



DOI:10.12404/j.issn.1671-1815.2309630

引用格式: 闫钦与, 卜凡亮, 王一帆. 基于脉冲神经网络优化的动态图链路预测[J]. 科学技术与工程, 2025, 25(4): 1522-1528.

Yan Qinyu, Bu Fanliang, Wang Yifan. Dynamic graph link prediction optimized by spiking neural networks[J]. Science Technology and Engineering, 2025, 25(4): 1522-1528.

自动化技术、计算机技术

基于脉冲神经网络优化的动态图链路预测

闫钦与, 卜凡亮*, 王一帆

(中国人民公安大学信息安全学院, 北京 100038)

摘要 动态图链路预测通过图上的历史交互预测未来节点间链路的形成或消失。为减少基于循环神经网络建模网络动态的方法在细粒度时间的动态图上具有较高能耗的问题, 提出一种基于脉冲神经网络优化的动态图链路预测模型, 通过融合脉冲神经网络的节点记忆更新模块, 脉冲化节点记忆的更新过程, 训练图神经网络学习动态图的演化动态并实现链路预测。在3个公开经典数据集上的结果表明, 模型在运行速度上得到提升, 并保留了准确性, 在动态图链路预测任务中具有较好的性能表现。

关键词 动态图; 链路预测; 图神经网络; 脉冲神经网络

中图分类号 TP181; 文献标志码 A

Dynamic Graph Link Prediction Optimized by Spiking Neural Networks

YAN Qin-yu, BU Fan-liang*, WANG Yi-fan

(School of Information Network Security, People's Public Security University of China, Beijing 100038, China)

[Abstract] Dynamic graph link prediction aims to predict the formation or disappearance of links between nodes in a graph based on their historical interactions. To address the issue of high energy consumption associated with modeling dynamic networks using recurrent neural networks at fine-grained temporal graphs, a dynamic graph link prediction model optimized by spiking neural networks was proposed. By the node memory updater incorporated spiking neural networks and the spiking update process of node memory, the evolving dynamics of dynamic graphs were learned by graph neural networks and the model achieved link prediction. The results on three publicly available classic datasets show that the proposed model exhibits improved runtime efficiency while maintaining accuracy, showcasing favorable performance in dynamic graph link prediction tasks.

[Keywords] dynamic graph; link prediction; graph neural networks; spiking neural networks

图是一种对节点和边进行建模的数据结构, 能够有效表达对象间的交互关系^[1], 链路预测是图数据挖掘领域的典型问题, 与真实世界联系紧密, 具有丰富的研究价值。对于图数据的挖掘研究, 图神经网络(graph neural networks, GNNs)功能强大且可靠^[2], 在社交网络^[3]、交通预测^[4]、生物学^[5]等多个领域得到广泛应用。但真实世界中的图结构并非处于理想条件下的静止状态。以社交网络为例, 用户状态和用户间的消息传递随着时间推移而不断更新。过去的模型在静态图上表现良好, 但难以捕捉动态图时域层面的特征^[6]。为克服静态图神经网络的局限性, 研究人员开发了动态图神经网络

(dynamic graph neural networks, DGNNs), 通过对动态图的结构、上下文信息和时间关系进行学习, 实现了在时域上对图演化机理的探究, 并根据时间粒度的不同分为离散时间动态图(discrete-time dynamic graphs, DTDGs)和连续时间动态图(continuous-time dynamic graphs, CTDGs)两种应用场景。

以往的动态图链路预测方法多是基于相似性指标, 根据节点属性和特征相似度进行预测, 但需要人工预设网络特征, 在处理整体规模庞大、情况演化复杂的图结构时性能不佳^[7]。当GNNs成功应用于静态图问题后, 研究人员开始尝试将其迁移到动态图链路预测任务, 利用DGNNs学习节点历史交

收稿日期: 2023-12-06; 修订日期: 2024-11-07

基金项目: 中国人民公安大学安全防范工程双一流专项(2023SYL08)

第一作者: 闫钦与(2000—), 男, 汉族, 内蒙古呼和浩特人, 硕士研究生。研究方向: 图神经网络。E-mail: yanqinyu@126.com。

*通信作者: 卜凡亮(1965—), 男, 汉族, 江苏徐州人, 博士后, 教授。研究方向: 社会治理、大数据与人工智能。E-mail: bufanliang@sina.com。

互以及动力学性质, 实现对未来链路的预测^[8]。其中一种方法是将动态图视为 DTDGs, 建模为一系列在固定时间间隔下的静态图快照^[9]。Sankar 等^[10] 通过结合结构邻域和时间动态的双重自注意力机制计算图中节点的表示。刘林峰等^[11] 基于自动编码器和门控循环单元处理转化为邻接矩阵的目标网络, 捕获时间维度的演化特征。相比而言, CTGDs 通过带有连续时间戳的事件列表记录节点或边的添加、删除以及特征转换, 更符合现实场景^[9]。Kumar 等^[12] 通过两个循环神经网络 (recurrent neural networks, RNNs) 来更新用户和物品每次交互时的节点嵌入。Rossi 等^[9] 引入基于 CTGDs 的通用归纳框架, 并通过记忆模块实现历史交互的整合。上述方法多以 RNNs 及其变体捕获时间维度的演化特征, 成功表示了节点的交互历史, 但高维向量和大量记忆单元的存在也提高了模型的计算成本和内存占用^[13], 因此需要以新的角度解决当前多数动态图链路预测方法在整体规模大和时间跨度长的动态图上计算受限问题。

基于计算机科学的 (artificial neural networks, ANNs) 技术在图像分类^[14]、目标检测^[15] 等领域取得卓越训练效果的同时, 研究人员注意到人脑基于脉冲的信息传递机制在认知和学习方面的明显优势^[16]。相较于使用浮点值的 ANNs, 基于神经科学的脉冲神经网络 (spiking neural networks, SNNs) 仅在膜电位达到阈值时通过脉冲传递二进制和异步信息^[17]。通过引入神经动力学和脉冲特性, SNNs 具有处理复杂时序数据的能力、较低的能耗以及较强的鲁棒性, 并展现出了持续学习的潜力^[18]。但复杂的时空传播过程和不可微分的脉冲触发机制导致深度 SNNs 的算法设计工作并不顺利^[16]。目前主流的两种构建深度 SNNs 的方法包括: ①假设 SNNs 与 ANNs 计算单元等价, 将 ANNs 训练的权重映射到 SNNs 的 ANN-SNN 转换方法^[19]; ②在反向传递时可用微函数近似神经元激活函数的基于替代梯度的直接学习方法^[20]。一些研究也开始将 SNNs 应用于图领域^[21], 但主要集中于静态图, 忽略了 SNNs 的动态特性, 因此在动态图研究领域仍具有广阔的探索空间。

为减少 TGN (temporal graph networks)^[9] 和 DySAT (dynamic self-attention network)^[10] 等模型在使用 RNNs 及其变体处理图动态特征时的计算消耗, 基于 TGN^[9] 模型, 融合 SNNs 的思想, 提出基于脉冲神经网络优化的动态图链路预测模型 DLOS (dynamic graph link prediction optimized by spiking neural networks)。与一些具有代表性的 DGNNs 模

型相比, DLOS 捕捉动态图结构中的复杂模式和关系的同时, 只在神经元发放脉冲而非每个固定时间步进行计算, 这种事件驱动的稀疏计算方式减少了计算消耗, 提高了链路预测任务的准确性。

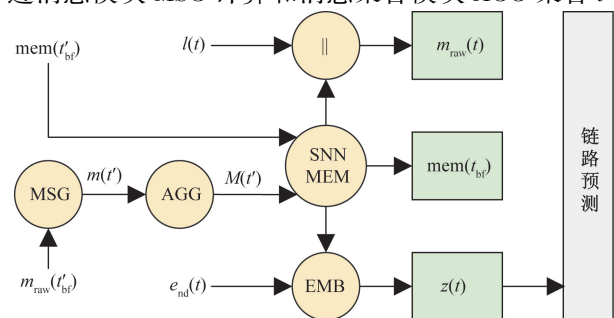
1 基于脉冲神经网络优化的动态图链路预测模型

1.1 问题定义

处理动态图的模型可分为将动态图映射到节点嵌入的编码器 (encoder) 和将嵌入作为链路预测等任务输入的解码器 (decoder)^[22]。DLOS 的编码器部分基于 TGN 模型^[9], 适用于 CTGDs 的链路预测任务。CTGDs 可表示为关于时间的事件集合 $G = \{e(t_1), e(t_2), \dots\}$, 其中时间 $0 \leq t_1 \leq t_2 \leq \dots$, 事件 $e(t)$ 则存在两种情况: ①节点本身的变化, 如节点的加入和退出, 可以用 $e_{nd}(t)$ 表示; ②节点间的交互, 如两节点建立联系, 可以用 $l(t)$ 表示。模型的目标是在给定动态图 G 后, 通过基于 SNNs 优化的记忆模块学习时间 $0 \leq t_1 \leq \dots \leq t_{bf} \leq t$ 下动态图中的历史事件集合 $G_H = \{e(t_1), e(t_2), \dots, e(t_{bf})\}$, 产生节点嵌入集合 $Z = \{z(t_1), z(t_2), \dots, z(t_{bf})\}$, 并通过解码器预测未来 $t > t_{bf}$ 时的交互链路 $l(t_{>bf})$ 。

1.2 模型架构

模型的总体任务是在动态图上实现链路预测, 即假设存在一个按时间顺序排列的交互列表, DLOS 将通过观察到的历史交互预测未来的交互。DLOS 的整体架构如图 1 所示, 其中, t' 为每个节点最后一个事件对应的时刻, t_{bf} 为紧接 t' 之前的时刻, 模型通过消息模块 MSG 计算和消息聚合模块 AGG 聚合 t'



MSG 为消息模块; AGG 为聚合模块; SNN-MEM 为融合 SNNs 的节点记忆模块; || 为连接操作; EMB 为嵌入模块; $m_{raw}(t'_{bf})$ 为 t' 时刻的原始消息; $mem(t'_{bf})$ 为 t'_{bf} 时刻的节点记忆; $m(t')$ 为通过消息模块处理的 t' 时刻原始消息; $M(t')$ 为通过消息模块和聚合模块处理的 t' 时刻原始消息; $l(t)$ 为 t 时刻的交互; $m_{raw}(t)$ 为交互经过更新和连接后的结果; $mem(t_{bf})$ 为经过 SNN-MEM 更新得到的节点记忆; $e_{nd}(t)$ 为节点事件; $z(t)$ 为节点嵌入

图 1 DLOS 架构图

Fig. 1 Framework diagram of DLOS

时刻的原始消息 $m_{\text{raw}}(t')$, 对融合 SNNs 的节点记忆模块 SNN-MEM 进行更新。 t'_{bf} 时刻的节点记忆 $\text{mem}(t'_{\text{bf}})$ 经过 SNN-MEM 更新得到 $\text{mem}(t_{\text{bf}})$, 交互 $l(t)$ 经过更新和连接得到 $m_{\text{raw}}(t)$, 节点个体事件 $e_{\text{nd}}(t)$ 经过更新和嵌入模块 EMB 得到节点嵌入 $z(t)$, 并最终用于链路预测。 DLOS 按时间顺序批处理所有节点间的交互, 它将每个节点的最后一条消息存储在消息存储模块中, 以便在预测该节点的下次交互之前对其进行处理。

在动态图 G 中每个与节点 n_i 有关的事件都对应一条消息 (message)。 DLOS 关键模块的训练流程如图 2 所示。 原始消息存储模块存储计算消息所需的原始消息, 这些原始消息来自模型已经学习的历史交互, 同时模型将新交互产生的记忆更新推迟至之后的批处理。 记忆可以使用先前批处理中得到的消息实现更新, 并通过刚刚更新的记忆计算嵌入。 这样的数据流动方式使得记忆相关模块的计算直接影响损失。 最后, 此次批处理交互对应的原始消息被存储在原始消息存储模块中以便未来使用。 DLOS 关键模块的具体内容将在下文进行论述。

1.3 消息模块

DLOS 可以通过消息模块计算消息从而更新 n_i 的节点记忆。 对于节点层面的事件, 设节点 n_i 在时间 t 发生事件 $e_{\text{ndi}}(t)$, 其对应的消息可表示为

$$m_i(t) = \text{MSG}_{\text{node}}[e_{\text{ndi}}(t), \text{mem}_i(t_{\text{bf}})] \quad (1)$$

对于节点间的事件, 设源节点 n_{src} 和目标节点 n_{dst} 在时间 t 存在交互 $l(t)$, 则可以分别计算出两节点的消息。

$$m_{\text{src}}(t) = \text{MSG}_{\text{src}}[l(t), \text{mem}_{\text{src}}(t_{\text{bf}}), \text{mem}_{\text{dst}}(t_{\text{bf}})] \quad (2)$$

$$m_{\text{dst}}(t) = \text{MSG}_{\text{dst}}[l(t), \text{mem}_{\text{dst}}(t_{\text{bf}}), \text{mem}_{\text{src}}(t_{\text{bf}})] \quad (3)$$

式中: $\text{mem}_i(t_{\text{bf}})$ 为节点 n_i 在时间 t 之前的节点记忆; m_{src} 为源节点的消息; m_{dst} 为目标节点的消息; MSG_{node} 、 MSG_{src} 和 MSG_{dst} 为事件对应的消息函数; $l(t)$ 为源节点与目标节点的交互; $\text{mem}_{\text{src}}(t_{\text{bf}})$ 为源节点在时间 t 之前的节点记忆; $\text{mem}_{\text{dst}}(t_{\text{bf}})$ 为目标节点在时间 t 之前的节点记忆。

由于同一节点可能涉及多个事件, 在进行批处理时可能会导致事件对应的消息大量涌现, 影响处理效率, 因此需要对这些消息进行聚合处理, 即

$$M_i(t) = \text{AGG}[m_i(t_1), m_i(t_2), \dots, m_i(t_N)] \quad (4)$$

式(4)中: M_i 为经过聚合后的节点消息; 时间 $t \geq t_1, t_2, \dots, t_N$; AGG 为聚合函数; $m_i(t)$ 为有关节点 n_i 的事件在不同时间下对应的消息。

经过聚合的消息将交给记忆模块进行进一步处理。

1.4 记忆模块

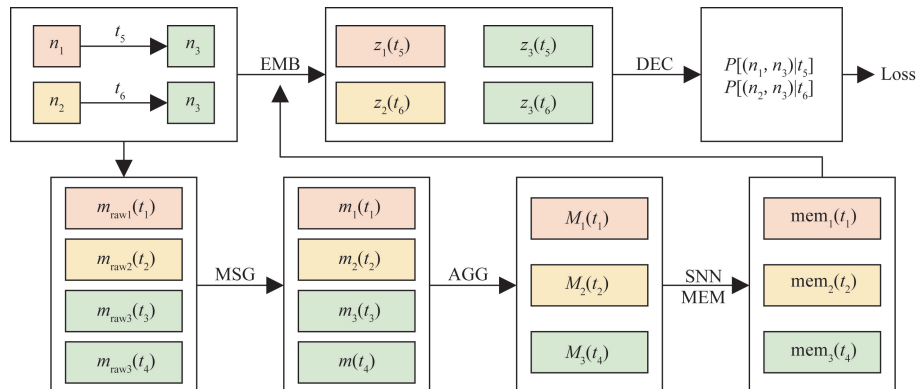
当动态图 G 出现新节点 n_{new} 时, 它的节点记忆将被初始化为零向量。 随着 G 的不断演化, n_{new} 自身可能发生改变, 或与其他节点出现新的交互, 其节点记忆也随之不断更新, 以压缩形式表示 n_{new} 在图中的历史。 对于节点自身的事件, 只更新相应节点的记忆; 对于节点间的交互事件, 事件发生后两个节点的记忆全部更新。

模型在时间 t 时的记忆 mem_{mdl} 可表示为

$$\text{mem}_{\text{mdl}} = \{\text{mem}_1(t), \text{mem}_2(t), \dots, \text{mem}_n(t)\} \quad (5)$$

式(5)中: $\text{mem}_i(t)$ 为时间 t 时节点 n_i 的记忆。

结合前文经过聚合处理的消息, 包含所有与 n_i 节点相关事件的节点记忆更新可表示为



MSG 为消息模块; AGG 为聚合模块; SNN-MEM 为融合 SNNs 的节点记忆模块; EMB 为嵌入模块; DEC 为解码器; Loss 为损失值; n_i 为节点; $m_{\text{raw}i}(t_i)$ 为 t_i 时刻的原始消息; $m_i(t_i)$ 为通过消息模块处理的 t_i 时刻原始消息; $M_i(t_i)$ 为通过消息模块和聚合模块处理的 t_i 时刻原始消息; $\text{mem}_i(t_i)$ 为 t_i 时刻的节点记忆; $z_i(t_i)$ 为节点嵌入; P 为某 2 个节点在某时刻存在交互的概率

图 2 DLOS 关键模块训练流程图

Fig. 2 Training workflow diagram of key modules in DLOS

$$\text{mem}_i = \text{MEM}[\text{mem}_i(t_{\text{tr}}), M_i(t)] \quad (6)$$

式(6)中: MEM 为可学习的记忆更新函数。

与以往应用 RNNs 及其变体的思路不同, 在 DLOS 中使用一个基于 SNNs 的记忆更新模块, 它的前向传播过程接受输入张量和隐藏状态张量, 并根据脉冲神经元的原理计算输出, 从而以更快的速度和更小的消耗对动态图中事件的历史进行保存并更新。

SNNs 中的神经元通常具有膜电位 (membrane potential) 和阈值 (threshold) 两个组成部分^[13]。膜电位代表神经元内部的电位状态, 它随时间变化, 并受输入信号和隐藏状态的影响。具体而言, 更新模块中线性变换层用于处理输入张量 \mathbf{x} 和隐藏状态张量 \mathbf{h} , 线性变换的结果则分别代表了输入和隐藏状态对膜电位的影响; 当膜电位超过阈值时, 神经元会发出脉冲信号, 并将膜电位重置为初始电位, 这些功能可以通过替代函数 (surrogate function) 间接实现。替代函数决定是否发出脉冲信号, 即是否将激活后的结果作为神经元的输出。不同于传统的 ANNs 通过将非线性激活函数 (如 Sigmoid 函数、ReLU 函数等) 应用于线性变换结果得到神经元激活状态, SNNs 的神经元激活状态由离散时间点上的脉冲事件表示。在 DLOS 中, 更新模块将经过线性变换后的结果相加融合输入和隐藏状态的信息, 然后传递给实现高斯误差函数 (erf) 的替代函数进行激活。替代函数默认输出脉冲, 在前向传播时使用 heaviside 函数, 而在反向传播时使用高斯误差函数的梯度的脉冲发放函数, 可表示为

$$g'(x) = \frac{\alpha}{\sqrt{\pi}} e^{-\alpha^2 x^2} \quad (7)$$

式(7)中: α 为控制反向传播时梯度的平滑程度的参数。

对应的原函数为

$$\begin{aligned} g(x) &= \frac{1}{2} [1 - \text{erf}(-\alpha x)] \\ &= \frac{1}{2} \text{erfc}(-\alpha x) \\ &= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\alpha x} e^{-t^2} dt \end{aligned} \quad (8)$$

式(8)中: erf 为误差函数; erfc 为误差互补函数; x 为自变量。

上述机制实现了脉冲信号与其他神经元的联通, 从而构成神经网络的连接和信息传递。凭借记忆更新模块, DLOS 能够记忆图中每个节点的长期依赖关系, 最终成功用于处理动态图的时间序列数据任务。

1.5 嵌入模块

在记忆模块中, 节点 n_i 的记忆仅在参与事件时更新。在长时间没有新事件发生的情况下, 可能导

致 n_i 的记忆出现过时。嵌入模块主要用于生成任意时间 t 时节点 n_i 的时间嵌入 $z_i(t)$, 从而避免记忆过时^[22]。 $z_i(t)$ 在不同场景下具有不同的计算方法, 包括时间投影^[12]、时间图注意力^[23]等。

$z_i(t)$ 可表示为

$$\begin{aligned} z_i(t) &= \sum_{j \in N_i^{(k)}([0, t])} h[e_{\text{ndi}}(t), e_{\text{ndj}}(t), l_{ij}(t), \text{mem}_i(t), \text{mem}_j(t)] \\ &= \text{EMB}(t, i) \end{aligned} \quad (9)$$

式(9)中: $N_i^{(k)}([0, t])$ 为时间间隔 $[0, t]$ 下 n_i 的 k 跳邻域; h 为一个可学习的函数; $e_{\text{ndi}}, e_{\text{ndj}}$ 分别为节点 n_i, n_j 对应的事件; l_{ij} 为节点 n_i, n_j 的交互; $\text{mem}_i, \text{mem}_j$ 分别为节点 n_i, n_j 的记忆; EMB 为嵌入函数。

当动态图中一个节点处于非活跃状态时, 它的一些邻居可能恰恰相反, 通过聚合它们的节点记忆, DLOS 能够计算最新的节点嵌入, 并进一步应用于下游的链路预测任务。

2 实验

2.1 数据集

为验证 DLOS 的性能, 使用 3 个广泛使用的 CTDGs 数据集进行实验。数据集的具体信息如表 1 所示。

公共数据集 MOOC^[12] 由学生在 MOOC 在线课程上包括观看网课、提出问题在内的各种行为组成, 包含 7 047 名用户与 98 个项目, 共出现超过 411 749 次交互; Reddit^[9] 为用户在 subreddits 网站上发布的一个月内的帖子, 包含 1 000 个最活跃的子 Reddit 作为项目, 并选择 10 000 个最活跃的用户, 共出现 672 447 次交互, 且带有来自 366 个 Reddit 被封禁用户的真实标签; Wikipedia^[9] 来自维基百科网站编辑一个月内的修改, 包含 1 000 个修改最多的页面作为项目, 并选择了至少进行 5 次编辑的编辑者共 8 227 人作为用户, 共出现 157 474 次交互, 且带有 217 个被封禁用户的真实标签。由于 MOOC 原始数据集不包含节点或边缘特征, 模型随机生成了 128 维边缘特征。在实验整个持续时间内按时间戳顺序将数据集以 75% -15% -15% 的比例划分, 用于训练、验证和测试。

表 1 数据集信息

Table 1 Dataset statistic

数据集	节点数	边数	最大边时间戳	类别数	标签数	边特征维度
MOOC	7 000	412 000	2.6×10^6	—	—	128*
Reddit	11 000	672 000	2.7×10^6	2	366	172
Wikipedia	9 000	157 000	2.7×10^6	2	217	172

注: * 表示随机化特征。

2.2 实验设置

为确保实验的公平性,本实验全部在相同的CPU: Intel Xeon Processor (Skylake, IBRS)和GPU: NVIDIA GeForce RTX3060 实验环境上使用 PyTorch 2.0.0 实现,并统一采用 AP (average precision) 指标、AUC (area under the curve) 指标和每次 epoch 的训练平均用时衡量模型链路预测的性能。AP 指标是准确率-召回率 (precision-recall, PR) 曲线与准确率轴 (precision) 所围成图形的面积。假设所有样本中 m 个正例对应的召回值 (recall) 能够计算出 m 个最大准确率,这些准确率取平均值即得到 AP。AP 反映了模型的平均精确度,其值越大模型的精确度越高;AUC 指标是 ROC (receiver operating characteristic curve) 曲线下的图形面积,可视作随机选择的缺失链路比随机选择不存在的链路得到更高分数的概率^[24]。如果 AUC 高于 0.5,说明模型比随机选择的结果更准确,且数值越大表示模型链路预测的效果越好。

为客观衡量 DLOS 的性能,选择 4 种代表性的可用于链路预测的 DGNNs 变体与 DLOS 进行比较,具体如下。

(1) APAN^[25] 基于记忆。通过分离模型推理,减轻图查询操作对模型推理速度的影响。该模型使用异步 CTDGs 算法,捕捉更高阶的结构特征,并在提高推理速度的同时实现竞争性的性能表现。

(2) DySAT^[10] 基于快照。通过结合自注意力机制和两个维度 (结构邻域和时间动态) 的联合注意力,计算节点表示。相比于现有的建模图演化的最先进的循环方法,动态自注意力方法具有高效性,并且在性能上表现出一致的优越性。

(3) TGAT^[23] 基于注意力。提出针对动态图的归纳表示学习方法,引入基于经典谐波分析的时间编码技术,通过堆叠 TGAT 层,模型能够将节点嵌入识别为时间函数,并能够在图演化过程中归纳地推断出新节点和观察节点的嵌入。

(4) TGN^[9] 基于记忆。针对连续时间的动态图、基于记忆模块的节点历史进行建模,通过时间嵌入进行图节点编码,并提出并行处理的训练策略,模型能够捕捉节点的长期依赖关系,在处理动态图时更好地理解节点的演化。

模型参数方面,在所有基线方法中,将感受野设置为 2 跳,每跳采样器的邻居数量固定为 10,邮箱大小在 APAN 中设置为 10,而在其他方法中设置为 1,时间嵌入维度设置为 100,动态节点嵌入的输出维度设置为 100,多头注意力机制的头数为 2,APAN 的记忆大小设置为 10,而 TGN 的记忆大小设

置为 1。对于基于快照的 DySAT,使用 3 个快照,每个快照的持续时间为 10 000 s。对于所有数据集的训练,使用 Adam 优化器,学习率设置为 0.000 1,丢弃率设置为 0.2,训练、验证和测试的批大小为 600,为避免出现过拟合在训练过程中采用早停法,当训练集上的损失减小的程度连续 5 次小于某个阈值时候停止继续训练。

2.3 实验结果与分析

DLOS 在 3 个数据集上的实验结果如表 2 所示,其中所有结果均为 10 次运行后的平均值,在每个数据集上运行的最优。通过对比可以发现,DLOS 在绝大多数场景下都有着更高的 AP 和 AUC,这是因为相较于其他基线,DLOS 基于性能出色的 TGN 模型^[9]进行改进,并基于 SNNs 重新设计记忆更新模块。SNNs 使用脉冲编码和事件驱动的方式来处理信息,具有更细致的时间建模。在 DGNNs 中,特别是在节点的记忆更新任务中,时间精度是非常关键的因素,而 SNNs 能够更好地捕捉到节点状态的微妙变化和时序关系。可以看出,DLOS 每次 epoch 的平均训练用时比基线模型更短,相较于速度较快的 APAN 模型耗时平均短 7%,比改进前的 TGN 模型耗时平均短 11%,具有更快的运行速度。这是由于 SNNs 的计算基于脉冲事件,与以往需要大量的存储模块来存储时间上下文信息和大量的训练数据的深度学习模型相比在计算上更为轻量。而传统的 RNNs 通常是基于离散的时间步进行计算的,每个时间步都需要进行前向传播和反向传播。得益于 SNNs 的时空特性,所设计的记忆更新模块聚合并更新信号,即脉冲尖峰,无需同传统模型一样处理固定的时间步在每个时间步,具有更低计算和内存开销,提高了在 DGNNs 中的运行速度。

表 2 数据集的预测结果

Table 2 Prediction results of dataset

数据集	模型	AP	AUC	用时/s
MOOC	APAN	98.89	99.42	9.6
	DySAT	98.88	99.33	37.2
	TGAT	98.31	98.78	25.0
	TGN	99.38	99.64	10.3
	DLOS	99.41	99.64	9.0
Reddit	APAN	96.96	97.96	19.8
	DySAT	98.52	98.39	49.3
	TGAT	99.62	99.66	59.7
	TGN	99.52	99.60	19.8
	DLOS	99.67	99.71	17.3
Wikipedia	APAN	97.00	97.85	3.7
	DySAT	96.65	96.90	11.9
	TGAT	97.99	98.21	18.8
	TGN	99.21	99.51	3.9
	DLOS	99.46	99.29	3.6

注:加粗的部分数据表示最优结果。

3 结论

针对动态图领域的链路预测问题, 提出基于脉冲优化的动态图链路预测模型 DLOS。DLOS 利用 TGN 模型的优势, 重新设计基于 SNNs 的记忆更新模块, 通过采用脉冲化的方式来更新节点记忆, 捕捉动态图不断演变过程中节点时间维度的特征, 记忆图中每个节点的长期依赖关系, 有效降低了在动态图上的计算成本和内存开销。在 3 个动态图公开数据集上对 DLOS 性能表现进行实验, 实验结果证明这一设计在不影响准确性的前提下, 成功地提高了模型的运行速度, 在动态图链路预测任务中展现出较好的性能。未来工作中, 将继续发挥 SNNs 的优势, 尝试挖掘包括网络暴力群组在内的动态社交网络结构的动力学模式, 研究时间跨度更大、图范围更广、生活中更加普遍的动态异构图场景。

参 考 文 献

- [1] 邹长宽, 田小平, 张晓燕, 等. 基于 GraphSage 节点度重要性聚合的网络节点分类研究[J]. 科学技术与工程, 2022, 22(32): 14306-14312.
Zou Changkuan, Tian Xiaoping, Zhang Xiaoyan, et al. A study of network node classification based on GraphSage node degree importance aggregation[J]. Science Technology and Engineering, 2022, 22(32): 14306-14312.
- [2] Lin H, Yan M, Ye X, et al. A comprehensive survey on distributed training of graph neural networks[J]. arXiv Preprint, 2022: arXiv: 2211.05368.
- [3] Wu Y, Lian D, Xu Y, et al. Graph convolutional networks with markov random field reasoning for social spammer detection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2020: 1054-1061.
- [4] 赵萍, 李欣, 朱少武. 基于时空图注意力神经网络的交通道路拥塞和异常预测[J]. 科学技术与工程, 2022, 22(3): 1271-1278.
Zhao Ping, Li Xin, Zhu Shaowu. Traffic road congestion and anomaly prediction based on spatio-temporal graph attention neural networks[J]. Science Technology and Engineering, 2022, 22(3): 1271-1278.
- [5] Fout A, Byrd J, Shariat B, et al. Protein interface prediction using graph convolutional networks[J]. Advances in Neural Information Processing Systems, 2017, 30.
- [6] 唐晨, 赵杰煜, 叶绪伦, 等. 动态图的链接预测模型[J]. 计算机科学与探索, 2022, 16(10): 2365-2376.
Tang Chen, Zhao Jieyu, Ye Xulun, et al. Link prediction model for dynamic graphs[J]. Journal of Frontiers of Computer Science & Technology, 2022, 16(10): 2365-2376.
- [7] 刘琳岚, 冯振兴, 舒坚. 基于时序图卷积的动态网络链路预测[J]. 计算机研究与发展, 2024, 61(2): 518-528.
Liu Linlan, Feng Zhenxing, Shu Jian. Dynamic network link prediction based on sequential graph convolution[J]. Journal of Computer Research and Development, 2024, 61(2): 518-528.
- [8] Jin M, Li Y F, Pan S. Neural temporal walks: Motif-aware representation learning on continuous-time dynamic graphs[J]. Advances in Neural Information Processing Systems, 2022, 35: 19874-19886.
- [9] Rossi E, Chamberlain B, Frasca F, et al. Temporal graph networks for deep learning on dynamic graphs[J]. arXiv Preprint, 2020: arXiv: 2006.10637.
- [10] Sankar A, Wu Y, Gou L, et al. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks[C]//Proceedings of the 13th International Conference on Web Search and Data Mining. New York: Association for Computing Machinery, 2020: 519-527.
- [11] 刘林峰, 于子兴, 祝贺. 基于门控循环单元的移动社会网络链路预测方法[J]. 计算机研究与发展, 2023, 60(3): 705-716.
Liu Linfeng, Yu Zixing, Zhu He. A link prediction method based on gated recurrent units for mobile social network[J]. Journal of Computer Research and Development, 2023, 60(3): 705-716.
- [12] Kumar S, Zhang X, Leskovec J. Predicting dynamic embedding trajectory in temporal interaction networks[C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & data mining. New York: Association for Computing Machinery, 2019: 1269-1278.
- [13] Li J, Yu Z, Zhu Z, et al. Scaling up dynamic graph representation learning via spiking neural networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2023, 37(7): 8588-8596.
- [14] 刘春容, 宁芊, 雷印杰, 等. 改进残差神经网络在遥感图像分类中的应用[J]. 科学技术与工程, 2021, 21(31): 13421-13429.
Liu Chunrong, Ning Qian, Lei Yinjie, et al. Application of improved residual network in sensing image classification[J]. Science Technology and Engineering, 2021, 21(31): 13421-13429.
- [15] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Columbus: IEEE, 2014: 580-587.
- [16] Fang W, Chen Y, Ding J, et al. SpikingJelly: an open-source machine learning infrastructure platform for spike-based intelligence[J]. Science Advances, 2023, 9(40): 1480.
- [17] Kim S, Park S, Na B, et al. Spiking-yolo: spiking neural network for energy-efficient object detection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2020, 34(7): 11270-11277.
- [18] 陈鼎, 黄杨茹, 彭佩玺, 等. 脉冲强化学习算法研究综述[J]. 计算机学报, 2023, 46(10): 2132-2160.
Chen Ding, Huang Yangru, Peng Peixi, et al. Research on spiking reinforcement learning algorithms: a survey[J]. Chinese Journal of Computers, 2023, 46(10): 2132-2160.
- [19] Bu T, Ding J, Yu Z, et al. Optimized potential initialization for low-latency spiking neural networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2022, 36(1): 11-20.
- [20] Wu Y, Deng L, Li G, et al. Spatio-temporal backpropagation for training high-performance spiking neural networks[J]. Frontiers in Neuroscience, 2018, 12: 331.

- [21] Zhu Z, Peng J, Li J, et al. Spiking graph convolutional networks [J]. arXiv Preprint, 2022; arXiv: 2205. 02767.
- [22] Kazemi S M, Goel R, Jain K, et al. Representation learning for dynamic graphs: a survey[J]. The Journal of Machine Learning Research, 2020, 21(1): 2648-2720.
- [23] Xu D, Ruan C, Korpeoglu E, et al. Inductive representation learning on temporal graphs[J]. arXiv Preprint, 2020; arXiv: 2002. 07962.
- [24] Lü L, Zhou T. Link prediction in complex networks: a survey [J]. Physica A: Statistical Mechanics and Its Applications, 2011, 390(6): 1150-1170.
- [25] Wang X, Lyu D, Li M, et al. APAN: asynchronous propagation attention network for real-time temporal graph embedding[C]// Proceedings of the 2021 International Conference on Management of Data. New York: Association for Computing Machinery, 2021: 2628-2638.