据集来评估HSR-VAE，包括PTB, Yelp, Yahoo和Dailydialog。表1总结了相应的数据集信息。其中，PTB,Yelp和Yahoo数据集应用于语言模型任务，Dailydialog数据集应用于对话响应生成任务。本文模型词向量的大小为512维，隐藏层的大小均为256维。

4.1　语言模型

神经网络语言模型是在给定一个文本序列的前提下，预测下一个词出现的概率。本文采用两个评估指标，包括NLL和PPL来评价模型的预测性能，NLL和PPL值越低说明预测文本越合理；KL值来评估模型是否出现后验崩溃。通过实验，本文模型与强基线模型进行了对比分析。(1)VAE-LSTM[10]：基于LSTM网络的VAE模型，采用KL-annealing方法缓解后验崩溃；(2)半摊销变分自动编码器(Semi-Amortized Variational AutoEncoders, SA-VAE)[29]：采用随机变分推理初始化变分参数；(3)循环变分自动编码器(Cyclical Variational AutoEncoder, Cyc-VAE)[14]：采用周期性模拟退火方法缓解KL散

表 1　数据集

数据集	训练集	验证集	测试集	词表(k)
PTB	42068	3370	3761	9.95
Yelp	100000	10000	10000	19.76
Yahoo	100000	10000	10000	19.73
Dailydialog	11118	1000	1000	22

度消失；(4)滞后变分自动编码器(Lagging Variational AutoEncoder, Lag-VAE)[15]：采用多次更新编码器而较少更新解码器；(5)批量归一化变分自动编码器(Batch Normalization Variational AutoEncoder, BN-VAE)[16]：在KL分布中采用BN正则避免后验崩溃；(6)TWR-VAE[17]：对每个时间步的隐藏状态值进行KL正则；(7)短程推理变分自动编码器(Short Run Inference Variational AutoEncoder, Sri-VAE)[18]：将VAE与Langevin Dynamics算法结合避免后验崩溃。

语言模型实验结果如表2所示，HSR-VAE的预测性能(NLL, PPL)优于所有基线模型。对两个数据集的评估结果进行平均，与基线模型TWR-VAE相比，本文模型在NLL值降低6，PPL值降低5.9，KL值提高5.6；与强基线模型BN-VAE相比，KL值提升1.1；与最新模型Sri-VAE相比，NLL和PPL分别降低29.2和42.6。实验结果表明HSR-VAE在语言建模任务中优异的性能。语言模型生成文本如表3所示，原始文本序列与生成文本序列越相似，说明模型的预测性能越好。

消融研究测试TWR-VAE与HSR-VAE模型在RNN, LSTM和GRU等不同循环结构的实验结果。

表 2 语言模型实验对比

模型	PTB			Yahoo		
	NLL ↓	PPL ↓	KL	NLL ↓	PPL ↓	KL
VAE-LSTM	101.2	101.4	0.0	328.6	61.2	0.0
SA-VAE	101.0	100.7	1.3	327.2	60.2	5.2
Cyc-VAE	102.8	109.0	1.4	330.6	65.3	2.1
Lag_VAE	100.9	99.8	7.2	326.7	59.8	5.7
BN-VAE	100.2	96.9	7.2	327.4	60.2	8.8
Sri-VAE	101.2	94.2	**10.1**	327.3	57.0	**16.1**
TWR-VAE	86.6	40.9	5.0	317.3	50.2	3.3
本文	**79.4**	**30.2**	9.1	**290.7**	**35.8**	8.7

同时，为测量隐变量$\boldsymbol{z}^t$采样输入数据信息量，即测量输入数据与隐变量之间的互信息，增加一个互信息评估(Mutual Information, MI)。其中，MI的计算方法如式(13)所示

$$\begin{aligned} I(x,\boldsymbol{z}) =& E_x[D_{\mathrm{KL}}(Q_\phi(\boldsymbol{z}^T|x)||P(\boldsymbol{z}^T))] \\ &- D_{\mathrm{KL}}(Q_\phi(\boldsymbol{z}^T)||P(\boldsymbol{z}^T)) \end{aligned} \tag{13}$$

消融实验结果如表4所示，与TWR-VAE相比，HSR-VAE的评估指标NLL和PPL值有明显降低，表明HSR-VAE预测文本更加合理。同时，本文还探究不同循环网络结构组合的实验效果，HSR-VAE

表 3 语言模型生成文本示例

模型	原始文本	生成文本
TWR-VAE	(1) it 's totally different	(1) it 's very ok
	(2) sec proposals may n	(2) terms officials may n
	(3) the test may come today	(3) the naczelnik may be
本文	(1) merrill lynch ready assets trust	(1) merrill lynch ready assets trust
	(2) all that now has changed	(2) what that now has changed
	(3) now it 's happening again	(3) now it 's quite again

表 4 消融研究实验对比

模型		Yelp				Yahoo			
		NLL ↓	PPL ↓	MI ↑	KL	NLL ↓	PPL ↓	MI ↑	KL
TWR-VAE_RNN		395.4	56.4	3.9	0.5	363.0	88.2	4.1	0.6
TWR-VAE_GRU		360.9	39.7	4.2	3.3	336.9	63.9	4.2	3.7
TWR-VAE_ LSTM		344.3	33.5	4.1	3.1	317.3	50.2	4.1	3.3
本文	RNN + RNN	400.9	57.3	2.6	1.3	366.3	90.8	3.2	2.4
	RNN + LSTM	340.3	31.3	3.7	5.1	303.2	41.7	3.3	6.0
	RNN + GRU	358.6	37.4	3.6	3.1	326.5	56.2	4.9	3.9
	LSTM + RNN	349.7	34.2	3.3	6.3	310.8	46.7	3.4	6.7
	LSTM + LSTM	340.4	31.1	3.7	10.2	310.4	45.6	3.5	8.8
	LSTM + GRU	341.5	31.4	3.3	7.0	295.7	38.1	3.5	7.9
	GRU + RNN	349.8	34.2	4.6	10.6	320.8	52.8	4.8	**11.2**
	GRU + LSTM	342.5	31.7	3.4	**10.7**	293.7	37.1	3.5	11.1
	GRU + GRU	**336.6**	**29.9**	3.5	7.3	**290.7**	**35.8**	3.4	8.7

的KL值表明，相比于单层循环网络结构，双层循环网络结构可更加有效地避免VAE后验崩溃；互信息MI值表明，双层循环网络架构会减少解码器获得的信息量。低MI值和高KL值表明，弱化编码器采样性能有助于避免VAE后验崩溃。

4.2 对话响应生成

对话响应生成的任务目标是根据用户的话语生成有意义的响应，然而，建立在序列对序列模型基础上的对话响应生成往往会产生例如“好”“嗯”“谢谢”等一般性的回答。针对该问题，一种有效的解决方案是采用条件变分自编码器(Conditional Variational AutoEncoder, CVAE)[30]，该模型采样编码器中的句子级别多样性，通过隐变量来学习潜在会话意图的分布，有效改善响应的多样性问题。本文以CVAE的结构基础对HSR-VAE进行扩展，进一步评估模型在对话响应生成任务中的效果。扩展模型损失计算如式(14)所示

$$l(\theta,\phi)_{\mathrm{HSR-VAE}} = E_{Q_\phi(\boldsymbol{z}^J|x_i,c)}[\lg P_\theta(x_i|\boldsymbol{z}^J,c)] - \frac{1}{J}\sum_{j=1}^{J} D_{\mathrm{KL}}(Q_\phi(\boldsymbol{z}^j|x_i,c)||P_\theta(\boldsymbol{z}^j|c)) \tag{14}$$

其中，c表示上下文内容编码，J表示对话窗口的大小，j表示第几个对话窗口。$P_\theta(x_i|\boldsymbol{z}^J,c)$表示重构损失，$D_{\mathrm{KL}}(Q_\phi(\boldsymbol{z}^j|x_i,c)||P_\theta(\boldsymbol{z}^j|c))$表示KL散度，即通过$Q_\phi(\boldsymbol{z}^j|x_i,c)$来拟合真实后验分布$P_\theta(\boldsymbol{z}^j|c)$。

对话响应生成任务中，本文基于Dailydialog[31]数据集进行对比实验。训练过程中，对话窗口的大小J设置为10，最大对话长度为40，采用贪婪解码来抽样响应，使得对话随机性完全取决于隐变量。所有基线模型采用的超参数相同，编码器和解码器都采用GRU模型，模型的隐藏状态值维度设置为300，隐变量维度大小为200。

在对比实验中，本文模型除了与基线模型TWR-VAE[17]、Wasserstein自动编码器(Wasserstein AutoEncoder, WAE)[8]、CVAE、独立变分自动编码器(Independent Variational AutoEncoder, IVAE)[32]进行对比，还与层次化基线模型RNN (Variational Hierarchical Conversation RNNs, VHCR)[26]、可变分层循环编码器(Variable Hierarchical Recurrent Encoder-Decoder, VHRED)[33]、基于强化学习方法的Seq2Seq生成性对抗网络(Seq2Seq Generative Adversarial Networks, SeqGAN)[34]进行对比。对话响应生成任务评估指标采用先前已有工作所采用的评价方法。(1)双语评估替补(BiLingual Evaluation Understudy, BLEU)。该评估指标展示了生成对话与参考序列的匹配程度。对于每个测试情境，计算每个响应的BLEU分数，并将n元语法查准率和n元语法召回率分别定义为平均分和最高分；(2)BOW。该评估展示了模型生成的回答和参考序列之间的词袋嵌入余弦相似度。本文采用3种度量方式计算单词嵌入的相似度：BOW-G(BOW-Greedy)是通过贪婪匹配的两个对话单词之间的平均余弦相似度，BOW-A(BOW-Average)是单词嵌入之间的平均余弦相似度，BOW-E(BOW-Extreme)是两个对话的单词嵌入的最大极值之间的余弦相似度。(3)Distinct。该方法通过计算生成的对话响应中的唯一n元语法(n=1,2)与所有n元语法的比率来衡量生成的对话响应的多样性。Intra-dist表示单次情境中单个响应内部的多样性；Inter-dist表示单次情境中多个响应之间的多样性。

对话响应生成实验结果如表5所示。HSR-VAE在各个评估指标均优于层次化基线模型VHRED和VHCR，表明在层次化的基础上进行时间步状态正则可提升生成对话的质量；与基线模型TWR-VAE相比，HSR-VAE在一些评估指标上有一定的优化，特别是在多样性评估指标Intra-dist和Inter-dist，表明层次化优化方法可有效提升对话响应生成任务的多样性。表6展示了对话响应生成任务中生成的可能的响应文本。

表 5　对话响应生成任务实验对比

模型	BLEU			BOW			Intra-dist		Inter-dist	
	R ↑	P ↑	F1 ↑	A ↑	E ↑	G ↑	dist-1 ↑	dist-2 ↑	dist-1 ↑	dist-2 ↑
SeqGAN	0.270	0.270	0.270	0.918	0.495	0.774	0.747	0.806	0.075	0.081
VHRED	0.271	0.260	0.265	0.892	0.507	0.786	0.633	0.771	0.071	0.089
VHCR	0.289	0.266	0.277	0.925	0.525	0.798	0.768	0.814	0.105	0.129
CVAE	0.265	0.222	0.242	0.923	0.543	0.811	0.938	0.973	0.177	0.222
WAE	0.341	0.278	0.306	0.948	0.578	0.846	0.830	0.940	0.327	0.583
iVAE	0.355	0.239	0.285	0.951	**0.609**	**0.872**	0.897	0.975	0.501	0.868
TWR-VAE	**0.407**	**0.281**	**0.333**	**0.952**	0.603	0.865	0.921	**0.990**	0.497	0.817
本文	0.377	0.268	0.313	0.919	0.583	0.859	**0.977**	0.969	**0.701**	**0.915**

表 6 对话响应生成文本示例

TWR-VAE	本文
对话1 问题: come in, come in, what can i do for you ? 期望响应: professor donald, are you giving your advanced geology course again next semester?	
(1) i would like to have a look at your computer.	(1) who have been working.
(2) hello, i'd like to buy a ticket to shanghai.	(2) and a lot of coffee.
(3) hello, i'm looking for a job.	(3) some advice of humor.
对话2 问题: does the bathroom have a bath or a shower? 期望响应: all our bathrooms have full size <unk> with a shower attachment.	
(1) yes, sir. everything is available. and we will charge it.	(1) sure, we have two. the room is <unk> and <unk>. it's $ 60 for a while.
(2) perhaps i can use a credit card to the bank. what would you like?	(2) that's a reserved room, but it has a maximum account with us today.
(3) sure. does that mean will be there in your room?	(3) yes, that is the fee.

5 结束语

本文提出层次化状态正则变分自编码器HSR-VAE，本文模型通过层次化方法编码隐藏状态矩阵，并且对编码后的隐藏状态矩阵各个时间步的状态值施加KL正则。同时，基于忆阻交叉阵列完成LSTM网络核心模块的权值矩阵计算，通过线上线下混合训练及实时推理，实现HSR-VAE模型的硬件加速。计算机仿真结果实验表明，在语言建模任务中，HSR-VAE不仅可以有效避免后验崩溃，且拥有比所有强基线模型更好的性能；消融实验研究表明，层次化编码和时间步状态正则的有效结合可应用于不同循环结构的VAE，并有效提升模型性能；在对话响应生成任务中，HSR-VAE可有效提升对话响应生成序列的多样性。上述实验结果都表现出本文模型的有效性，进一步研究可以将HSR-VAE应用在其他任务，如机器翻译等。

参 考 文 献

[1] KINGMA D P and WELLING M. Auto-encoding variational bayes[C]. The 2nd International Conference on Learning Representations, Banff, Canada, 2014.

[2] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, *et al*. Generative adversarial networks[J]. *Communications of the ACM*, 2020, 63(11): 139–144. doi: 10.1145/3422622.

[3] VAN DEN OORD A, LI Yazhe, and VINYALS O. Representation learning with contrastive predictive coding[J]. arXiv: 1807.03748, 2018.

[4] RAZAVI A, VAN DEN OORD A, and VINYALS O. Generating diverse high-fidelity images with VQ-VAE-2[C]. The 33rd Conference on Neural Information Processing Systems, Vancouver, Canada, 2019.

[5] LI Xiao, LIN Chenghua, LI Ruizhe, *et al*. Latent space factorisation and manipulation via matrix subspace projection[C/OL]. The 37th International Conference on Machine Learning, 2020.

[6] LI Ruizhe, LI Xiao, LIN Chenghua, *et al*. A stable variational autoencoder for text modelling[C]. The 12th International Conference on Natural Language Generation, Tokyo, Japan, 2019. doi: 10.18653/v1/W19-8673.

[7] FANG Le, LI Chunyuan, GAO Jianfeng, *et al*. Implicit deep latent variable models for text generation[C]. The 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 2019. doi: 10.18653/v1/D19-1407.

[8] GU Xiaodong, CHO K, HA J W, *et al*. DialogWAE: Multimodal response generation with conditional wasserstein auto-encoder[C]. The 7th International Conference on Learning Representations, New Orleans, USA, 2019.

[9] JOHN V, MOU Lili, BAHULEYAN H, *et al*. Disentangled representation learning for non-parallel text style transfer[C]. The 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019. doi: 10.18653/v1/P19-1041.

[10] BOWMAN S R, VILNIS L, VINYALS O, *et al*. Generating sentences from a continuous space[C]. The 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 2016: 10–21.

[11] YANG Zichao, HU Zhiting, SALAKHUTDINOV R, *et al*. Improved variational autoencoders for text modeling using dilated convolutions[C]. The 34th International Conference on Machine Learning, Sydney, Australia, 2017: 3881–3890.

[12] XU Jiacheng and DURRETT G. Spherical latent spaces for stable variational autoencoders[C]. The 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 2018: 4503–4513.

[13] SHEN Dinghan, CELIKYILMAZ A, ZHANG Yizhe, *et al*. Towards generating long and coherent text with multi-level latent variable models[C]. The 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019: 2079–2089.

[14] HAO Fu, LI Chunyuan, LIU Xiaodong, *et al*. Cyclical

annealing schedule: A simple approach to mitigating KL vanishing[C]. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, USA, 2019: 240–250. doi: 10.18653/v1/N19-1021.

[15] HE Junxian, SPOKOYNY D, NEUBIG G, *et al*. Lagging inference networks and posterior collapse in variational autoencoders[C]. The 7th International Conference on Learning Representations, New Orleans, USA, 2019.

[16] ZHU Qile, BI Wei, LIU Xiaojiang, *et al*. A batch normalized inference network keeps the KL vanishing away[C/OL]. The 58th Annual Meeting of the Association for Computational Linguistics, 2020: 2636–2649. doi: 10.18653/v1/2020.acl-main.235.

[17] LI Ruizhe, LI Xiao, CHEN Guanyi, *et al*. Improving variational autoencoder for text modelling with timestep-wise regularisation[C]. The 28th International Conference on Computational Linguistics, Barcelona, Spain, 2020: 2381–2397. doi: 10.18653/v1/2020.coling-main.216.

[18] PANG Bo, NIJKAMP E, HAN Tian, *et al*. Generative text modeling through short run inference[C/OL]. The 16th Conference of the European Chapter of the Association for Computational Linguistics, 2021: 1156–1165.

[19] SILVA F, SANZ M, SEIXAS J, *et al*. Perceptrons from memristors[J]. *Neural Networks*, 2020, 122: 273–278. doi: 10.1016/j.neunet.2019.10.013.

[20] LIU Jiaqi, LI Zhenghao, TANG Yongliang, *et al*. 3D Convolutional Neural Network based on memristor for video recognition[J]. *Pattern Recognition Letters*, 2020, 130: 116–124. doi: 10.1016/j.patrec.2018.12.005.

[21] WEN Shiping, WEI Huaqiang, YANG Yin, *et al*. Memristive LSTM network for sentiment analysis[J]. *IEEE Transactions on Systems, Man, and Cybernetics:Systems*, 2021, 51(3): 1794–1804. doi: 10.1109/TSMC.2019.2906098.

[22] ADAM K, SMAGULOVA K, and JAMES A P. Memristive LSTM network hardware architecture for time-series predictive modeling problems[C]. 2018 IEEE Asia Pacific Conference on Circuits and Systems, Chengdu, China, 2018: 459–462. doi: 10.1109/APCCAS.2018.8605649.

[23] GOKMEN T, RASCH M J, and HAENSCH W. Training LSTM networks with resistive cross-point devices[J]. *Frontiers in Neuroscience*, 2018, 12: 745. doi: 10.3389/fnins.2018.00745.

[24] LI Can, WANG Zhongrui, RAO Mingyi, *et al*. Long short-term memory networks in memristor crossbar arrays[J]. *Nature Machine Intelligence*, 2019, 1(1): 49–57. doi: 10.1038/s42256-018-0001-4.

[25] LIU Xiaoyang, ZENG Zhigang, and WUNSCH II D C. Memristor-based LSTM network with in situ training and its applications[J]. *Neural Networks*, 2020, 131: 300–311. doi: 10.1016/j.neunet.2020.07.035.

[26] PARK Y, CHO J, and KIM G. A hierarchical latent structure for variational conversation modeling[C]. 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, USA, 2018: 1792–1801. doi: 10.18653/v1/N18-1162.

[27] CHUA L. Memristor-the missing circuit element[J]. *IEEE Transactions on Circuit Theory*, 1971, 18(5): 507–519. doi: 10.1109/TCT.1971.1083337.

[28] STRUKOV D B, SNIDER G S, STEWART D R, *et al*. The missing memristor found[J]. *Nature*, 2008, 453(7191): 80–83. doi: 10.1038/nature06932.

[29] KIM Y, WISEMAN S, MILLER A C, *et al*. Semi-amortized variational autoencoders[C]. The 35 th International Conference on Machine Learning, Stockholm, Sweden, 2018: 2678–2687.

[30] ZHAO Tiancheng, ZHAO Ran, and ESKENAZI M. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders[C]. The 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, 2017: 654–664. doi: 10.18653/v1/P17-1061.

[31] LI Yanran, SU Hui, SHEN Xiaoyu, *et al*. DailyDialog: A manually labelled multi-turn dialogue dataset[C]. The Eighth International Joint Conference on Natural Language Processing, Taipei, China, 2017: 986–995.

[32] KHEMAKHEM I, KINGMA D P, MONTI R P, *et al*. Variational autoencoders and nonlinear ICA: A unifying framework[C]. The Twenty Third International Conference on Artificial Intelligence and Statistics, Palermo, Italy, 2020.

[33] SERBAN I V, SORDONI A, LOWE R, *et al*. A hierarchical latent variable encoder-decoder model for generating dialogues[C]. The 31st AAAI Conference on Artificial Intelligence, San Francisco, USA, 2017.

[34] YU Lantao, ZHANG Weinan, WANG Jun, *et al*. SeqGAN: Sequence generative adversarial nets with policy gradient[C]. The Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, USA, 2017: 2852–2858.

胡小方：女，教授，研究方向为忆阻器、神经网络、机器学习、非线性系统与电路.

杨　涛：男，硕士生，研究方向为自然语言处理、忆阻器.

责任编辑：余　蓉