# 关系图卷积模型

随着深度学习的发展，基于神经网络的机器学习方法成为人工智能研究中的基石，然而神经网络所具有的不可解释性仍然具有隐患。图神经网络模型作为一种新兴方法，具有更强的可解释性，它各个样本建模为具有联系的整体，不再孤立地处理数据，并且在多种任务上表现出显著优势。针对本文所研究的客户订单数据，我们相应构建关系图卷积模型对其进行分析。

## （一）基本原理

### 1. 图数据构建

本文所分析的问题中存在多个数据表，分别记录了客户信息、订单信息和商品信息，客户表和商品表中每个样本都有一个唯一编码，订单表通过将唯一编码对应的方式将客户和商品对应起来，记录了所有客户的购买行为。如果要研究客户的交易行为，传统的分析方法一般假设样本之间是相互独立的，通过特征工程方法，将订单表和商品表的信息手动提取并合并到客户表中，这样很大程度上损失了商品表和订单表的信息。

近年来，随着图神经网络的兴起，【引用一些文献】人们倾向于关注数据之间的联系，即不再认为样本之间完全独立，而是存在各种经济、社会或物理层面的联系。Sanchez-Lengeling, Benjamin 等[1]介绍了基于拉普拉斯矩阵图卷积神经网络模型（GCN）的基本原理。大量实验表明，图神经网络模型在有关系建模的数据中具有优势。对于本文研究，我们可以将客户和商品定义为不同类型的节点，利用订单关系构建边连接，该模型为异质图，参考图 1 。

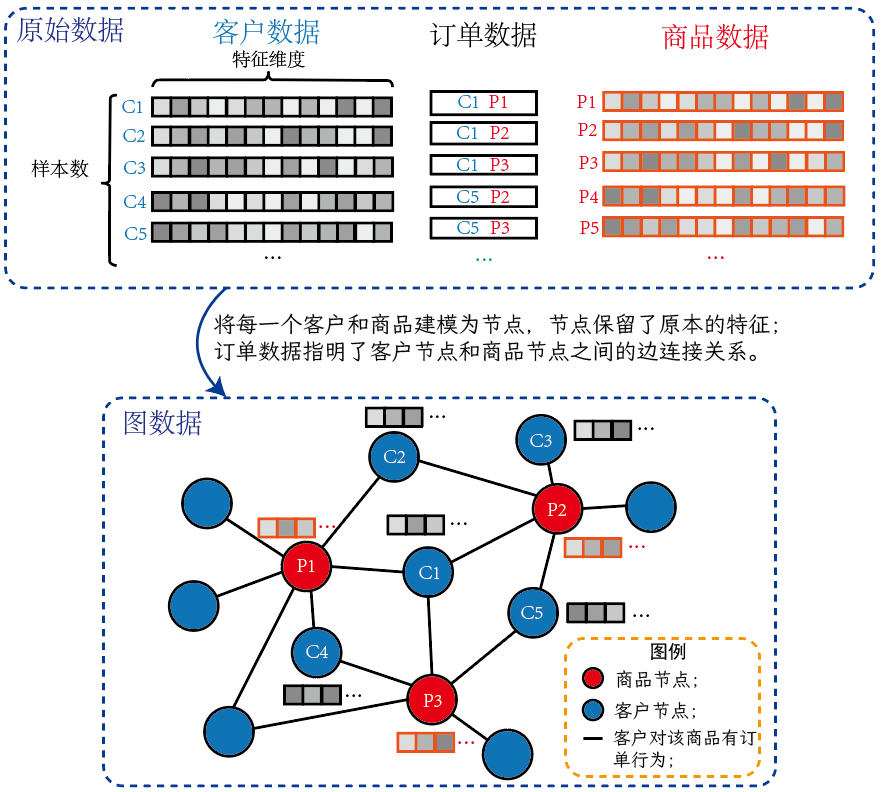


图 1 异质图构建方法

本模型的目的在于构建链接预测任务，这是一种无监督学习方法。模型将给出一个概率，衡量商品节点和客户节点之间存在边连接的可能性。对于商品供给方来说，可以依据该模型给出的概率制定生产和销售策略，对于需求方来说，他们可以获得更加具有针对性的个性化商品推荐。

### 2. 图卷积神经网络

图卷积神经网络（GCN）的卷积是指信息聚合，其原理核操作方法与卷积神经网络（CNN）中的卷积并不一致。对于图数据，可以得到其邻接矩阵、所有节点的特征矩阵和度矩阵。我们首先求其拉普拉斯矩阵：

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

拉普拉斯矩阵引入了自环和归一化操作，有利于优化信息聚合效果，聚合公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

其中是非线性函数，如sigmoid、ReLU等，是可训练权重矩阵，是信息聚合次数，本文模型中每次训练聚合两次。相对于传统的神经网络模型，图神经网络的主要贡献在于引入了拉普拉斯矩阵，将不同的样本关联起来，实现信息传递机制。

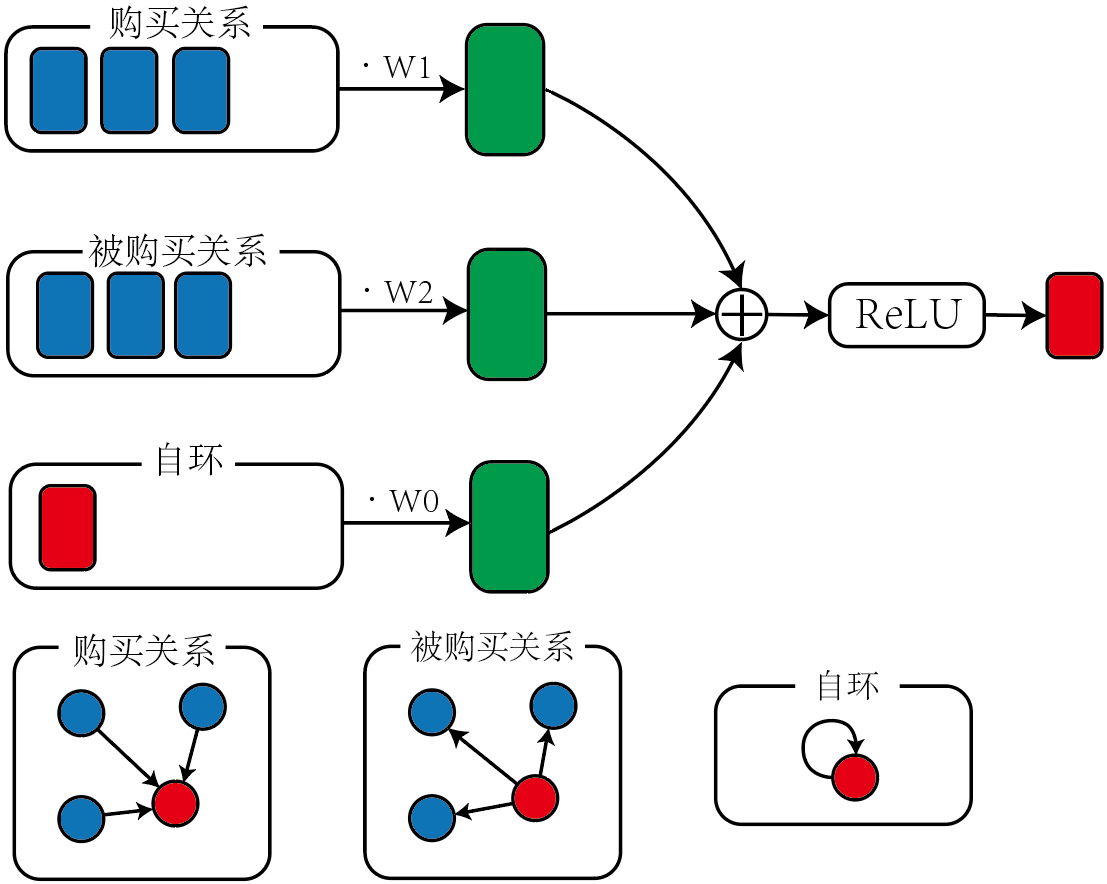


图 2 本文关系图卷积模型

### 3. 关系图卷积模型

传统的GCN模型只能处理同质图模型，其中所有节点都是相同类型，例如都是客户节点，并且每个节点的特征维度一致，每个节点之间的关系也一致，这意味着公式（2）中仅需要一个矩阵即可。在本文模型中，由于存在两种不同的节点类型，因此同一条边也具有两种相反的含义，即购买和被购买两种关系，它们统一于一条边的表示中。在模型层面，由于不同节点的特征维度不一样，我们需要对不同的关系给出不同的可训练权重，该方法由Michael Schlichtkrull等人[2]于2017年提出，称为关系图卷积模型，他们首次在异质关系型数据中应用图神经网络并且取得良好效果。本文模型参考图 2 。

其中是对应自环运算的可训练权重，是对应购买关系的可训练权重，是对应被购买关系的可训练权重。我们可以指定一个输出维度，使得各个关系通过线性变化后维度一致，对应元素加和并且通过ReLU激活输出。信息聚合公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (3) |

其中是可训练权重，表示所有关系类型的集合，是指在关系下节点的邻居集合，表示节点的特征矩阵，是节点的特征矩阵，最外层是非线性激活函数，上标中表示信息聚合的轮数，的取值可以根据任务具体情况而定，一般可以令其等于。

信息聚合可以看作是编码过程，目的是将相似节点的信息尽可能聚合到一起，突出该类别的特征便于聚类。然而，编码后无法立即给出边连接的预测概率，因此需要再构建一个解码器。Michael Schlichtkrull等在其研究中采用DistMult[3]作为解码方案，表现为将相互连接的两个节点的向量做点积，这样每两个点得到一个单值作为两者之间边存在与否的逻辑值，其单值没有意义，但不同数值之间的大小可以用于对比，逻辑值越大越有可能存在边连接。

本文在实验中发现DistMult方法效果在所研究的数据条件下并非最优，因此定义一种新的解码器方案，即将相连接的两个节点的特征向量相互拼接代表两个节点之间边的特征，将该特征矩阵经过一个线性层输出，最后经过sigmoid函数输出得到边存在与否的概率值，本文称其为全连接解码，其结构如图 3 所示：

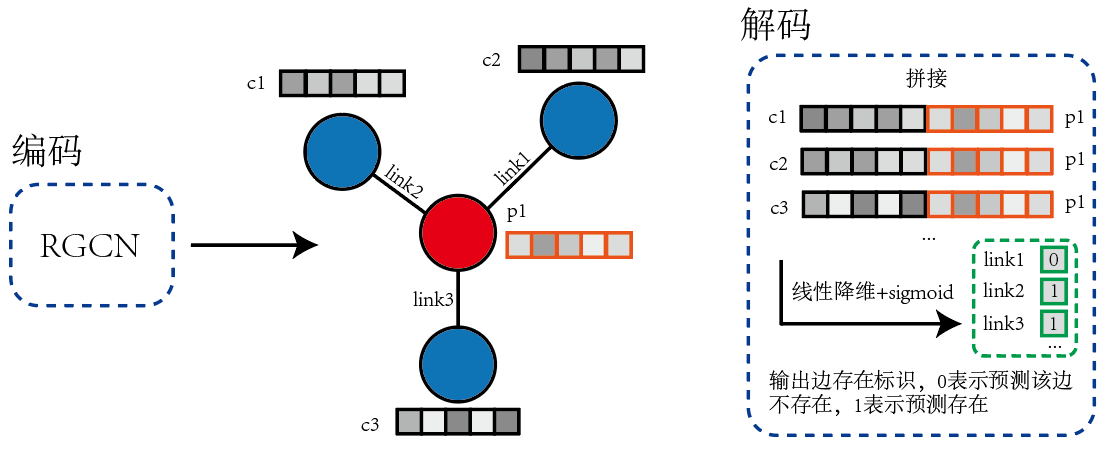


图 3 解码器框架

### 4. 数据分割

如果样本都从图中采集，那么所有的边都应预测为存在，这种情况下模型无需学习，只需对所有边输出1即可获得非常高的准确性。因此，在分割数据集时，将按照正例：负例为1：1的比例生成原本不存在的边连接，如果模型的输出全为1，则准确性只有0.5，从而激励模型学习图结构特征。

本文将原本数据集分割为训练集、验证集、测试集，比例为0.7:0.15:0.15。【在论文某处需要说明一下数据量】。

### 5. 损失函数

解码器输出边存在概率后，与真实标签对比即可计算出二元交叉熵（Binary Cross Entropy）损失并训练模型参数，公式如下。

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

其中是样本个数，是真是标签，是模型给出的预测标签。此外，DistMult解码器方法也可以配合采用边缘损失（Margin loss）函数训练模型。

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

其中是正例得分，与之对应，是负例得分。该损失函数鼓励模型输出较大的正例逻辑值和较小的负例逻辑值，在实验部分我们将给出相应实验结果。

### 6. 节点特征随机编码

在大部分实务逻辑中，由于相关法律约束，用户、商品等主体的详细信息往往是未知的，或不具备太多实际意义，因此在实务场景中，训练一个节点特征编码器是必要的，即只需要知道不同节点之间的连接关系，而不需要知道节点自身的任何信息。在这种情况下，模型将学习纯粹的图结构特征。

本文同样给出这种模型，给每个节点分配一个可训练的随机特征编码，为了方便处理，所有节点的特征维度都是一样的。我们将在实验部分展示了这种方案的训练效果作为对比。

## （二）实验结果

本文构建了三种具有不同细节实现的模型。主要区别在于解码器、损失函数、节点是否随即编码三个方面。在模型一中，解码器采用全连接方案，损失函数采用二元交叉熵，节点使用原本特征。在模型二中，解码器采用全连接方案，损失函数采用二元交叉熵，节点特征采用可训练的随机编码。在模型三中，解码器采用DistMult方案，损失函数采用边缘损失，节点使用可训练随机编码。具体差异参考表格 1。

每种模型都采用了多个种子训练，用相同阴影标记对应曲线在不同种子情况下同一训练回合的最大最小值区间。

前两种模型衡量指标中有AUC值和平均精度。AUC用于衡量二分类模型性能指标，其值等于ROC曲线（接收者操作特征曲线）下的面积，ROC曲线以真阳性率（True Positive Rate）为纵轴，假阳性率（False Positive Rate）为横轴绘制实验结果曲线。AUC的取值范围在0到1之间，值越接近1表示模型性能越好，值越接近0.5表示模型性能越接近随机预测。平均精度计算公式为，其中TP是真阳性率，FP是假阳性率。精确率关注的是模型在预测正例时的准确性，而AUC则是综合考虑了模型在不同阈值下的整体性能。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 模型一 | 模型二 | 模型三 |
| 解码器 | 全连接 | 全连接 | DistMult |
| 损失函数 | 二元交叉熵 | 二元交叉熵 | 边缘损失 |
| 节点特征 | 原本特征 | 可训练随机编码 | 可训练随机编码 |

表格 1 各模型训练方案

### 模型一

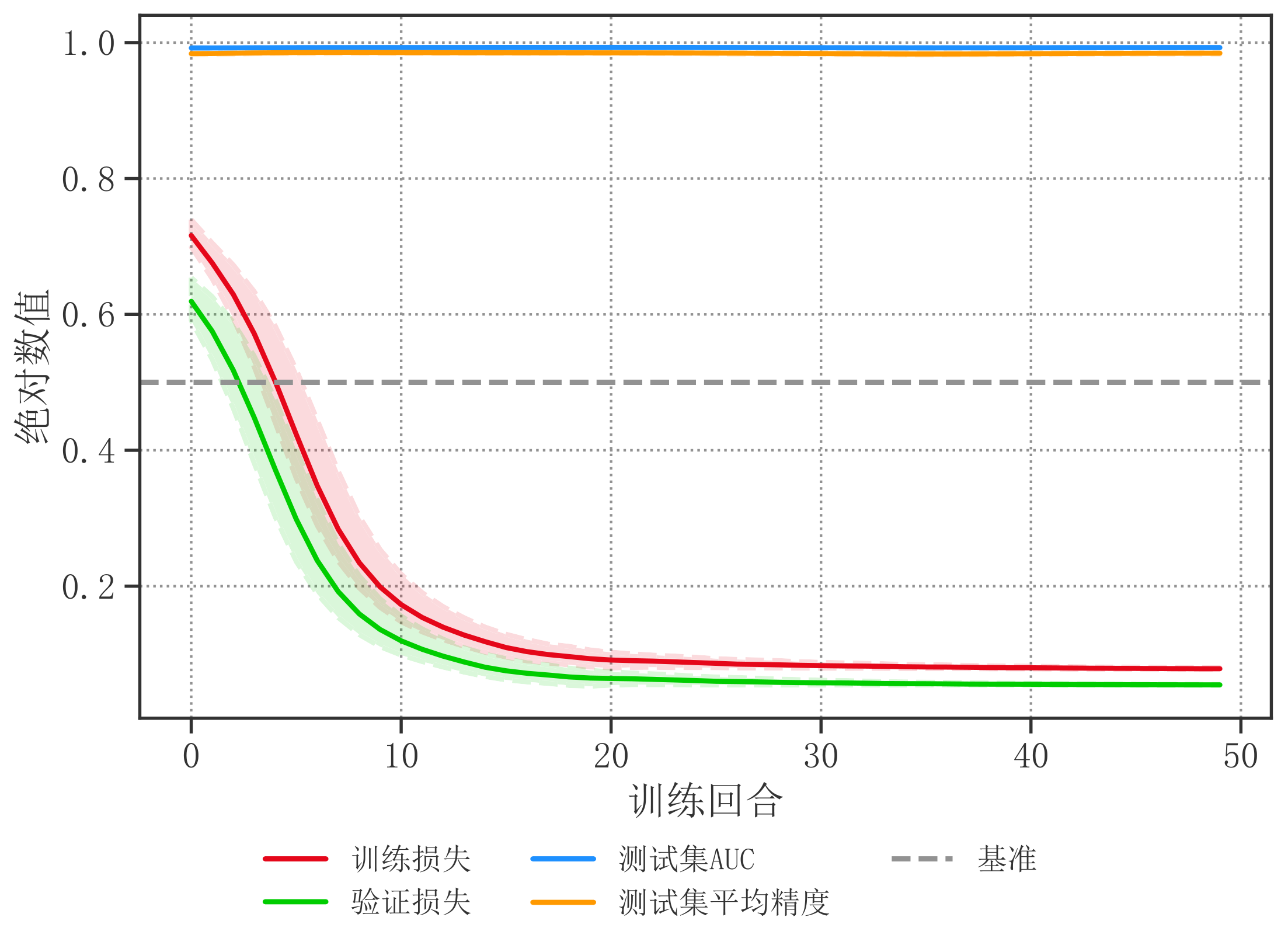


图 4 模型一训练结果

模型一的随机种子取值范围是43到52。训练损失和验证损失平稳收敛，测试集AUC和平均精度较高。

### 模型二

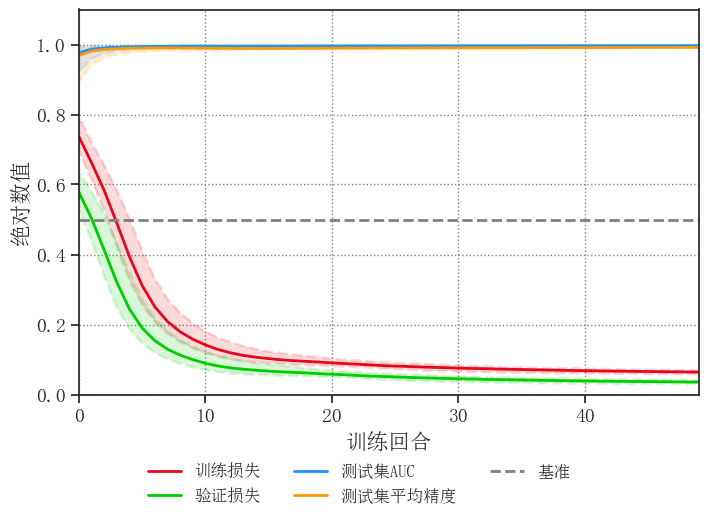


图 5 模型二训练结果

模型二的随机种子取值范围是43到52。收敛水平与模型一基本一致。由于在实际场景中节点特征位置的情况更常见，因此节点随机编码方法仍然不失为一种可行方案。

### 模型三

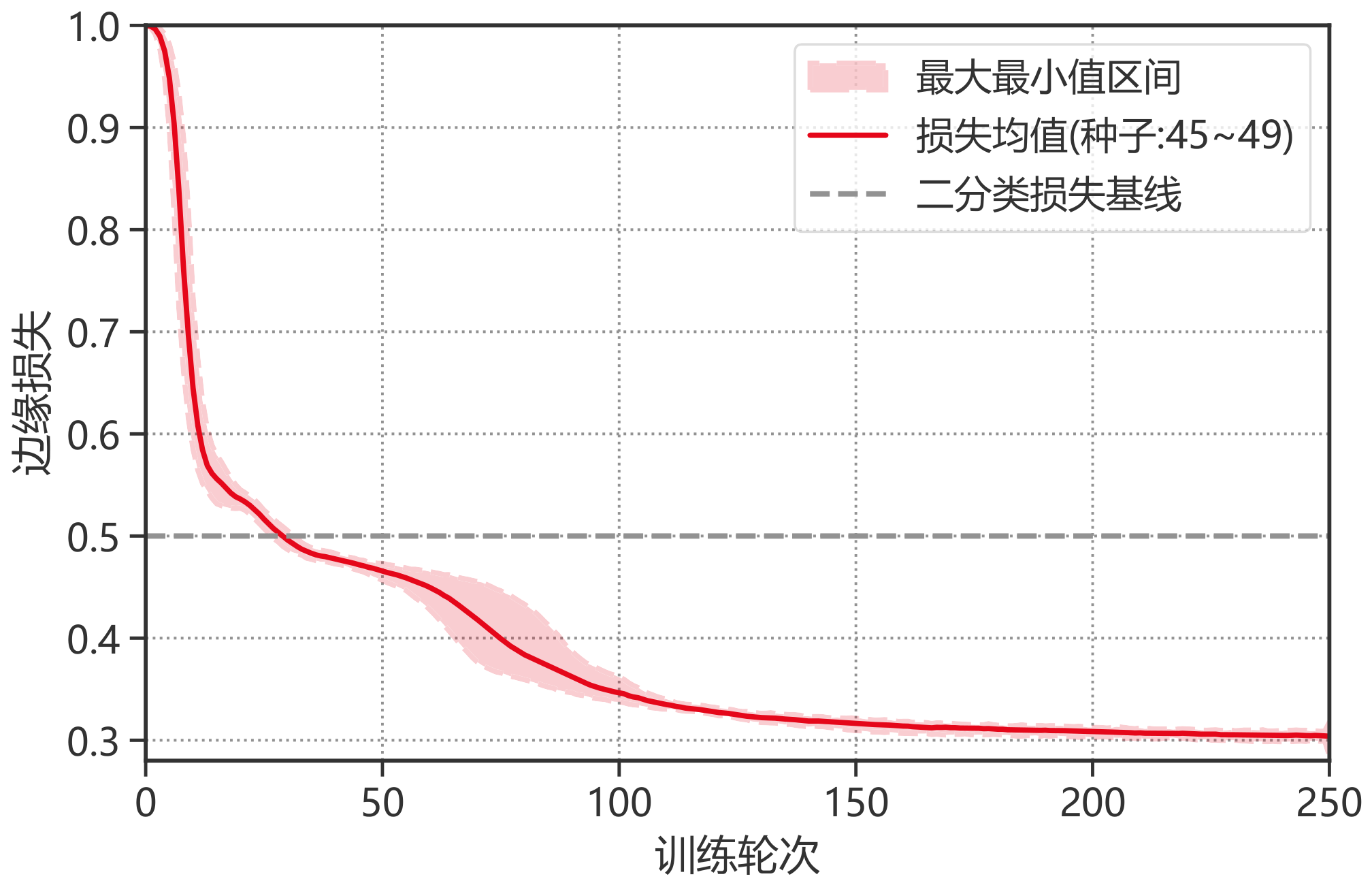


图 6 模型三训练结果

模型三的随机种子取值范围是45到49，观察到模型同样可以平稳收敛，但收敛位置高于前两种模型。由于该模型实现采用与前两种模型不同的算法框架，且训练效果不及前两种模型，因此没有进一步计算其他指标。

## （三）模型意义

【我就不写了。。】

# 参考文献

[1] SanchezLengeling, Reif B A, Pearce E A, et. al. A Gentle Introduction to Graph Neural Networks[J]. Distill, 2021.

[2] Schlichtkrull M, Kipf T N, Bloem P, et. al. Modeling Relational Data with Graph Convolutional Networks[J]. arXiv, 2017.

[3] Yang B, Yih W, He X, et al. Embedding Entities and Relations for Learning and Inference in Knowledge[C]. //Yoshua Bengio, Yann LeCun.3rd International Conference on Learning Representations (ICLR), 2015.