

南 开 大 学

本 科 生 毕 业 论 文 （ 设 计 ）

中文题目： 基于图卷积神经网络的反洗钱方法

外文题目： An anti-money-laundering method
based on graph convolutional network

学 号： 1813033

姓 名： 王子旭

年 级： 2018 级

学 院： 软件学院

系 别： 软件工程

专 业： 软件工程

完成日期： 2022 年 5 月

指导教师： 刘明铭 教授

程学旗 研究员（中科院计算所）

关于南开大学本科生毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或没有公开发表的作品内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年 月 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

摘 要

“洗钱”是指通过合法的金融流程处理、转化非法所得财物，以达到掩盖违法收入来源的行为。洗钱常常与毒品、走私、赌博、贩卖人口等严重违法犯罪行为相关，且往往涉案金额巨大，据联合国近期报告的估算，全球每年由洗钱产生的收益占各国 GDP 总和的 2% ~ 5%，即 1.6 万亿 ~ 4 万亿美元。洗钱者的诡计多端加上涉及到的交易数据量十分庞大，人工检测和监督已然失效。探索基于大数据和人工智能技术的反洗钱方法事关国家经济发展、民众幸福生活，具有重要的现实意义。

本文利用中国建设银行提供的一周内的交易流水等数据，参考 AAAI2020 中的 EvolveGCN 模型，构建了适用于动态图的、以循环神经网络驱动训练的图卷积神经网络模型 AML-eGCN。通过半监督学习的方式对该模型进行训练，检测出了涉嫌参与洗钱的账号，并同静态图卷积神经网络模型 GCN、SGC 等进行了对比和分析。

关键词：图卷积神经网络；反洗钱；半监督学习

Abstract

Money laundering refers to the act of processing and transforming illegally obtained property through legal financial processes to cover up the source of illegal income. Money laundering is often related to serious illegal and criminal acts such as drugs, smuggling, gambling and human trafficking. Besides, the amount involved is often huge. According to recent United Nations' reports, the global annual income generated by money laundering accounts for 2% \sim 5% of the total GDP of all countries, which is \$1.6 trillion to \$4 trillion dollars. Tricks of money launderers, and the huge amount of transaction data make manual detection and supervision impossible. Therefore, it is of great practical significance to explore anti money laundering methods based on big data and artificial intelligence technology, which is related to national economic development and people's happiness.

Referring to the EvolveGCN model proposed in AAAI2020, I construct a dynamic graph convolutional network called AML-eGCN which is driven by recurrent neural network to detect the suspected accounts in oneweek's transaction flow data provided by China Construction Bank. The model is trained through semi-supervised method and compared to static graph convolutional network models like GCN and SGC in the paper.

Key Words: graph convolutional network; anti money laundering

目录

摘要	I
Abstract	II
目录	III
第一章 引言	1
第一节 研究背景	1
第二节 问题描述	1
第三节 文献综述	2
第四节 主要工作	2
第五节 章节介绍	3
第二章 理论基础	4
第一节 图卷积神经网络	4
第二节 循环神经网络	8
第三节 半监督学习	12
第四节 本章小结	14
第三章 模型原理	15
第一节 数据存储和处理方式	15
第二节 RNN 驱动训练	15
第三节 模型评价机制	15
第四章 实验	16
第一节 实验条件	16
第二节 实验流程	16
第三节 实验结果	16
第五章 讨论和总结	17
致 谢	XVIII
个人简历	XIX

第一章 引言

第一节 研究背景

洗钱是违法犯罪分子掩饰其非法所得的行为，处在违法犯罪链条的下游。将大量黑钱通过合法的流程转化为看似合法的收入，严重扰乱着社会正常的生产生活，极易滋生贪污腐败，直接或间接地侵害着人民生命财产安全。为精准打击洗钱行为，维护社会稳定、保护人民生命和财产安全、维持国家长治久安，由中国人民银行领头、同公安有关部门联合，各大银行纷纷开展反洗钱项目。本文作者利用中国建设银行提供的一周内的交易流水及少量标签数据展开研究。

第二节 问题描述

基于如下数据对所有账户进行分类，类别包括：涉嫌参与洗钱的黑标账户和未参与洗钱的白标账户。数据列表如下：

1. 一周内的交易流水记录 *edge_list.csv*。每一行表示一次交易，包括发起转账账户标识 *from_acc_id*、接受转账账户标识 *to_acc_id*、交易发生时间 *timestamp* 以及交易金额 *money*
2. 账户的特征映射 *node_features.csv*。每一行表示一个账户的特征向量，该向量是经过特征工程人工提取出来的。
3. 账户的标签 *node_labels.npy*。这是一个形状为 $2 * \text{num_nodes}$ 的矩阵，第一列表示账户标识，第二列则为账户标签。标签为-1 则表示无标签账户，为 0 则表示黑标账户，为 1 则表示白标账户。




把每个账户当作节点、每一次交易当作边，那么交易流水信息便可以构成图结构（下文将交替使用账户与节点，交易与边，请注意辨析）。在此基础上，反洗钱任务可以看成对这个图的节点进行二分类。结合中国建设银行提供的数

据，以及所在场景，该反洗钱任务具有如下显著特点：

1. 边带有权重。每条边都具有交易时间、交易金额的信息。
2. 具有多重边。两个固定的节点之间可以有多条边，表示两个账户在不同时间点的交易。
3. 邻接矩阵稀疏。大部分账户的交易关系较为简单，表现为图的邻接矩阵十分稀疏。
4. 带标签数据比例低。大部分的账户是没有标签的，我们必须依靠少量的标签对分类模型进行训练。
5. 样本类别分布不均衡。参与洗钱的账户是极少数的，表现为不同标签的比例差距很大。

特点一要求模型把图的拓扑结构、图的时序性一起考虑；特点二则对图的表示方式造成困难；特点三则使得图中的信息量少且分布不均匀，增加训练难度；特点四则要求模型的泛化能力足够强，使得少量的标签也能够训练出适用于其他大量节点的分类模型；特点五则要求我们慎重选择模型的评价方式，特别是需要重视样本比例极端情况下准确率虚高的现象。本文构建的模型很好地解决了上述特点造成的问题。解决方案详见第三章。

第三节 文献综述

经调研，当前主流的反洗钱算法可大致分为两类：基于统计学的大数据分析算法和谱图理论的方法。基于统计学的方法通过对洗钱模式进行分析，归纳、构建出若干种重要指标，再对海量交易流水信息进行指标的统计。最后将指标的统计结果进行分布检测，据此找出具有特殊指标的账户，即为涉嫌参与洗钱的账户。该领域的代表有：基于‘转入清空’模式的 Monlad、基于‘沉默爆发’模式的 SleepingBeauty、 HoloScope、 FlowScope、 EngineSpoke

第四节 主要工作

本文的主要工作如下：1. 参考了 AAAI2020 的 EvolveGCN 模型^[Evolvegcnn]，基于中国建设银行的带有少部分的数据，采用半监督学习的方法训练模型，具

有较高的分辨出黑标账户（涉嫌洗钱的账户）的准确率。2. 采用 RNN^[GRU] 驱动 GCN 的训练，融合了图的拓扑信息和时序信息，和同属于图卷积神经网络的 GCN^[kipfGCN]、SGC^[SGC] 等方法进行了比较和分析。

第五节 章节介绍

1. 第一章为引言部分。本章共五个小节，分别为研究背景、问题描述、文献综述、主要工作和章节介绍。
2. 第二章为理论基础部分。主要介绍 EvolveGCN 的理论基础，包括三个小节：图卷积神经网络-GCN、循环神经网络-RNN 以及半监督学习原理。
3. 第三章为模型原理部分。具体介绍 EvolveGCN 的结构、训练和评价机制。分为三小节：数据存储和处理方式、RNN 驱动训练、模型评价机制。
4. 第四章为实验部分。详细介绍实验的软硬件条件、实验的具体流程和可视化的实验结果。共分为三小节：实验条件、实验流程和实验结果。
5. 第五章则为讨论和总结。主要为本文工作的总结性分析和展望。

第二章 理论基础

第一节 图卷积神经网络

2.1.1 欧式空间的卷积

2012 年 Alex、Hinton 等人提出的 AlexNet^[AlexNet] 吹响了卷积神经网络的号角。时至今日，卷积神经网络在深度学习领域取得了惊人的发展，如对图像进行语义分割、达到像素级预测的 FCN^[FCN]，用于视频或者音频分类的 Slow-Fast 架构^[VideoSlowFast, AudioSlowFast] 和迁移学习的研究^[Transferable] 等等。这些成果都是通过在欧式空间定义卷积操作，在同一个卷积层内通过共用的卷积核对邻居的信息进行聚合，再通过非线性变换来连接、堆叠多个卷积层从而对欧式空间中的数据进行特征提取，最后将提取到的特征运用于各种任务，如图像分类、音频分类等。这些在欧式空间数据上定义的卷积满足了平移不变性和局部性 (见图 2.1(a),(b))，因而实现了同层共用一个卷积算子学习相同的局部特征。多卷积层的堆叠则将各层学习到的特征结合起来，得到最终的特征映射。

2.1.2 图卷积

对于非欧空间的数据，比如图数据，由于其本身不具备平移不变性和局部性 (见图 2.1(c),(d))，不能简单地从欧式空间的卷积神经网络中移植过来卷积的定义方法，必须在图结构上重新定义卷积。图卷积的定义方式有两种：谱方法^[kipfGCN, SGC, ChebyNet] 和空间方法^[GAT, GraphSage, ConfGCN]。谱方法根据图上的卷积定理，通过谱域定义卷积再从谱域变换会时域；空间方法则着眼图的拓扑结构，通过定义各种聚合函数完成卷积。本文关注谱方法下的图卷积神经网络。

(1) 卷积定理: 信号卷积的傅立叶变换等价于信号的傅立叶变换的乘积：

$$F(f \times g) = F(f) \cdot F(g) \quad (2.1)$$

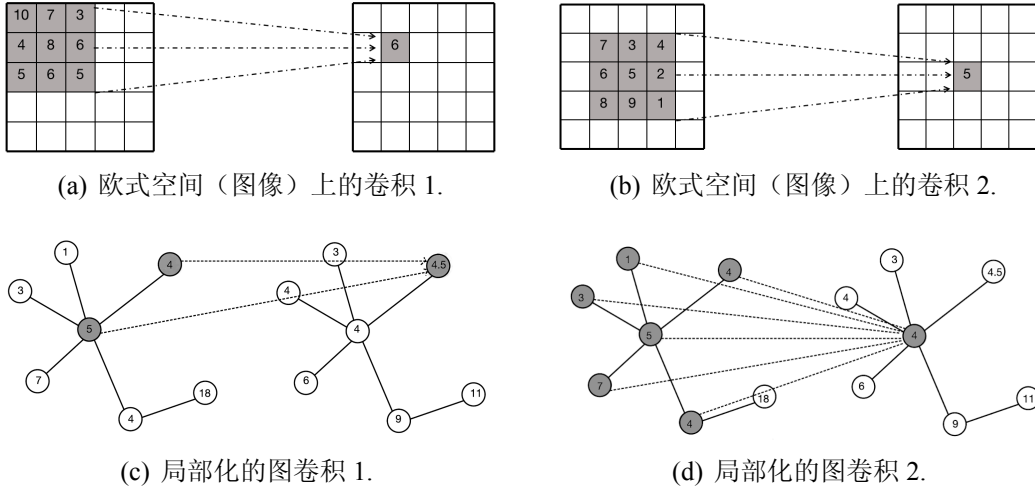


图 2.1 图像上的卷积与图卷积。第一排的两张图片展示了 CNNs 中的卷积。可以看到 (a) 和 (b) 中卷积算子 (灰色部分) 规模相同, 且天然具有聚合邻居的作用, 满足了平移不变性和局部性; 第二排的两张图片则展示了局部化的图卷积, 显然在不同位置的卷积算子规模不同, 导致这样的图卷积不满足平移不变性。

由此可得:

$$f \times g = F^{-1}(F(f) \cdot F(g)) \quad (2.2)$$

其中 $F(\cdot)$ 为图上的傅立叶变换, $F^{-1}(\cdot)$ 则表示图上的傅立叶逆变换。

(2) 图上的傅立叶变换、傅立叶逆变换: 根据谱图理论, 图上的傅立叶变换是以归一化的图的拉普拉斯矩阵 \mathbf{L} 的特征向量为基底的变换, 其中 $\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ 。对 \mathbf{L} 进行特征分解 (或称谱分解) 有:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \quad (2.3)$$

其中, \mathbf{U} 为特征向量组成的矩阵, 每一列为一个特征向量 u_i ; $\mathbf{\Lambda}$ 则为以特征值为作为对角线处值的对角矩阵。在此基础上, 傅立叶变换将图上的原始信号 x 转换为谱域下的信号 \hat{x} 的过程表达为:

$$\hat{x} = \mathbf{U}^T x \quad (2.4)$$

同时可以得到, 傅立叶逆变换将图上的谱域信号 \hat{x} 转换为原始信号 x 的过程表

达为:

$$x = \mathbf{U}\hat{x} \quad (2.5)$$

(3) 基于卷积定理的图卷积: 至此, 我们可以利用卷积定理定义图卷为:

$$x *_G y = \mathbf{U}((\mathbf{U}^T x) \circ (\mathbf{U}^T y)) \quad (2.6)$$

不妨把公式 2.6 中的 x 作为输入信号, $\mathbf{U}^T y$ 作为卷积核。为方便之后模型的公式化表达, 我们对卷积核进行参数化, 具体做法为: 把向量 $\mathbf{U}^T x$ 和向量 $\mathbf{U}^T y$ 的哈达玛积转化为对角矩阵 g_θ 和向量 $\mathbf{U}^T x$ 的乘积, 则公式 2.6 转化为:

$$GConv(x) = \mathbf{U} g_\theta \mathbf{U}^T x \quad (2.7)$$

其中, $GConv(\cdot)$ 表示图卷积算子, g_θ 表示新的、经过参数化表示的卷积核。

(4) 利用切比雪夫多项式参数化卷积核: 为了避免对图的拉普拉斯矩阵 \mathbf{L} 进行特征分解时带来的巨大计算量, ChebyNet^[ChebyNet] 利用正则化的特征值矩阵 $\hat{\Lambda}$ 的切比雪夫多项式继续参数化卷积核 g_θ 并结合公式 2.3 有如下推导:

$$\begin{aligned} GConv(x) &= \mathbf{U} g_\theta \mathbf{U}^T x \\ &= \mathbf{U} \left(\sum_{k=0}^K \theta_k T_k(\hat{\Lambda}) \right) \mathbf{U}^T x \\ &= \left(\sum_{k=0}^K \theta_k T_k(\hat{\mathbf{L}}) \right) x \end{aligned} \quad (2.8)$$

其中, θ_k 是需要学习的网络参数, $T_k(\cdot)$ 表示切比雪夫多项式的第 k 项, $\hat{\Lambda} = \frac{2\Lambda}{\lambda_{\max}} - \mathbf{I}_n$, $\hat{\mathbf{L}} = \frac{2\mathbf{L}}{\lambda_{\max}} - \mathbf{I}_n$ 。

(5) 对 ChebyNet 的一阶近似: 公式 2.8 中的 K 表示卷积算子涵盖的范围为 K 跳, 即每个节点会同 K 跳内的节点做卷积。Kipf 等人基于 ChebyNet 的参数化方法卷积算子, 构建了 ChebyNet 的一阶近似, 即一阶梯图卷积神经网络 (GCN)^[kipfGCN], 具体方法见下。

首先约束 $K = 1$ ，且 $\lambda_{max} = 2$ ，则可以得到：

$$GConv(x) = \theta_0'x + \theta_1'(\mathbf{L} - \mathbf{I}_n)x = \theta_0'x - \theta_1'\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \quad (2.9)$$

为了减少计算量并防止可能的过拟合，我们令 $\theta = \theta_0' = -\theta_1'$ 于是公式 2.9 化简为：

$$GConv(x) = \theta(\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})x \quad (2.10)$$

此外，为了维持数值稳定，防止出现梯度爆炸和梯度消失的问题，我们对公式 2.10 作出如下正则化：

$$\mathbf{I}_n + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \rightarrow \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}} \quad (2.11)$$

其中 $\tilde{\mathbf{A}} = \mathbf{I}_n + \mathbf{A}$ ， $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$

最终我们得到 GCN^[kipfGCN] 中的卷积算子为：

$$GConv(x) = (\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}})x\theta \quad (2.12)$$

将卷积算子延伸至多通道情况下，可以得到 GCN 在第 m 层的前向传播公式为：

$$\mathbf{X}^{m+1} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}^m\Theta^m) \quad (2.13)$$

其中， σ 为层间的激活函数 (*ReLU, sigmoid, tanh* 等)， $\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ 是非参数化的一阶近似卷积核， Θ^m 则为需要学习的第 m 层的网络参数。GCN 的模型结构如下图所示。

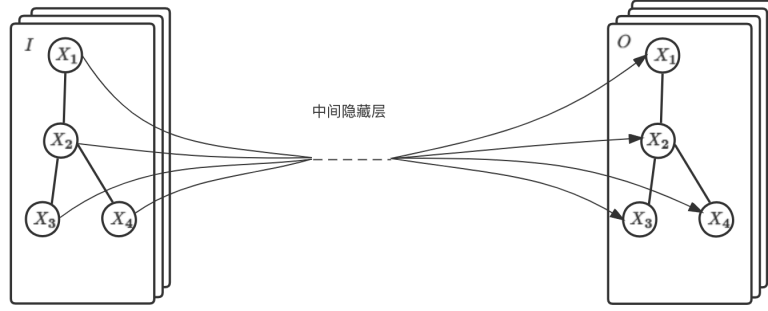


图 2.2 GCN 模型结构图例； I 表示输入通道数目即输入的图中每个节点的特征向量维度， O 则表示输出通道数；输入的图结构在所有层中是共享的，因此图卷积核在各层也是共享的；

第二节 循环神经网络

上节我们详细地介绍了图卷积和欧式空间卷积的不同之处和定义的难点，并讲述了本文关注的图卷积神经网络 GCN 中一阶近似图卷积 (公式 2.12) 的理论推导过程。本节我们将详细阐明 EvolveGCN 中用于驱动参数训练的部分—循环神经网络。

2.2.1 序列模型和潜变量的引入

现实生活中存在着具有时序结构的数据，比如按照单词顺序排列而成的句子、按照时间顺序呈现的视频或者音频、按照时间顺序排列的银行交易数据等等。序列模型就是针对这样的数据设计的用来接受某一序列作为输入，并预测该序列后续模型。循环神经网络便是当今使用最为广泛、效果最好的的序列模型。

序列模型同非序列模型的本质区别在于，序列模型中当前的数据同先前观察到的数据是相关的。不妨假设我们在 t 时刻观察到数据 x_t ，共有 T 个时刻的数据组成序列 \mathbf{x} ，那么序列 \mathbf{x} 出现的概率为 $p(\mathbf{x})$ 。由于 x_t 之间具有如上所述的相关性，根据条件概率公式：

$$p(\mathbf{x}) = p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_1, x_2) \cdot \dots \cdot p(x_T|x_1, \dots, x_{T-1}) \quad (2.14)$$

上式中的某一项表示序列中某一时刻的观察值 x_t 的概率，对该项中的历史信息

x_1, x_2, \dots, x_{t-1} 建模有:

$$p(x_t|x_1, x_2, \dots, x_{t-1}) = p(x_t|f(x_1, x_2, \dots, x_{t-1})) \quad (2.15)$$

其中的 $f(\cdot)$ 表示对过去信息的建模。潜变量模型通过引入潜变量 h_t 来表示 t 时刻之前的历史信息 $f(x_1, x_2, \dots, x_{t-1})$:

$$p(x_t|x_1, x_2, \dots, x_{t-1}) = p(x_t|h_t) \quad (2.16)$$

因此基于潜变量的序列模型 (循环神经网络) 可以描述为: 接受一个序列 $\mathbf{x} = x_1, x_2, x_3, \dots, x_T$ 作为输入, 模型运行至 t 时刻时, 通过映射 $F(\cdot)$ 将上一个时刻的潜变量 h_{t-1} 和当前的输入信息 x_t 映射到 t 时刻的潜变量 h_t , 然后根据 h_t 预测 t 时刻的结果 $y_t = p(x_t|h_t)$ 。其中, 组合上一时刻的潜变量和当前时刻的信息的映射 $F(\cdot)$ 成为了不同类型循环神经网络的主要区别。接下来我们将针对这方面的不同来说明基础的 RNN-Vanilla RNN 以及 GUR 和 LSTM 的原理。

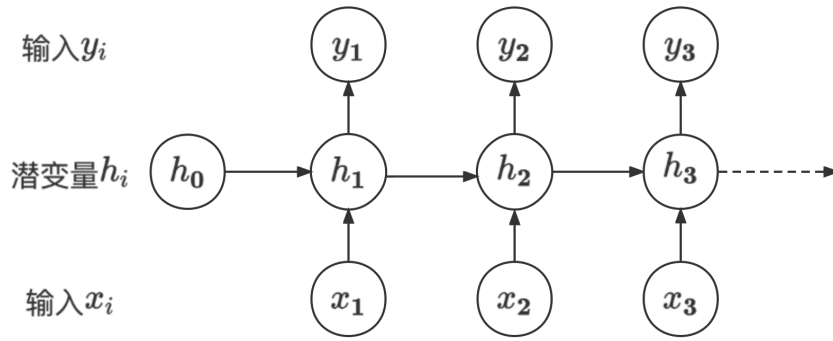


图 2.3 各种 RNNs 宏观基本结构

2.2.2 Vanilla RNN

Vanilla RNN 是最简单明了的 RNN, 它通过对 h_{t-1} 和 x_t 做线性变换得到潜变量 h_t 。

$$h_t = \Phi(W_{hh}h_{t-1} + W_{hx}x_t + b_h) \quad (2.17)$$

同样的，输入 $y_t = p(x_t|h_t)$ 也是简单地对 h_t 进行线性变化得到。

$$y_t = \Phi(W_{hy}h_t + b_y) \quad (2.18)$$

其中， Φ 表示非线性的激活函数， W_{**} 表示权重， b_* 则表示偏移。

2.2.3 LSTM 和 GRU

相比 Vanilla RNN，LSTM 和 GRU 则复杂得多。它们通过引入若干门和候选状态对历史信息进行更新、遗忘和筛选，从而使得新的潜变量 h_t 具有更为精细、准确的历史信息。其详细流程分述如下。

(1)LSTM: 长短时记忆 (Long Short Term Memory) 又称 LSTM，它在 Vanilla RNN 的基础上引入了和隐变量形状相同的记忆单元 C_t 来记录额外的信息，并通过引入输入门 I_t ，遗忘门 F_t ，输出门 O_t 来控制记忆单元筛选历史信息。各个门的计算方法为：

$$I_t = \sigma(x_t W_{xi} + h_{t-1} W_{hi} + b_i) \quad (2.19)$$

$$F_t = \sigma(x_t W_{xf} + h_{t-1} W_{hf} + b_f) \quad (2.20)$$

$$O_t = \sigma(x_t W_{xo} + h_{t-1} W_{ho} + b_o) \quad (2.21)$$

其中， σ 是非线性激活函数 *ReLU*，输入门 I_t 控制哪些信息输入到记忆单元 C_t ，输出门 O_t 负责配合记忆单元 C_t 得到 h_t ，遗忘门 F_t 则负责重置 C_t 中的内容。此外，LSTM 还包括候选记忆单元 \tilde{C}_t 。它同三个门的计算方式类似，只是采用了 *tanh* 作为激活函数，其计算公式为：

$$\tilde{C}_t = \tanh(x_t W_{xc} + h_{t-1} W_{hc} + b_c) \quad (2.22)$$

有了上述三个门以及候选记忆单元，LSTM 便可以得到新的记忆单元 C_t 和隐变

量 h_t 。某个步骤内完整的 LSTM 流程见图2.4。

$$C_t = F_t \circ C_{t-1} + I_t \circ \tilde{C}_t \quad (2.23)$$

$$h_t = O_t \circ \tanh(C_t) \quad (2.24)$$

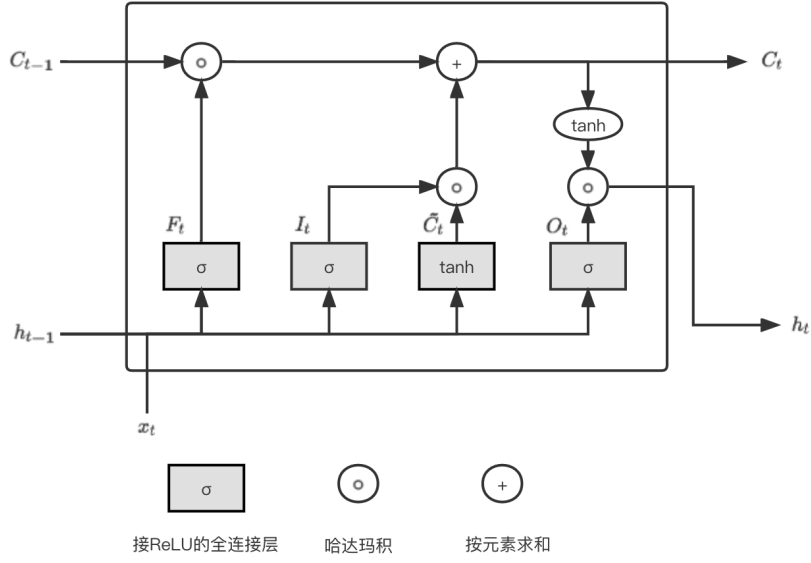


图 2.4 LSTM 流程图

(2)GRU: 门控循环单元 (Gated Recurrent Unit) 又称 GRU, 它同 LSTM 类似, 通过引入候选潜变量 \tilde{h}_t 来记录额外的信息, 并通过重制门 R_t 和更新门 Z_t 来控制候选潜变量筛选历史信息。重制门和更新门的计算方式为:

$$R_t = \sigma(x_t W_{xr} + h_{t-1} W_{hr} + b_r) \quad (2.25)$$

$$Z_t = \sigma(x_t W_{xz} + h_{t-1} W_{hz} + b_z) \quad (2.26)$$

和 LSTM 中的候选记忆单元类似, GRU 具有候选潜变量 \tilde{h}_t ,

$$\tilde{h}_t = \tanh(x_t W_{xh} + (R_t \circ h_{t-1}) W_{hh} + b_h) \quad (2.27)$$

其中的 $R_t \circ h_{t-1}$ 部分实际上是对历史信息进行了筛选：当 R_t 接近 1 时，公式 2.27 等价于公式 2.17，GRU 将近似退化为 Vanilla RNN；而当 R_t 接近 0 时， \tilde{h}_t 则完全不考虑历史信息，只考虑当前输入 x_t 。根据候选潜变量 \tilde{h}_t 以及更新门，我们可以得到新的潜变量如下：

$$h_t = Z_t \circ h_{t-1} + (1 - Z_t) \circ \tilde{h}_t \quad (2.28)$$

当更新门 Z_t 接近 1 时，模型则保留原有状态 h_{t-1} ；而当其接近 0 时，模型则倾向选择候选潜变量 \tilde{h}_t 作为新的潜变量 h_t 。某个步骤内完整的 GRU 流程见图 2.5。

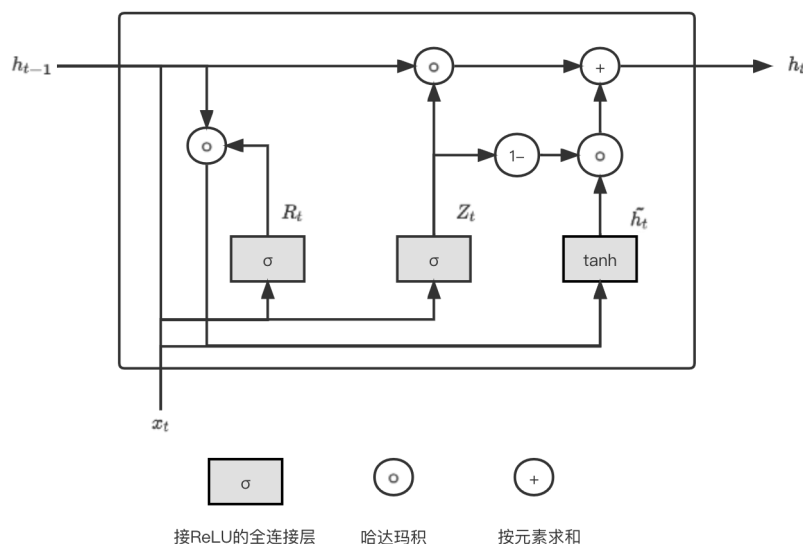


图 2.5 GRU 流程图

第三节 半监督学习

机器学习方法可以按照样本的标记情况分为三类：所有样本数据都无标签的无监督学习、所有样本数据都有标签的监督学习以及本节的主要讨论对象、仅部分样本数据有标签的半监督学习。

无监督学习的样本数据不具有标签，因此只能从样本数据的分布和它们之

间的关系着手，将样本划分至不同簇（聚类）或者是给出高维样本的低维表示（降维）。典型的无监督算法有：聚类算法中的 K 均值（K-means）和降维算法中的主成分分析（Principle Component Analysis）等。

监督学习则是机器学习领域中发展最成熟、最为迅速的一类算法，它利用有标签的样本数据训练出从样本数据到标签的映射，从而对模型未见过的样本数据的标签进行预测。典型的监督学习算法有：线性回归 (Linear Regression)、线性判别分析 (Linear Discriminative Analysis)、Logistic 回归 (Logisitic Regression)、支持向量机 (Support Vector Machine)、朴素贝叶斯 (Naive Bayes)、决策树 (Decision Tree) 以及多层感知机 (Multi-Layer Perceptron) 等。

实际问题中，海量样本数据是容易获得的，然而由于样本数据的标记工作专业性要求高、工作量巨大，完成对这些海量样本数据的标记需要消耗大量时间、人力等资源。因此有标签的样本数据和无标签的样本数据往往是同时存在的，而且无标签样本数据常占很高的比例。为了更充分地利用样本数据、减少数据和资源的浪费并提高模型的效果，人们尝试把大量没有标签的样本数据同少数有标签的样本数据一起注入到模型的训练过程中，半监督学习方法由此应运而生。结合第一章第二节对问题的描述，本文拟完成的反洗钱任务的样本数据表现为小部分样本数据有标签、大部分样本数据无标签，因此应该采用半监督学习的模型。下面介绍第二章第一节中的 GCN 是如何进行半监督学习的。

从公式2.12可以看出 GCN 的卷积核 $\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}} \in \mathbb{R}^{n*n}$ 是非参数化的，即它只取决于整张图的拓扑结构（邻接矩阵）。这样的特点使得我们能够利用所有样本数据得到全局共享的卷积核，接着对所有样本的特征映射 $\mathbf{X} \in \mathbb{R}^{n*d}$ 依次进行卷积操作、仿射变换和非线性激活操作。至此 GCN 完成了一次前向传播 (Forward) 而没有关心样本数据的标记情况。GCN 完成前向传播转而进行后向传播 (Backward) 时则需要考虑样本数据的标记情况：只利用带有标签的样本对应的输出层特征映射和标记情况进行损失的计算和网络参数的更新。

综上所述，GCN 利用全体样本数据的邻接矩阵和特征映射完成了前向传播，而利用带有标签的样本数据完成了后向传播从而实现半监督学习，其训练过程的伪代码如下：

算法 2.1 GCN 训练过程伪代码

Inputs:模型层数 m 模型 $model$ 所有样本形成的邻接矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 所有样本的特征映射 $\mathbf{X}^0 \in \mathbb{R}^{n \times d}$ 第一层网络参数 Θ^0 损失函数 \mathbf{L} 所有样本的标记情况 $Labels \in \mathbb{R}^n$ 带有标签的训练样本的筛选器 $train_mask \in \mathbb{R}^{n_{train}}$

- 1: 由 \mathbf{A} 得到全局共享的卷积核 $\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$
 - 2:
 - 3: **% Forward:**
 - 4: **for** i **in** $range(m)$ **do**
 - 5: $\mathbf{X}^{i+1} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{X}^i \Theta^i)$
 - 6: **end for**
 - 7:
 - 8: **% Backward:**
 - 9: 输出的特征映射: $\mathbf{X}^m \in \mathbb{R}^{n \times d^m}$
 - 10: 筛选出带有标签的训练样本的特征映射: $\mathbf{X}_{train}^m = \mathbf{X}^m[train_mask]$
 - 11: 筛选出带有标签的训练样本的标记情况: $Labels_{train} = Labels[train_mask]$
 - 12: 利用带有标签的训练样本计算损失: $Loss = \mathbf{L}(\mathbf{X}_{train}^m, Labels_{train})$
 - 13: $Loss.backward()$
 - 14: $Update(model)$
-

第四节 本章小结

本章对 EvolveGCN 所需的理论基础进行了介绍。详细地说明了图卷积神经网络 GCN 和循环神经网络 GRU、LSTM 的原理, 并介绍了 GCN 进行半监督学习的运行。接下的一章我们将以此为基础, 详细介绍 EvolveGCN 的模型原理。

第三章 模型原理

第一节 数据存储和处理方式

1. 按时间划分；多重边的合并；有向图转为无向图；稀疏矩阵的存储；

第二节 RNN 驱动训练

上图和解析；

算法 3.1 EvolveGCN 单元的运算过程

Inputs:

前一时刻 GCN 的参数, 亦即前一时刻 RNN 的隐藏单元 Θ_{t-1}^l

当前时刻 GCN 的输入特征映射, 亦即当前时刻 RNN 的输入单元 \mathbf{X}_t^l

- 1: 结合公式 (2.25) 至 (2.28) 为一个 GRU 单元的操作 $gru(input, previous_hidden)$, 使用 $gru(.)$ 对参数进行更新有:
 - 2: $\Theta_t^l = gru(\mathbf{X}_t^l, \Theta_{t-1}^l)$
 - 3: 利用公式 (2.13) 对 \mathbf{X}_t^l 进行图卷积操作:
 - 4: $\mathbf{X}_t^{l+1} = Gconv(\mathbf{X}_t^l, \Theta_t^l)$
-

第三节 模型评价机制

分类别的 precision、recall 和 f1；着重关注黑标节点的 precision

第四章 实验

算法 4.1 EvolveGCN 参数对齐的方式 (以第一层、第一时刻为例)

Inputs:

第 1 层第 1 时间步的输入的特征映射: $\mathbf{X}_1^1 \in \mathbb{R}^{n*d}$

第 1 层第 1 时间步输出的特征映射的列数 d_1

Outputs:

对齐后 GRU 的输入特征映射: $\mathbf{W}_1^1 \in \mathbb{R}^{d*d_1}$

- 1: **function** : $\mathbf{W}_1^1 = transfer(\mathbf{X}_1^1, d_1)$
 - 2: 初始化打分向量:
 - 3: $scores \in \mathbb{R}^d$;
 - 4: 以 $-stdv$ 和 $stdv$ 为左右边界, 按均匀分布得到 $scores$ 向量:
 - 5: $scores \sim uniform(-stdv, stdv)$, 其中 $stdv = \frac{1}{\sqrt{d}}$
 - 6: 计算辅助向量: $y = \frac{\mathbf{X}_1^1 * scores}{\|scores\|}$, 其中 $y \in \mathbb{R}^n$
 - 7: 在 y 中筛选出 d_1 行有: $indices = top - indices(y, d_1)$
 - 8: 对输入做变换有: $\mathbf{W}_1^1 = [\mathbf{X}_1^1 * tanh(y)]_{indices}$
-

第一节 实验条件

第二节 实验流程

第三节 实验结果

第五章 讨论和总结

1. 优点：适用于动态图，把时序信息融合在训练过程中
2. 缺点：基于 GCN，使得其必然是 Transductive 的，且对内存有极高要求

致 谢

乌飞兔走，白驹过隙；时光荏苒，岁月如歌。

转眼间我和南开的故事就要画上句号了，四年时光里有过入学前的憧憬和激动，也有过重复中的枯燥、单调和乏味；有过输掉第一场篮球比赛的落寞和不甘，也有过乘胜追击时的酣畅淋漓；有过第一个学期的迷茫和困惑，也有过保研成功、金榜题名的欣慰和满足；有过收到‘好人卡’的苦涩和自卑，也最终收获了一段平淡幸福，充满希望的感情；有过深夜的酒，也有过清晨的书。有过酸甜，有过苦辣，平凡而又充实。

感谢父母和家人们的多年来的照料和教诲；感谢南开大学给了我更高的平台、更广阔的视野；感谢软件学院各位老师们的悉心教导，让我进入到蓬勃发展的计算机领域；感谢女友的一路支持、鼓励和陪伴；感谢各位球友一起挥汗如雨，享受运动的快乐；感谢为我提供未来学习、科研机会的中科院计算所；感谢所有为我本科时期的学习、生活提供便捷的工作人员和同学们。

最后，感谢不曾停止努力的自己！

个人简历

基本信息:

姓名: 王子旭

性别: 男

出生日期: 2000 年 03 月 23 日

通信地址: 南开大学软件学院/中国科学院计算技术研究所

电话: 15631008979

E-mail: 1813033@mail.nankai.edu.cn

教育背景:

2018.09-2022.07 南开大学 软件学院 软件工程 学士