**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

专 业： 软件工程

系 别： 软件工程

学 院： 软件学院

指导教师： 刘明铭

程学旗

完成日期： 2022年5月\*\*日

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 论文《EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs》，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。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：反洗钱；图卷积神经网络；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%∼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 paper *EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs* proposed in AAAI2020, I construct a dynamic graph convolutional network 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.

**Keywords**: graph convolution network; anti money launder; semi-supervised learning

目 录

[目 录 III](#_Toc1116987853)

[第一章 引言 1](#_Toc2060286944)

[第一节 研究背景和意义 1](#_Toc1216343580)

[第二节 问题描述 1](#_Toc1189713267)

[第三节 文献综述【是否要和问题描述换位置】 2](#_Toc290641252)

[第四节 主要工作 3](#_Toc1429709086)

[第五节 章节介绍 3](#_Toc926082119)

[第二章 理论基础 4](#_Toc1848184224)

[第一节 图卷积神经网络 4](#_Toc1228782560)

[第二节 循环神经网络 8](#_Toc1945736368)

[第三节 半监督学习 12](#_Toc110160460)

[第四节 本章小结 14](#_Toc335947506)

[第三章 EvolveGCN的模型原理 15](#_Toc535225379)

[第一节 数据处理 15](#_Toc1871431217)

[第二节 模型结构和训练方式 16](#_Toc1098970157)

[第三节 评价机制 19](#_Toc2032064499)

[第四节 本章小结 20](#_Toc1475596452)

[第四章 实验 21](#_Toc1208413208)

[第一节 实验条件 21](#_Toc1047937177)

[第二节 实验流程 21](#_Toc1166744792)

[第三节 实验结果 21](#_Toc806538387)

[第四节 本章小结 21](#_Toc573890445)

[第五章 讨论和总结 22](#_Toc1027650438)

[第一节 本文总结 22](#_Toc1657422292)

[第二节 进一步展望 22](#_Toc1286076407)

[参考文献 23](#_Toc663265394)

# 第一章 引言

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。

## 第一节 研究背景和意义

从海量的交易流水中检测出参与洗钱的账户是十分困难的，原因除了数据量极大之外，还有犯罪分子为躲避监测对洗钱行为进行了伪装和隐藏，比如通过合法地手段经过很多中间账户将黑钱分散开并最终汇聚到同一个目标账户、洗钱的过程中加入一些合法的交易以掩盖非法交易、快速地将黑钱转移至合法账户等等。为了能从海量地交易数据中高效地检测出涉嫌参与洗钱的账号，我们利用图卷积神经网络来对交易网络中的节点进行分类。

## 第二节 问题描述

反洗钱任务可以描述为：基于如下数据对所有账户进行二分类，即把账户分为涉嫌参与洗钱的黑标账户和未参与洗钱的白标账户。

1. 一周内的交易流水记录 edgelist.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。

1. 账户的标签 node\_labels.npy。这是一个形状为 2 ∗ 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。

|  |  |
| --- | --- |
| 节点个数 |  |
| 边个数 |  |
| 特征维度 |  |
| 带黑标的节点个数和占比 |  |
| 带白标的节点个数和占比 |  |
| 时间跨度 |  |
|  |  |

表 1.1 原始数据统计表

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

当前主流的反洗钱算法有：基于规则的方法、基于图挖掘技术的方法以及基于深度学习的方法。

基于规则的方法通过人工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等。【】HoloScope、【】FlowScope、【】EngineSpoke。

基于图挖掘技术的方法：Fraudar、BP

基于深度学洗的方法：GCN、SGC

## 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[[[1]](#endnote-0)]， 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. 利用循环神经网络RNN[[[2]](#endnote-1)]驱动一阶近似图卷积神经网络GCN[[[3]](#endnote-2)] 参数训练的方法融合了交易流水图的拓扑信息和时序信息。3. 将EvolveGCN同GCN以及简化的图卷积神经网络SGC[[[4]](#endnote-3)] 等方法进行了比较和分析。

## 第五节 章节介绍

本文共分为五章，各章的内容如下：

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。该部分详细介绍了反洗钱的任务的背景、具体要求和国内外的研究现状，并给出本文在该任务上的主要工作和各章内容的简述。

第二章为理论基础部分。主要介绍 EvovleGCN 的理论基础，包括三个小 节：图卷积神经网络、循环神经网络以及半监督学习原理。每小节详细介绍了对应原理的数理推导和发展过程，为接下来的模型原理介绍做好铺垫。

第三章为模型原理部分。具体介绍 EvolveGCN 的结构、训练方式、评价机

制和一些实现需要注意的细节，共分为三小节：数据存储和处理方式、RNN 驱动训练、模型评价机制。详细地展示了EvovleGCN是如何结合了动态图的拓扑结构信息和时序信息来完成训练的。

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。

第五章则为讨论和总结。对本文工作作出了总结分析和展望。

# 理论基础

本文关注的EvolveGCN[1]模型是一种以循环神经网络驱动图卷积神经网络训练的动态图卷积神经网络。因此在介绍其运行方式前，需要对图卷积神经网络和循环神经网络的理论基础进行详细的介绍。此外，本章还会讨论GCN[3]的半监督学习方法，为之后模型原理介绍做铺垫。

## 第一节 图卷积神经网络

下面我们介绍图卷积和欧式空间的卷积不同之处、基于谱方法的图卷积定义方式以及图卷积神经网络GCN的理论推导过程。

### 2.1.1 欧式空间的卷积

2012年Alex、Hinton等人提出的AlexNet[[[5]](#endnote-4)]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[[[6]](#endnote-5)](Fully convolutional networks), 用于视频或者音频分类的Slow-Fast架构和迁移学习的研究[Transferable] 等等。这些成果都是通过在欧式空间的数据上定义卷积操作，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(如ChebyNet、GCN、SGC等)和空间方法(如GraphSage、Graph Attention Networks等)。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。EvolveGCN选择使用实现较为简单的GCN作为模型的一部分，因此接下来我们将关注谱方法下的图卷积神经网络，并给出GCN以及SGC前向传播公式的推导过程。

（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×g* ) *= F*( *f* ) *F*( *g* ) (2.1)

|  |  |
| --- | --- |
| (a) 欧式空间（图像）上的卷积1 | (b) 欧式空间（图像）上的卷积2 |
| (c) 图卷积1 | (d) 图卷积2 |
| 图2.1 欧式空间上的卷积和图卷积的对比：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。 | |

对公式2.1两边同时做傅立叶逆变换得到

(2.2)

其中，为图上的傅立叶变换，则表示图上的逆傅立叶变换。

1. 图上的傅立叶变换：

根据谱图理论，图上的傅立叶变换是以归一化的拉普拉斯矩阵**L**的特征向量为基底的变换，图的归一化的拉普拉斯矩阵**L**定义为：

(2.3)

其中，为单位矩阵，为图的度矩阵，为图的邻接矩阵（0-1矩阵或者带权重的矩阵）。对于无向图来说**L**是对称矩阵，这表示它可以进行特征分解且它的特征向量互相正交、其特征向量组成矩阵是正交矩阵，因此对**L**进行特征分解（或称为谱分解）有：

(2.4)

为**L**的特征向量组成的矩阵，每一列为一个特征向量；则为以特征值为作为对角线处值的对角矩阵，。图上的傅立叶变换即为以中的特征向量为基底的变换，因此通过傅立叶变换将图上的原始信号x转换为谱域下的信号 的过程可以表达为：

= *x* (2.5)

同时可以得到，通过傅立叶逆变换将图上的谱域信号 转换为原始信号 *x*的过程：

(2.6)

1. 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积如下：

(2.7)

其中表示图卷积运算符， 表示哈达玛积，即同形状的矩阵按位相乘的运算。妨把公式(2.7)中的*x*作为输入信号，则可以作为卷积核。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：把向量和向量的哈达玛积转化为对角矩阵和向量的乘积，则公式(2.7)转化为：

(2.8)

其中表示图卷积算子，则表示使用对角矩阵的形式重新表达了的卷积核。

1. 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进行特征分解时带来的巨大计算量，ChebyNet[[[7]](#endnote-6)]利用正则化的特征值矩阵 的切比雪夫多项式继续参数化卷积核并结合公式(2.4)有如下推导：

(2.9)

其中，是需要学习的网络参数，(.)表示切比雪夫多项式的第*k*项，归一化的特征值矩阵，对应的， 表示归一化后的拉普拉斯矩阵**L**的最大特征值。

1. 对ChebyNet的一阶近似：

公式(2.9)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[3]，具体方法见下。

首先约束 *K* = 1，且 = 2，则可以得到：

(2.10)

为了减少计算量并防止过拟合，我们令 于是公式(2.10)可以化简为：

(2.11)

考虑到维持数值稳定，防止训练过程中出现梯度爆炸或者梯度消失的问题，我们对公式(2.11)作如下的正则化：

(2.12)

其中，+，。把公式(2.12)套入到公式(2.11)可以得到GCN中的卷积算子：

(2.13)

将得到的卷积算子延伸至多通道的情况，可以得到GCN在第*m*层的前向传播公式为：

(2.14)

其中(.) 为层间的非线性激活函数（ReLU，sigmoid，tanh等），是非参数化的一阶近似卷积核，为第*m*层需要学习的网络参数。综上所述，给出GCN结构如下如所示：

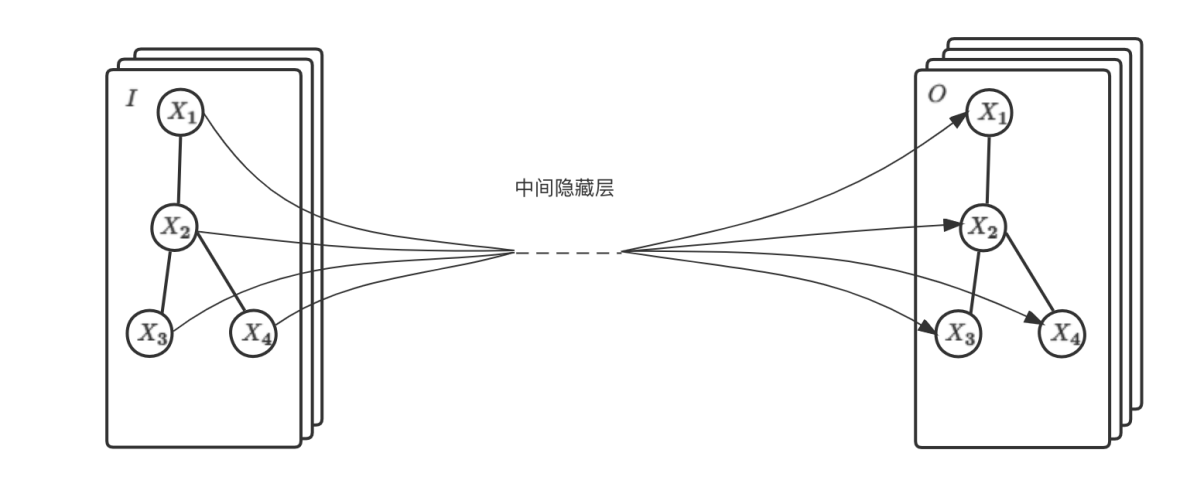


图 2.2 GCN 模型结构图例；I表示输入通道数目即输入的图中每个节点的特征向量维度，O则表示输出通道数；输入的图结构在所有层中是共享的，因此图卷积核在各层也是共享的；

## 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中一阶近似图卷积的理论推导过程。本节我们将详细阐明EvolveGCN中用于结合时序信息、驱动模型参数训练的部分–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，并预测该序列后续的模型。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的数据是相关的而不是相互独立的。不妨假设我们在 *t* 时刻观察到数据, 共有 *T* 个时刻的数据组成序列 ***x***，序列 ***x*** 出现的概率为 *p*(***x***)。由于 之间具有相关性，根据条件概率公式有：

(2.14)

上式中的第*t*项表示序列中*t*时刻的观察值的概率，对该项中的历史信息 建模有：

(2.15)

其中的 *f* (.) 表示对过去信息的建模。潜变量模型通过引入潜变量来表示*t*时刻之前的历史信息 ：

(2.16)

至此我们可以将基于潜变量的序列模型 (或者说循环神经网络) 可以描述为：模型依次接受一个序列 作为输入，运行至第*t*项时，通过映射*F*(.)将上一项对应的潜变量 和当前的输入信息 映射到第*t*项的潜变量，然后根据预测第*y项*的结果。其中，组合之前一项的潜变量和当前项的信息的映射*F*(.)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。

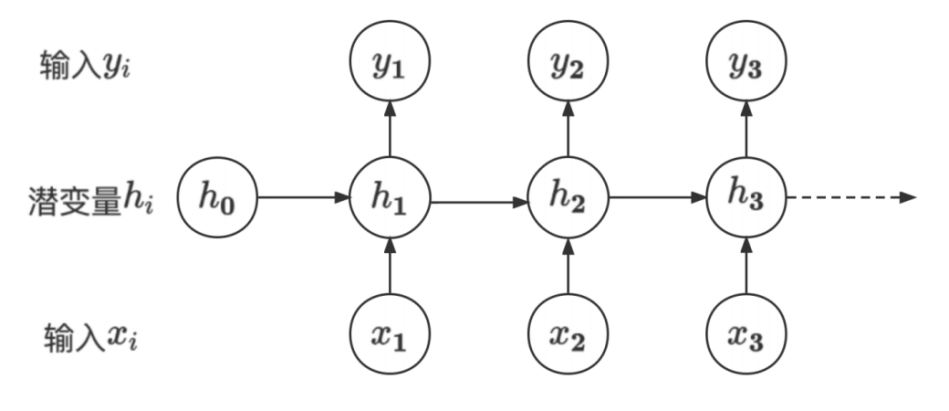


图 2.3 RNNs的基本通用结构示意图

### **2.2.2 Vanilla RNN**

Vanilla RNN 是最简单明了的RNN，它通过对和做线性变换得潜变量。

(2.17)

同样地，输出 也是简单地对进行线性变化得到。具体表达式如下公式2.18：

(2.18)

其中，表示非线性的激活函数，表示权重，则表示偏移。

### **2.2.2 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 ht 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来记录额外的信息，并通过引入输入门 , 遗忘门 , 输出门 来控制记忆单元筛选历史信息。各个门的计算方法为：

(2.19)

(2.20)

(2.21)

其中， 是非线性激活函数 ReLU，输入门控制哪些信息输入到记忆单元 ，输出门负责配合记忆单元 得到，遗忘门 则负责重置 中的内容。此外，LSTM 还包括候选记忆单元。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：

(2.22)

至此我们得到了输入门、输出门、遗忘门以及候选记忆单元，连同上一项的记忆单元，LSTM 得到新的记忆单元 和隐变量的方式见公式2.23和2.24。 某个步骤内完整的 LSTM 流程见图2.4。

新的记忆单元：

(2.23)

新的潜变量：

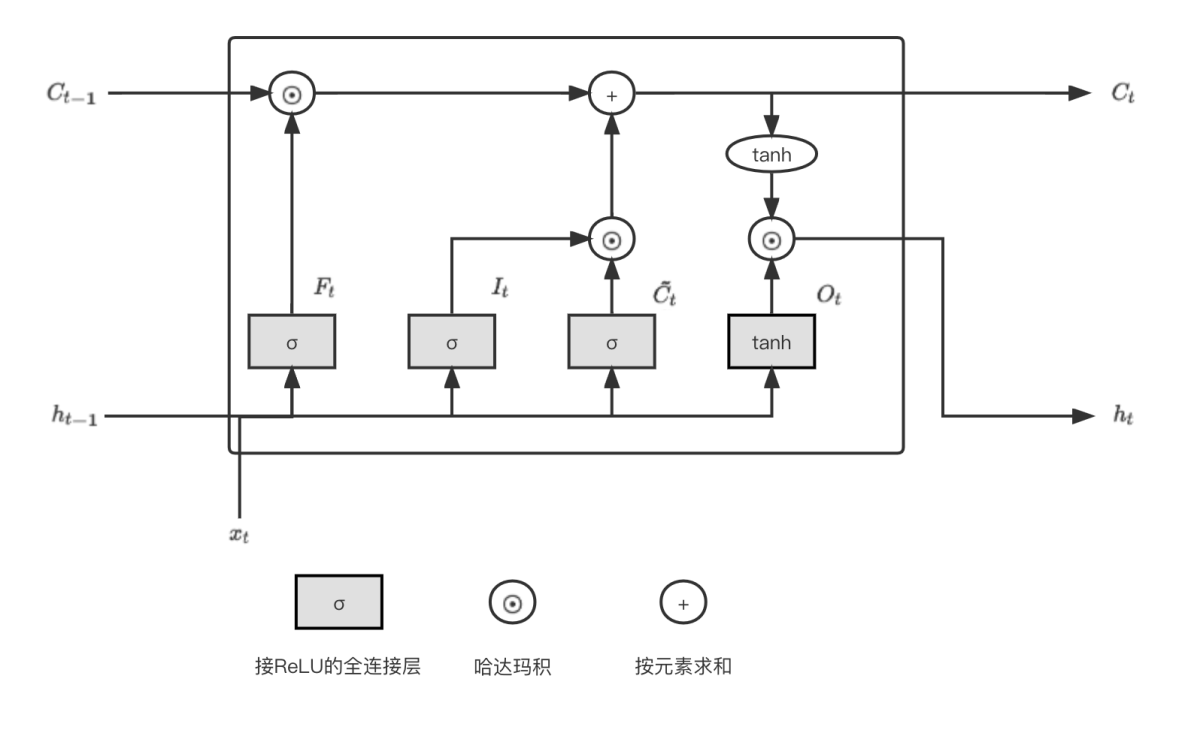
(2.24) 

图 2.4 LSTM一个单元内的流程图

(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 来记录额外的信息，并通过重制门 和更新门 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：*z*

(2.25)

(2.26)

和 LSTM 中的候选记忆单元类似，GRU 具有候选潜变量,

(2.27)

其中的部分实际上是进行了对历史信息的筛选：当 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 接近 0 时，则完全不考虑历史信息，只考虑当前输入。根据候选潜变量 以及更新门，我们可以得到新的潜变量如下：

(2.28)

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

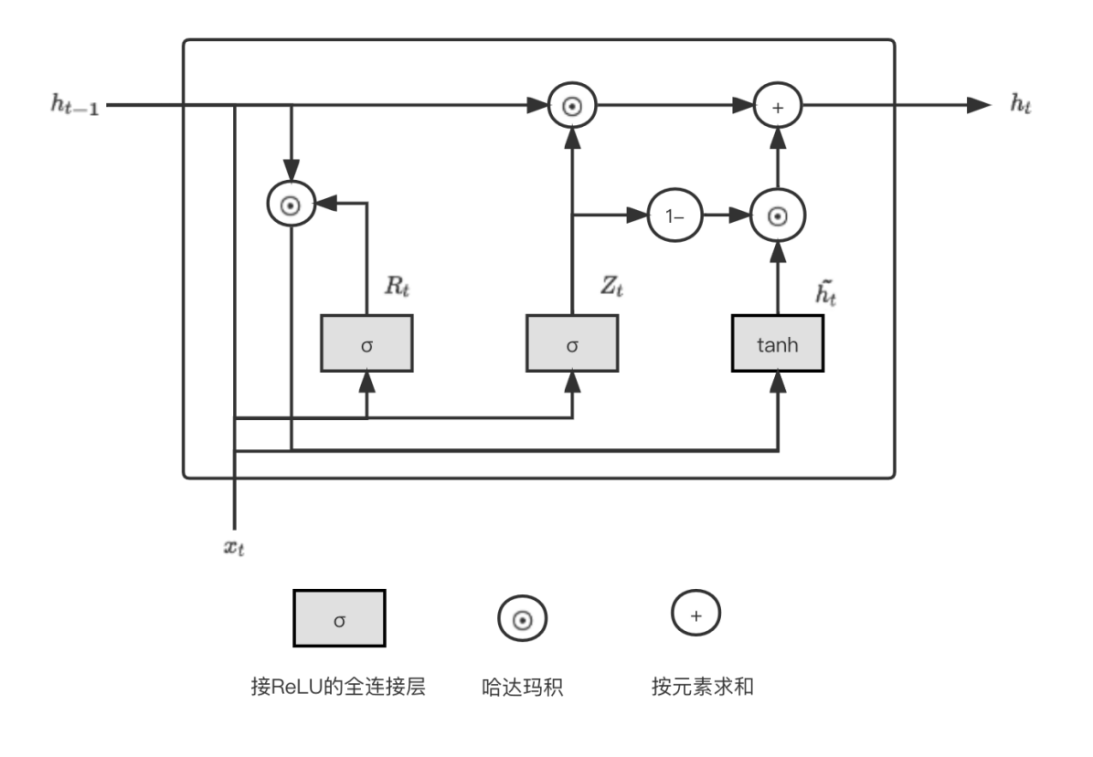


图 2.5 GRU一个单元内的流程图

## 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 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 **X** 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：

|  |
| --- |
|  |

## 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 进行半监督学习的训练过程。下一章我们将以此为基础，详细介绍 EvolveGCN 的模型原理。

# 第三章 EvolveGCN的模型原理

接下来将以第二章理论介绍为基础，详细说明EvolveGCN的模型原理，包括对原始数据的预处理方式以及模型的结构、训练方式以及评价机制。

## 第一节 数据处理

第一章第二节的问题描述给出了原始数据：交易流水信息edgelist.csv, 节点特征信息feats.csv以及节点标记信息labels.npy，同时列出了由于原始数据的特性所需要考虑的关键问题。本节介绍如何对原始数据进行处理以及处理方式对解决关键问题的作用。

### 3.1.1 对edgelist.csv按照时间分块

把流水记录edgelist.csv以相等的时间间隔依次分成若干个时间块，每个时间块内的交易流水信息可以构成一个子图，各时间块内交易流水的子图按照时间块顺序依次排列可以得到交易流水的子图序列。这样处理的好处是把一系列瞬时的交易行为通过分块的方式转化为各时间段内所有交易行为的综合体现，使得模型既能够针子图内的交易数据作出分析，也可以对子图间的带有时序信息的模式变换作出分析，从而更有效地捕捉到交易中的异常模式。

### 3.1.2 获取各时间块子图的邻接矩阵

为了对子图做图卷积来进行分析，我们使用带权重的邻接矩阵来表示子图，其中(*i, j*)位置的权重表示所在时间段内账户*i*和*j*交易金额的总和。由表\*\*\*中所示，交易网络较为稀疏，因此我们采用稀疏矩阵来实现。

在实现过程中，需要注意原先的图是有向图，其邻接矩阵是不对称的，无法作为GCN的输入。因此我们忽略交易的方向性，通过为每条边添加反向边的方式将原先的有向图转化为无向图，以获取对称的邻接矩阵。

### 3.1.3 获取各子图的节点掩码

为了快捷地获取各子图节点的特征映射，我们为每个子图求出节点掩码

某个子图的节点掩码为：，长度为所有节点的个数，

第i位表示编号为i的节点是否属于子图：True表示节点在子图中，False表示不在。

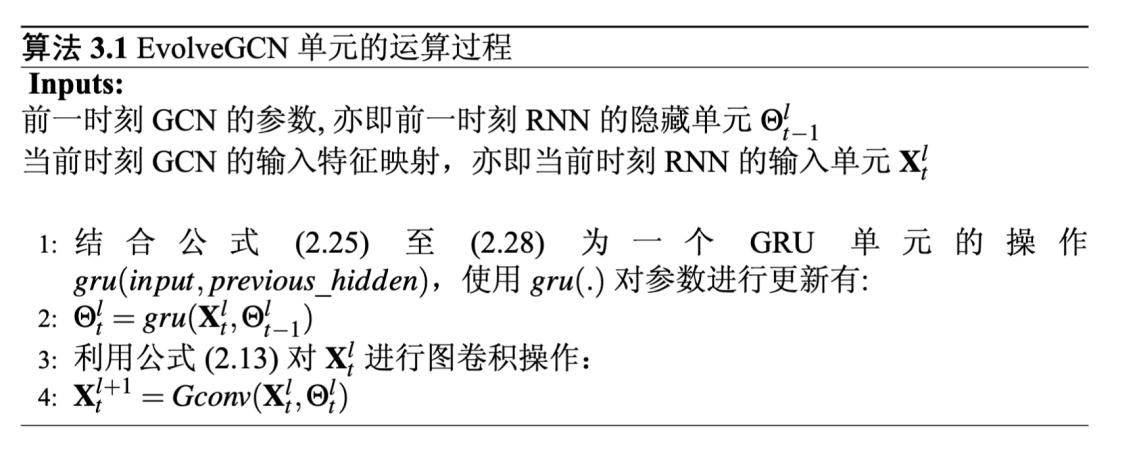
### 3.1.4 划分训练集、交叉验证集和测试集

由第二章第三节知所有账户参与前向传播、带有标签的账户参与模型的反向传播，因此需要对带标签的节点划分训练集、交叉验证集和测试集。三个集合中的样本比例：0.7 : 0.2 : 0.1，其中应尽量保持训练集中黑标和白标的比例接近1：1, 以减少样本比例失衡的影响。划分方式可以借鉴子图的节点掩码，同样利用长度为节点总数的列表来表示训练集掩码、交叉验证集掩码和测试集掩码。

## 第二节 模型结构和训练方式

本节将介绍如何利用上述经过预处理的数据，通过EvolveGCN后加一个多层感知机(MLP)来训练一个二分类器，完成反洗钱任务。

通过第一节的数据处理，我们将原先的超大交易流水图按照交易发生的时间分割成了若干子图组成的、带有时序信息的动态图序列。EvolveGCN对这些子图的操作分为两个方向：纵向采用GCN来学习子图的结构信息，在横向则把GCN中的网络参数作为隐藏单元（潜变量），采用RNN来学习子图序列中的时间信息。把两个方向的操作结合起来便可以形成EvovleGCN的一个单元，多个EvolveGCN单元在横向和纵向交织排列即可得到完整的EvolveGCN结构。其中EvolveGCN单元内的计算过程见算法3.1（鉴于GRU较为容易实现且效果同LSTM大致相同，我们采用GRU作为模型中的RNN单元）。



上述EvovleGCN单元按照时间顺序横向延展、按照层次顺序纵向延展得到EvolveGCN的整体结构如下图3.1。

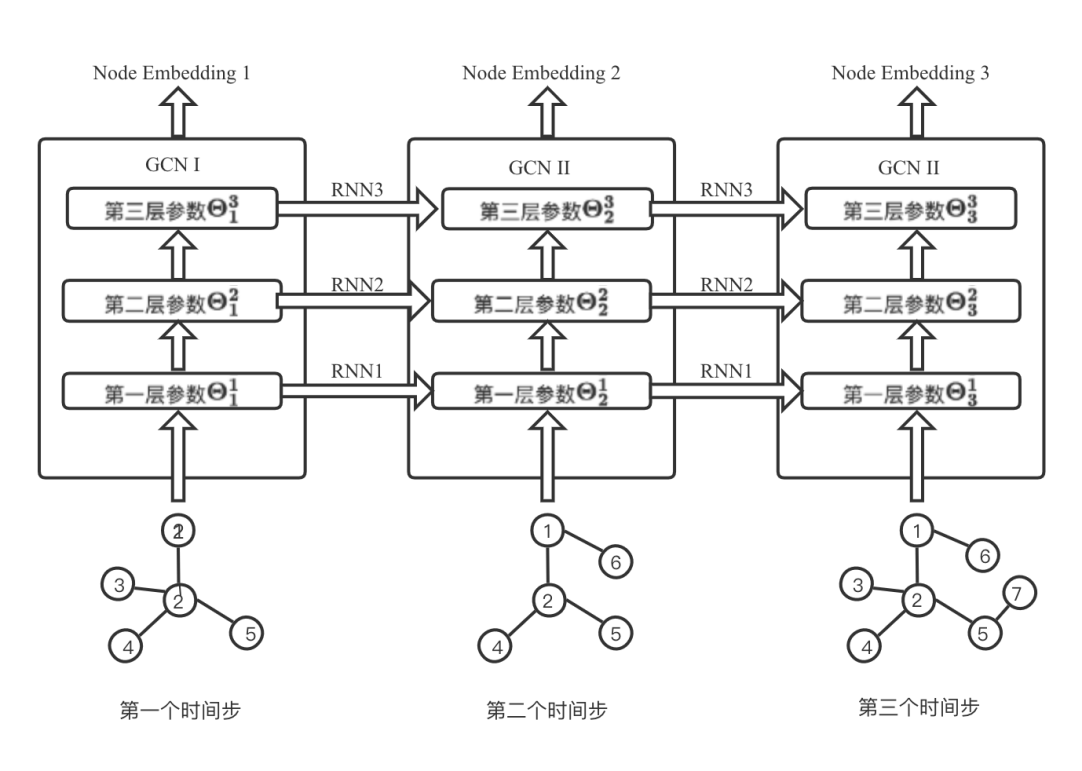


图 3.1 EvolveGCN宏观结构图；图中展示了前三个时间步数据在模型中的前向传播过程。纵向为各子图的GCN部分，参数中下标表示时间步数，上标表示层数；横向为RNN部分，RNN1表示以各时间步GCN的第1层参数作为隐藏单元的循环神经网络；Node- -Embedding 表示各时间步输出的特征映射。

接下来详细说明EvolveGCN通过前向传播得到各时间步的输出特征映射的具体流程，参照图3.2：

（1）从第一个时间步开始：取出所有节点的特征映射作为RNN1的第一项输入，并初始化作为RNN1前一项潜变量，带入2.2.2所述的LSTM或者GRU中得到新的潜变量--第一个时间步GCN的第一层的权重参数。取出邻接矩阵，根据公式2.11对正则化得到，求对应的度矩阵得到；带入公式2.13可以得到第一个时间步GCN的第一层输出：

1. 接着对第二个时间步做类似的工作：所有节点的特征映射作为RNN1的第二项输入、（1）中得到的作为RNN1前一项潜变量，经过LSTM或者GRU一个单元得到新的潜变量--；接着通过求的第二个时间步GCN的第一层输出，其中的计算同（1）中所述。
2. 重复（1）或（2）中的步骤，制止完成最后一个时间步的运算，至此我们将得到所有时间步GCN的第一层输出, T为总时间步数。
3. 初始化RNN的潜变量，并以第一层输出作为第二层输入，重复（1）（2）（3）的步骤，即可最终得到各个时间步的特征映射。

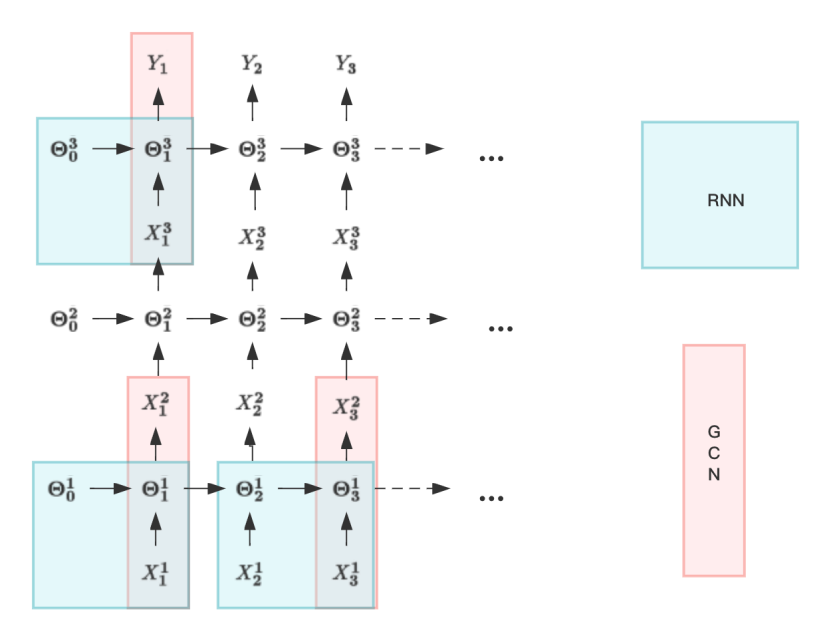


图 3.2 EvolveGCN参数变换示意图；左侧表示在前向传播过程中参数计算、更新过程。 右侧为图例，蓝色方块表示RNN的一个单元，红色块则表示GCN中一层。图中只标 识出部分RNN和GCN单元。

上述结构使得EvolveGCN只需要通过训练多层RNN即可完成所有GCN中各层网络参数的训练，而不需要单独训练GCN的参数，因此EvolveGCN的参数规模和多层RNN的规模一致。 将局部GCN的网络参数作为RNN的隐藏单元的设计在理论上是十分巧妙的，它使得模型在保持规模不过高的前提下实现了融合动态图数据的结构信息和时序信息的目的，可谓事半功倍。

最后，为了完成分类任务，需要对输出的特征映射做平均池化，将得到的池化过的特征输入到一个多层感知机(见图3.3)中得到二维的特征映射，表示第*n*个样本第0维特征。对按行做sofmax操作得到所有节点属于两个类别的概率矩阵**，**其中表示第n个样本属于第0类即黑标的概率，则表示该样本为白标的概率，取较大者作为分类结果。其中。

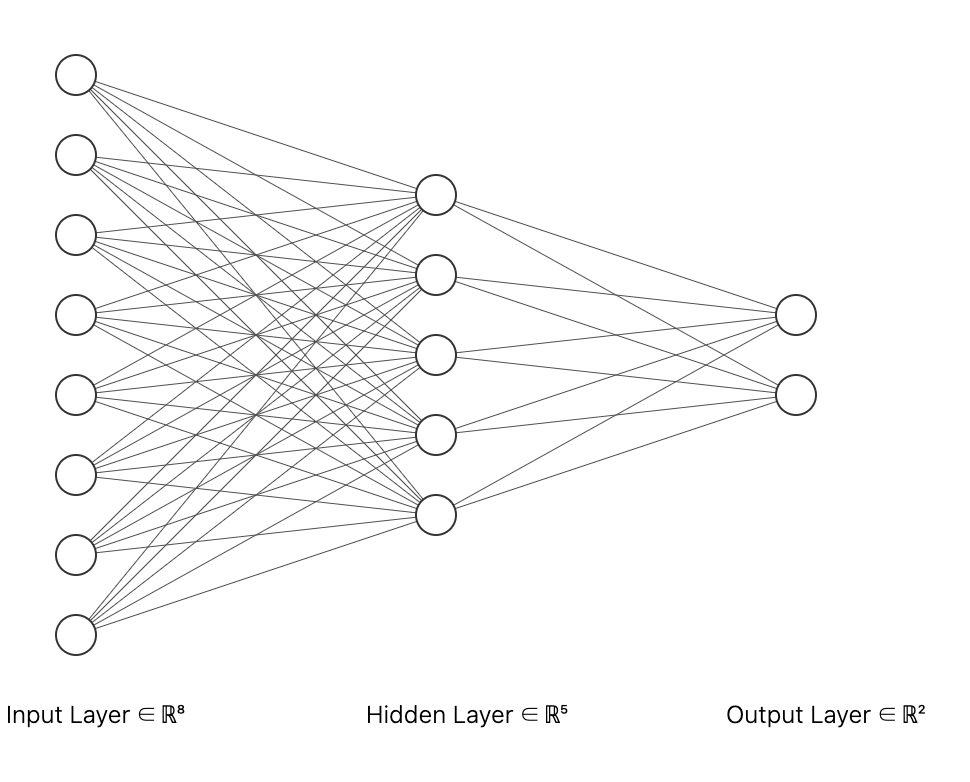


图 3.3 多层感知机的示意图

同第二章第三节中GCN的半监督学习过程，在完成前向传播后使用训练集掩码筛选出所有带有标签的训练集节点以及对应的标签，接着对这些节点的标签向量和模型预测的分类结果向量做交叉熵损失，再经过梯度下降更新模型的参数，从而完成一轮训练。训练多轮直至损失收敛。

## 评价机制

本节将介绍本文采用的主要评价机制。

### 3.3.1 混淆矩阵和混淆表格

混淆矩阵(Confusion Matrix)是监督学习算法中常用来可视化预测情况的结果统计矩阵。它统计了各类样本的真实分类以及预测分类情况，可以用于展示出样本被混淆的程度，是一系列监督学习算法评价指标的基础。

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 预测的分类情况 | |
| 黑标 | 白标 |
| 实际的分类情况 | 黑  标 | BB | BW |
| 白  标 | WB | WW |

表 3.1 反洗钱任务中分类结果的混淆矩阵；其中BB表示实际为黑标账户且被分类为黑标账户的样本个数；BW则表示实际为黑标账户且被分类为白标账户的样本个数；类似的有WB表示实际为白标账户但被分类为黑标账户的样本个数，WW表示实际为白标且被分类为白标的样本个数。

在得到混淆矩阵的前提下，指定分类任务中的正例后即可得到对应的混淆表格(Confusion Table)如下：

|  |  |  |
| --- | --- | --- |
|  | Classified as Positive | Classified as Negative |
| Positive  Sample | True Positive(TP):  Number of positive samples that are predicted as positive. | False Negative(FN):  Number of positive samples that are predicted as negative. |
| Negative  Sample | False Positive(FP):  Number of negative samples that are predicted as positive. | True Negative(TN):  Number of negative samples that are predicted as negative. |

表 3.2 确定正例样本后的混淆表格；

其中Positive Sample为正例，Negative Sample则为负例

其中True Positive表示实际为正例且预测为正例的个数，False Positive则表示实际为负例但预测为正例的样本个数；类似的，True Negative表示实际为负例且预测为负例的样本个数，False Negative表示实际实际为正例但预测为负例的样本个数。表3.2中标为绿色的部分True Positive和True Negative表示分类正确的部分，标为红色的False Positive和False Negative则表示分类错误的部分。

### 3.3.2 Accuracy、Error Rate、Precision和Recall

在指定正例并得到混淆表格后，我们可以定义如下评价指标：

1. 正确率Accuracy：

。表示所有分类正确的样本占所有样本的比例。

1. 错误率Error Rate：

。表示所有分类错误的样本占所有样本的比例。不难看出：Error Rate = 1- Accuracy。

1. 准确率Precision：

。表示被分类为正例的样本中有多少是正确的。

1. 召回率Recall：

。表示实际为正例的样本中有多少是被模型正确分类为正例。

在反洗钱任务中黑标样本和白标样本分布极其不均衡，在这样的情况下，占比较大的白标样本的评价指标会出现虚高情况。因此我们应该重点审视黑标为正例的情况下的各指标情况。

### 3.3.3 F1、PR曲线和ROC曲线

一般地，Precision和Recall是两个互相牵制、此消彼长的。通俗来讲，当模型倾向于追求分类为正例的样本实际为正例(也就是Precision高)时，模型分类为正例的要求会更加严格，导致并非所有实际为正例的样本都能被分为正例(即Recall低)；而当模型追求把所有实际为正例的样本全部分类为正例时，分类为正例的要求会降低，从而导致准确率降低。因此为了更好的衡量模型对黑标的分类情况，需要同时考虑Precision和Recall的值。

1. PR曲线:

横轴为Recall纵轴为Precision的曲线。统计同一个模型，不同超参数下的Precision和Recall，在PR曲线中，挑选兼顾Precision和Recall的那组超参数作为最优超参数。

1. F1：

F1是Precision和Recall的调和平均数即 。F1越大，表示模型分类效果越好。

1. ROC：

ROC是横轴为假阳性率，纵轴为准确率的曲线。其中假阳性率表示了所有负例中有多少被错认为正例。当随机分类时，ROC对应区间[0,1]上斜率为1的过原点的线段。模型的ROC越是在这条线段的上方表示性能越好。因此我们计算模型ROC的曲线下面积(AUC)即可对模型的性能作出定量分析。ROC的AUC越大，表示模型的分类能力越强。

## 第四节 本章小结

本章详细介绍了EvolveGCN的数据预处理方式、模型结构和训练过程以及常用的分类结果评价指标。接下来的一章里将介绍EvolveGCN实现的一些细节，利用可视化的方式展现模型的分类结果并同GCN、SGC进行对比和分析。

# 第四章 实验

## 第一节 实验条件

|  |  |  |
| --- | --- | --- |
| 数据来源 | 中国建设银行提供的经脱敏处理的账户交易流水以及账户信息数据。 | |
| 软件条件 | 操作系统 | Mac OS12.3.1 (21E258) |
| 开发工具 | Visual Studio Code |
| 编程语言和框架 | Python3.8 + Pytorch1.8.1 |
| 硬件条件 | GPU型号: Nvidia Tesla V100S (Memory 32GB) | |

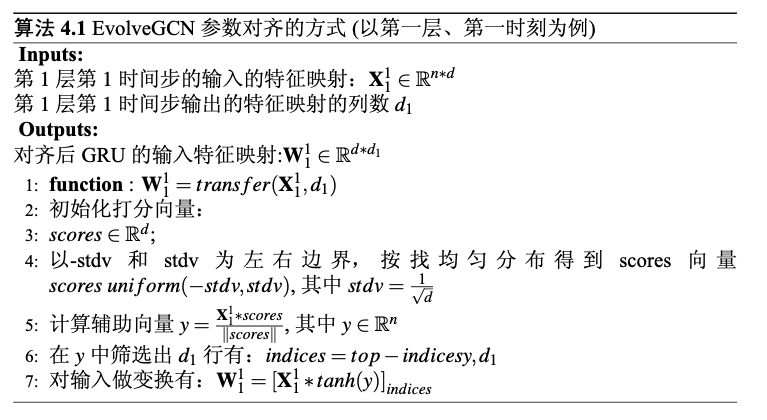
表4.1 实验条件表

## 实现细节

### 4.2.1 EvolveGCN中的参数规模对齐

在第三章第二节中，我们通过算法3.1和图3.1、图3.2对EvolveGCN的运行流程进行了介绍。在实现中我们还需要注意参数的规模问题，即GRU中输入矩阵的列数必须和隐藏单元矩阵的列数相同、GCN中输出矩阵的列数必须和网络参数的列数相同。以第一层为例，首先从GCN的角度来看：假设节点个数为n, 每个节点的特征向量维度为d，经过图卷积操作后第二层的节点特征向量维度是则有 **， 。**从GRU的角度看，和的规模需保持一致，因此有**。**为了输入能够和隐藏单元参与GRU运算，需要对在最大限度保留其信息的同时进行变换，使得形状和匹配**。**

参考Cangea等人的工作[[[8]](#endnote-7)]我们通过引入一个辅助打分向量scores来对输入进行线性变换。其伪代码如下算法4.1所示：



把参数对齐的过程带入到EvolveGCN单元中，可以得到修改后的EvolveGCN单元计算过程如下：

(4.1)

(4.2)

其中表示的列数， 表示对做算法4.1所述的变换后再求转置的结果，其形状同，因此可以顺利输入到GRU单元。需要注意的是，GCN方向上EvolveGCN单元不需要对输入进行变换。

### 4.2.2 GCN的简化版本SGC和实现

参考Wu Flix等人的工作[4]，我们得知GCN中真正起作用的是多层卷积操作，而层间起到连接作用的非线性激活函数对最终的结果影响并不大。因此，我们可以将GCN中多层的非线性激活函数省去，从而将公式(2.14)改为 。不妨令卷积核**K**，并将多层公式统一可以得到：

(4.3)

其中表示图卷积操作的层数，是单层卷积核的m次方、表示SGC整体的卷积核。此外由于省去了层与层之间的非线性激活函数，层与层之间的参数就可以统一成一个的参数。

由于卷积核只同图的结构有关，因此我们可以把的计算过程一并放入数据预处理中，在得到整体卷积核 后SGC的结构近似一个线性模型，非常容易实现。

## 实验结果

### 4.3.1 EvolveGCN的运行结果

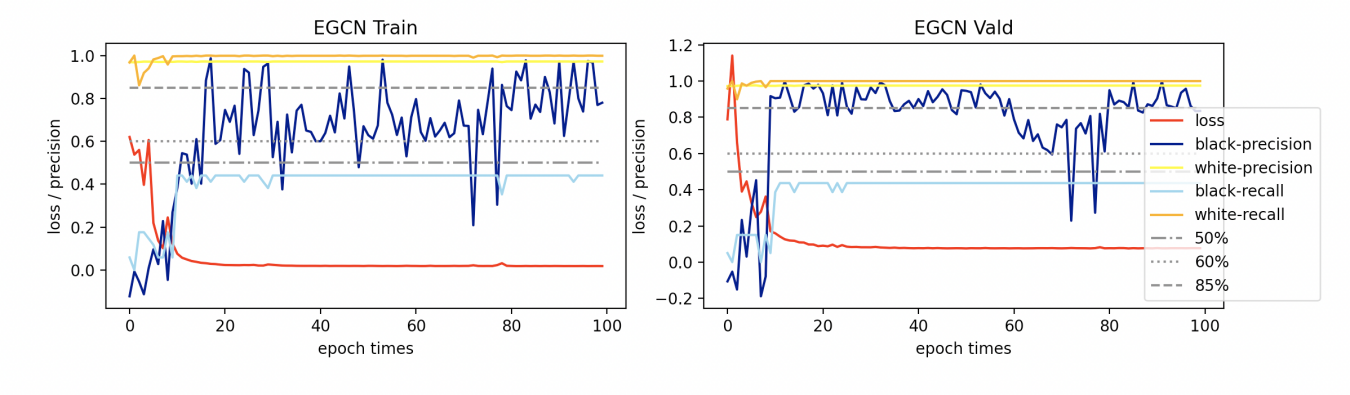


图 4.1 EvolveGCN在训练集和测试集中的表现；红色的线条表示损失值，深蓝色线条表示以黑标为正例时的准确率，浅蓝色线条表示黑标为正例时的召回率；黄色和橙色的线条则分别表示以白标账户为正例时的准确率和召回率。

首先从图4.1可以看出，黄色和橙色线条所表示的白标账户的准确率和召回率都十分接近1, 这很好地反映出由于黑标和白标账户的比例极其不均衡导致的白标账户准确率和召回率虚高现象。接下来我们将重点关注训练过程中损失的变化以及黑标账号的分类情况。

训练集和测试集的损失值成减速下降趋势，并逐渐稳定在0.05左右，表明模型的拟合效果良好。黑标账户的准确率在经过约15轮训练后在80%上下浮动，但会偶尔出现下降的情况(60-80轮中的两次下降)。这表明EvolveGCN模型判断为黑标的账户有极大可能参与了洗钱过程。同时不难发现，黑标账户的召回率稳定在45%附近，这表明虽然模型的黑标账户准确率较高，但很难把所有涉嫌参与洗钱的账户全部找出，将会忽略一些参与洗钱的账户。总体来说，该模型是有一定使用价值的。

### 4.3.2 GCN的运行结果

GCN模型的训练和测试结果见图4.2。同上，GCN的运行结果中白标账户的准确率和召回率都接近一，重点分析损失的变化以及黑标账户的分类情况。

## 

图 4.2 GCN在训练集和测试集上的表现

GCN的在训练集上的表现良好，损失值减速下降直到收敛于0.08附近，且在收敛后具有90%附近小幅摆动的黑标准确率和60%附近的黑标召回率。但是，它在测试集上的表现十分糟糕，甚至出现了损失降低后又上升的现象，表明GCN在处理这类数据时容易导致过拟合，模型泛化能力差。

### 4.3.3 SGC的运行结果

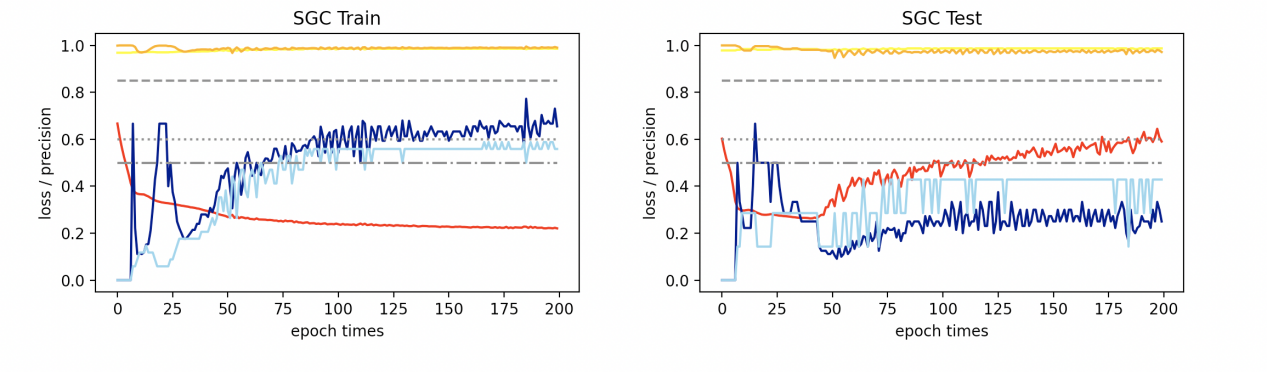


图 4.3 GCN在训练集和测试集上的表现

总体来看，SGC和GCN在训练集和测试集上表现基本一致：训练集上黑标准确率和召回率均可稳定在较高水平，然而在测试集上则出现loss先降后升、准确率和召回率都低于50%的情况。这表明SGC对GCN的优化仅仅体现在减少了不必要的计算，并没有针对动态图的特征进行改进，导致模型的拟合能力较GCN并无明显提升。

4.3.4 三个模型的ROC曲线对比

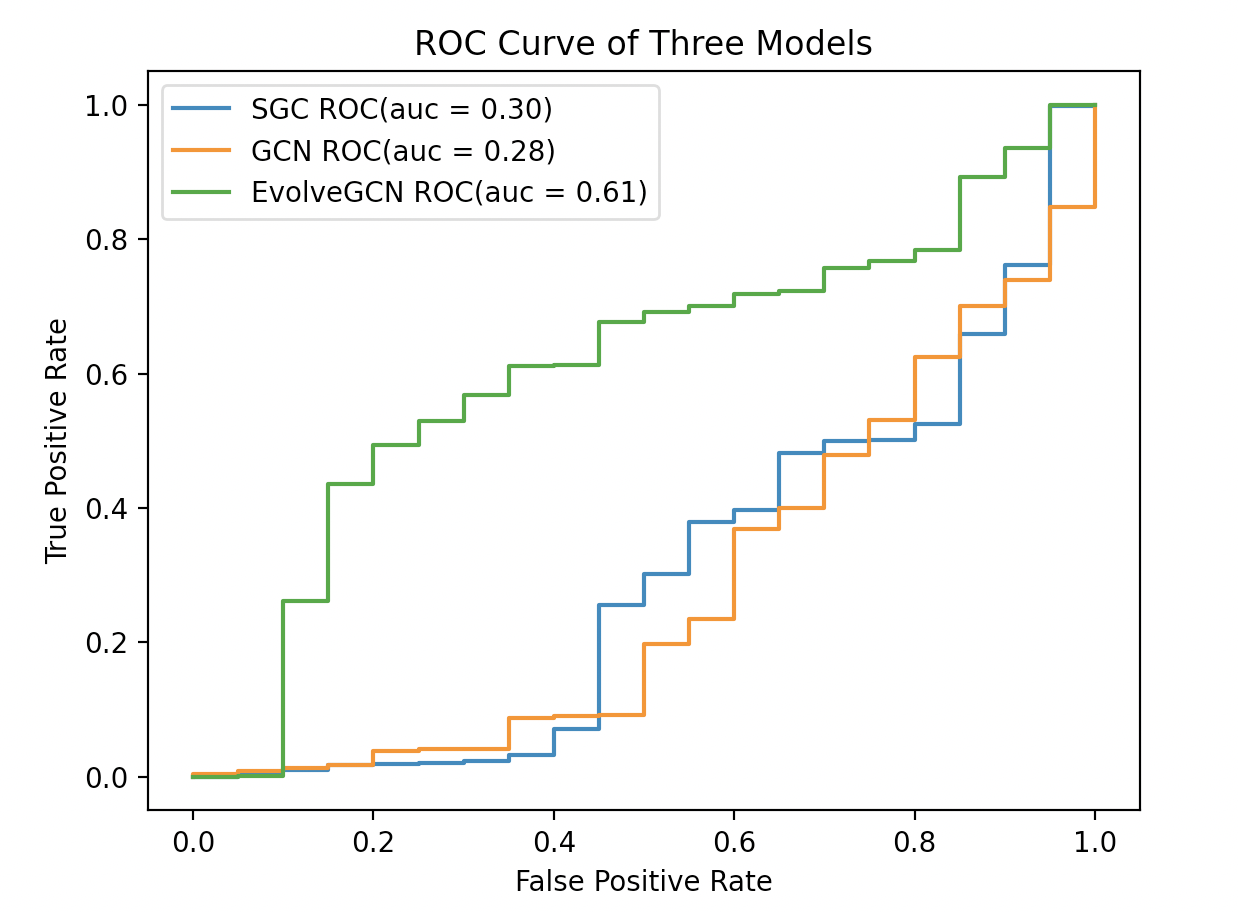


图 4.4 EvolveGCN、SGC和GCN的ROC曲线对比图

EvolveGCN模型由于考虑了时序性，并将其同图的结构信息融合在一起，从而使得模型的性能明显超过了静态的GCN和SGC模型。显示出EvolveGCN中利用循环神经网络驱动图神经网络这一关键思想的正确性，是值得继续深入研究的。

## 第四节 本章小结

本节首先讲述了EvovleGCN以及SGC的实现细节，为读者复现实验结果提供了保障。接着对EvolveGCN、GCN以及SGC三个模型的训练结果进行了展示，并依次重点分析了黑标准确率以及召回率，对模型进行了初步分析。最后我们对比了三个模型的ROC曲线，直观地展现了模型拟合能力的对比情况，证实了EvovleGCN中利用RNN驱动GCN训练的正确性、有效性。为继续深入研究动态图上的深度学习方法提供了一种潜在的可行范式。接下来的一章对作者初次独立科学研究的过程进行了总结和分析，得出了作者自己的科研方法论以及对EvolveGCN模型优缺点、应用方向以及发展前景的认知。

# 第五章 讨论和总结

## 第一节 本文总结

本文着眼于反洗钱任务，对国内外研究现状进行了对比分析，最终选择了适用于动态图的EvolveGCN模型来完成该任务。为介绍EvolveGCN模型的原理，本文首先梳理了基于谱方法的图卷积神经网络发展历程并介绍了几种常用的循环神经网络。此外，由于用于研究的数据标签量较少，我们介绍了在这样的图数据下进行半监督学习的一种方法，并将该方法应用于实验部分的所有模型的训练过程中。

在完成基础部分的介绍后，本文详细地介绍了EvovleGCN的组成单元以单元之间连接和扩展的方式，并展示了模型运行过程中的参数更新过程，使得读者能够准确、快速地理解EvolveGCN模型的原理和创新之处。在接下来的实验部分，我们依次分析了EvolveGCN、GCN以及SGC三者在黑标账户的准确率和召回率方面的表现。并比较了三者的ROC图像，并对三者的异同给出了初步解释。综合实验结果来看，模型具有一定的实际应用的价值，但是距离真正实现带给“洗钱者”以沉重震慑和打击的目标仍具有不小的差距。总体来讲，本文为作者第一次研究尝试，过程中取得了不小的进步，取得了一定的成果，可谓作者推开科研大门的尝试，为接下来研究生生涯的学习、科研和工作做了试探、预热、铺垫和准备。

## 第二节 进一步展望

在科学研究方法论方面，通过这次近似独立科研的尝试中我学习到了做科研的关键：

1. 研究任务以问题为导向还是以方法为导向只能择其一，不可贪图兼得。
2. 文献查阅、筛选、阅读以及理解能力对提升科研效率和质量十分重要！
3. 沟通和交流是必不可少的环节，做学术必须和同行保持定期沟通，方可互通有无，激发灵感。

在本文重点关注的EvolveGCN模型方面，随着研究的深入，对该模型的优缺点有了如下认知：

|  |  |
| --- | --- |
| 优点 | 缺点 |
|  |  |

因此我们可以认为在将使用深度学习完成反洗钱任务的重难点为：

1.

2.

3.

# 参考文献

1. [] Aldo Pareja, Giacomo Domeniconi, Jie Chen, et al. Evolvegcn: Evolving graph convolu-

   -tional networks for dynamic graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020: 5363 ∼ 5370. [↑](#endnote-ref-0)
2. [] Cho K, Gulcehre B M C, Bahdanau D, et al. Learning phrase representations using RNN encoder decoder for statistical machine translation. arXiv preprint arXiv:1406.1078, 2014. [↑](#endnote-ref-1)
3. [] Thomas N Kipf and Max Welling. Semisupervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016. [↑](#endnote-ref-2)
4. [] Felix Wu, Amauri Souza, Tianyi Zhang, et al. Simplifying graph convolutional networks. In: International conference on machine learning, 2019: 6861 ∼ 6871. [↑](#endnote-ref-3)
5. [] Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 2012, 25. [↑](#endnote-ref-4)
6. [] Jonathan Long, Evan Shelhamer and Trevor Darrell. Fully convolutional networks for

   semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015: 3431 ∼ 3440. [↑](#endnote-ref-5)
7. [] Michael Defferrard, Xavier Bresson and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems, 2016, 29. [↑](#endnote-ref-6)
8. [] Cangea, C.; Velickovic, P.; Jovanovic, N.; and Thomas Kipf, P. L. 2018. Towards sparse hierarchical graph classififiers. In *NIPS Workshop on Relational Representation Learning*. [↑](#endnote-ref-7)