2024-2025 学年
知识工程（双语）实验报告

任课教师：吴天星

院　系 _____人工智能学院_____

专　业 _____人工智能_____

姓　名 _____陆文韬_____

任　务 _____知识嵌入_____

1 实验三

1.1 实验任务

1. 阅读 OpenKE Readme 文档及其样例程序，了解数据集结构及模型参数意义，训练满足以下条件的 TransE、TransH 模型，并获取测试性能，要求：
   - Embedding size = 100；
   - 数据集为 WN18RR
     其他参数可自行调整
2. 将代码运行过程与实验结果（链接预测）截图形成实验报告。

1.2 基本概念

1.2.1 OpenKE

OpenKE（Open Knowledge Embedding）是一个开源的知识图谱嵌入（Knowledge Graph Embedding）框架，旨在为研究者和开发者提供一个方便、高效的工具，用于在知识图谱中学习实体和关系的低维嵌入表示。知识图谱嵌入的目标是将图中的节点（实体）和边（关系）映射到一个低维向量空间，使得在这个向量空间中，实体和关系的相似性可以被量化，从而提高图谱分析和推理的能力。

主要特点：

- **支持多种模型**：OpenKE 实现了多种经典的知识图谱嵌入方法，包括 TransE、TransH、TransR、DistMult、ComplEx 等。每种方法具有不同的目标函数和优化策略，适用于不同类型的知识图谱任务。
- **易于使用**：OpenKE 提供了简洁的 API 和命令行工具，使得使用者能够快速进行模型训练、评估和测试。
- **高度模块化**：框架将嵌入学习、模型训练、评估和推理等过程分解为多个模块，方便扩展和定制化。
- **大规模数据支持**：OpenKE 能够处理大规模的知识图谱数据，支持高效的计算和分布式训练。
- **开放源码**：OpenKE 是一个开源项目，用户可以自由使用、修改和扩展其功能，社区也会不断更新和优化框架。

主要应用：

- **知识图谱补全**：通过学习实体和关系的嵌入表示，可以用于推测知识图谱中的缺失事实，例如，给定一对实体，预测它们之间可能的关系。

- **实体和关系分类**：通过向量空间中的距离度量，可以进行实体或关系的分类任务，例如，分类相似类型的实体或关系。
- **推理和问答**：知识图谱嵌入可以用于图谱推理，从已知事实中推导出新的事实，或者用于智能问答系统中，从知识图谱中提取相关信息。

安装和使用：

OpenKE 通常依赖于 Python 和 TensorFlow 等深度学习库，安装时可以通过 GitHub 下载源代码或使用 pip 安装。它的使用通常包括以下步骤：

1. **数据预处理**：将知识图谱数据（如三元组形式的实体-关系-实体）转换为适合模型训练的格式。
2. **模型选择**：选择适当的嵌入方法（如 TransE、DistMult 等）。
3. **模型训练**：使用提供的工具进行模型训练。
4. **评估与推理**：使用训练好的模型进行推理，生成新的知识或进行评估。

1.2.2 TransE

TransE（Translation-based Embedding）是一种经典的知识图谱嵌入方法，用于学习知识图谱中实体和关系的低维表示。该方法的基本思想是通过将三元组（实体 1，关系，实体 2）映射到一个低维向量空间，使得实体和关系在该空间中可以通过向量加法和向量相似性来进行表示。

模型原理：

TransE 的核心思想是将关系建模为从一个实体到另一个实体的“平移”操作。具体来说，给定三元组 (h, r, t) 表示实体 h、关系 r 和实体 t，TransE 模型试图学习向量表示 $\mathbf{h}$、$\mathbf{r}$ 和 $\mathbf{t}$，使得以下关系成立：

$$\mathbf{h} + \mathbf{r} \approx \mathbf{t}$$

即，实体 h 与关系 r 的向量和应该接近实体 t 的向量。在训练过程中，模型通过最小化目标函数来优化这些向量表示，使得符合关系的三元组的向量加法结果最小化，而不符合关系的三元组的加法结果最大化。

损失函数：

TransE 使用以下基于距离的损失函数来进行模型训练：

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [\gamma + d(\mathbf{h} + \mathbf{r}, \mathbf{t}) - d(\mathbf{h}' + \mathbf{r}, \mathbf{t}')]_+$$

其中，$d(\mathbf{a}, \mathbf{b})$ 是向量 $\mathbf{a}$ 和 $\mathbf{b}$ 之间的距离（通常使用欧氏距离），S 是正样本三元组集合，S' 是负样本三元组集合，γ 是一个超参数，$[x]_+$ 表示取最大值函数（即 $\max(0, x)$）。

主要特点：

- **简单高效**：TransE 通过加法操作来建模关系，使得计算和优化相对简单且高效，适用于大规模知识图谱。
- **通用性**：TransE 可以处理各种类型的知识图谱数据，包括稀疏关系和复杂的三元组。
- **扩展性**：尽管 TransE 简单有效，但它也有一些局限性，特别是在处理复杂的关系（如一对多或多对多的关系）时。为了克服这些局限性，后续提出了许多基于 TransE 的变种（如 TransH、TransR 等）。

应用场景：

TransE 广泛应用于以下几个领域：

- **知识图谱补全**：TransE 可以用于推断知识图谱中缺失的事实，预测三元组中的缺失部分（如关系或实体）。
- **实体链接与关系预测**：通过计算实体和关系的向量表示，TransE 能够有效地进行实体链接和关系预测任务。
- **图谱推理**：TransE 嵌入的向量空间可以用于推理，从已知的事实中推导出新的信息。

1.2.3 TransH

TransH（Translation-based Embedding with Hyperplane）是对 TransE 模型的改进，旨在处理 TransE 在面对一些复杂关系时的不足，尤其是在一对多、多对一和多对多的关系上。TransH 通过引入超平面来建模关系，使得实体的嵌入向量能够更灵活地适应不同类型的关系。

模型原理：

TransH 的核心思想是在每个关系上引入一个超平面，并将实体嵌入到该平面上。这意味着对于每个关系 r，TransH 为每个实体学习一个超平面，并将实体嵌入向量投影到该超平面上。给定三元组 (h, r, t) 表示实体 h、关系 r 和实体 t，TransH 模型试图学习向量表示 $\mathbf{h}$、$\mathbf{r}$ 和 $\mathbf{t}$，使得以下关系成立：

$$\mathbf{h} + \mathbf{r} \approx \mathbf{t}$$

但与 TransE 不同，TransH 在实体嵌入时会先通过一个超平面投影操作。具体而言，TransH 首先将实体 h 和 h 的超平面（对应关系 r）进行投影，然后进行加法操作：

$$\mathbf{h}_r = \mathbf{h} - \frac{\mathbf{h} \cdot \mathbf{r}}{\|\mathbf{r}\|^2}\mathbf{r}$$

$$\mathbf{t}_r = \mathbf{t} - \frac{\mathbf{t} \cdot \mathbf{r}}{\|\mathbf{r}\|^2}\mathbf{r}$$

这两个投影后的实体向量 $\mathbf{h}_r$ 和 $\mathbf{t}_r$ 将被加到一起，形成最终的实体关系嵌入。

损失函数：

TransH 使用与 TransE 类似的基于距离的损失函数来进行模型训练：

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [\gamma + d(\mathbf{h}_r + \mathbf{r}, \mathbf{t}_r) - d(\mathbf{h'}_r + \mathbf{r}, \mathbf{t'}_r)]_+$$

其中，$d(\mathbf{a}, \mathbf{b})$ 是实体向量之间的距离（通常使用欧氏距离），S 是正样本三元组集合，S' 是负样本三元组集合，γ 是一个超参数，$[x]_+$ 表示取最大值函数（即 $\max(0, x)$）。

主要特点：

- **更好地处理复杂关系**：通过引入超平面，TransH 能够更好地处理一对多、多对一和多对多的关系，因此比 TransE 更灵活。
- **保持 TransE 的优点**：与 TransE 相比，TransH 在计算效率上几乎没有增加复杂度，因此仍然具有较高的计算效率。
- **灵活性和可扩展性**：TransH 的模型结构具有很强的灵活性，能够处理多种复杂关系，并且可以扩展到其他复杂模型中。

应用场景：

TransH 在知识图谱嵌入任务中有广泛应用，特别是在需要处理多种关系类型的情况下。主要应用场景包括：

- **知识图谱补全**：TransH 可以有效地预测知识图谱中的缺失三元组，尤其是在复杂关系下。
- **实体链接和关系预测**：TransH 能够对多类型的关系进行建模，从而更好地进行实体链接和关系预测。
- **复杂图谱推理**：由于 TransH 能够处理更复杂的关系，因此它在图谱推理任务中表现得更加高效。

1.2.4 WN18RR 简介

WN18RR 是一个改进版的知识图谱数据集，基于 WordNet 18（WN18）构建，用于知识图谱嵌入任务中的关系预测和图谱补全。与原始的 WN18 数据集相比，WN18RR 进行了去除冗余和错误数据的处理，旨在为知识图谱嵌入模型提供一个更加标准化和挑战性的测试集。

数据集背景：

WN18RR 是一个从 WordNet 词汇数据库中提取的子集，包含了词汇之间的实体和关系。WordNet 是一个大型的英语词汇数据库，其中词汇被组织成同义词集（synsets），并通过各种语义关系（如“是一个（is-a）”、“属于（part-of）”等）进行连接。WN18RR 由以下组成：

- **实体：**包含了多个实体，如名词、动词等，代表不同的词汇。
- **关系：**包括语义关系，如同义词、上位词、下位词、部分关系等。
- **三元组：**数据集中包含了多达 18,000 个三元组，每个三元组由实体 1、关系、实体 2 组成。

WN18RR 的改进：

WN18RR 在 WN18 的基础上做了以下几个重要的改进：

- **去除冗余：**删除了 WN18 数据集中冗余的三元组，以避免模型过拟合。
- **消除噪声：**通过清洗数据，去除了错误的三元组，确保数据集的质量。
- **更加标准化：** WN18RR 遵循了更为严格的数据预处理规范，确保每个关系和实体的表示更加一致。

主要特点：

- **多样化的关系类型：**数据集中包含了多种类型的关系，涉及不同的语义层次和结构，适用于复杂的关系建模任务。
- **挑战性：**由于去除了冗余和噪声，WN18RR 成为了一个更具挑战性的知识图谱补全任务，适合评估知识图谱嵌入算法的性能。
- **广泛应用：**WN18RR 被广泛用于评估各种知识图谱嵌入方法，如 TransE、DistMult、ComplEx 等，是关系预测任务中的标准数据集之一。

应用场景：

WN18RR 主要用于以下几个应用场景：

- **知识图谱补全：**通过训练嵌入模型，WN18RR 被广泛用于推断知识图谱中的缺失三元组，特别是关系预测任务。
- **关系预测：**数据集中的不同关系类型使得 WN18RR 非常适合用来训练和评估关系预测任务。
- **图谱推理：**通过推理已知关系，WN18RR 帮助评估图谱推理模型的能力。

1.3 实验平台

- 系统版本：Ubuntu 22.04 LTS
- GPU：NVIDIA GeForce RTX 4060Ti 16GB
- CUDA：11.2

1.4 实验指标和测试条件

1.4.1 实验指标

- MRR（Mean Reciprocal Rank）：平均倒数排名。对于每个预测答案，计算它的排名的倒数（1/排名），然后对所有预测取平均值。MRR 值越高表示模型对正例的排序质量越高
- MR（Mean Rank）：平均排名。表示预测中正例的平均排名，数值越小表示模型对正例的预测越靠前
- Hit@10, Hit@3, Hit@1：命中率指标，表示预测的前 10、前 3 或前 1 个答案中包含正例的比例。数值越高表示模型在前 N 个预测中包含正例的概率越大

1.4.2 测试条件

- r(raw) 和 l(raw)：分别表示右侧和左侧原始数据的测试结果
- r(filter) 和 l(filter)：分别表示右侧和左侧经过过滤的数据的测试结果。过滤后的数据通常会去除一些可能干扰模型预测的噪声样本
- averaged(raw) 和 averaged(filter)：分别表示原始和过滤数据下的平均表现

1.4.3 具体实验要求

- 数据集采用 WN18RR
- Embedding size 改为 100
- 采用 averaged(filter) 的 Hit@10 指标来衡量训练结果的优劣
- 考虑到实际的训练时长问题，我们将 training times 从 1000 改为 100，从而节约了大量的训练时长，从而有更多的时间去尝试寻找最佳的参数组合

1.5 结果分析

针对 TransE 和 TransH 两个模型分别对模型中的 margin 和 learning rate 两个参数进行网格搜索，得到三维的散点图。通过分析散点图的变化趋势得出模型的最佳参数。

1.5.1 TransE

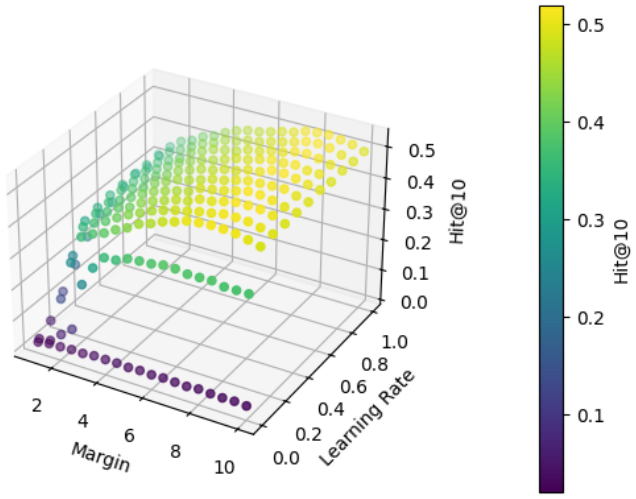


图 1: TransE

metric:	MRR	MR	hit@10	hit@3	hit@1
l(raw):	0.152628	2708.246338	0.439056	0.231653	0.014678
r(raw):	0.212987	2137.862549	0.529675	0.298660	0.060306
averaged(raw):	0.182807	2423.054443	0.484365	0.265156	0.037492
l(filter):	0.214404	2686.296143	0.490108	0.366305	0.032546
r(filter):	0.255958	2132.784912	0.552967	0.403318	0.068922
averaged(filter):	0.235181	2409.540527	0.521538	0.384812	0.050734

0.521538
0.5215379595756531

图 2: TransE 最佳参数链接预测结果

从中可以看到，当 margin 在 8 到 8.5 的时候，模型的得分更高，而 learning rate 的推荐值则为大于 0.2。最佳的参数：margin 为 8.5，learning rate 为 0.9。

1.5.2 TransH

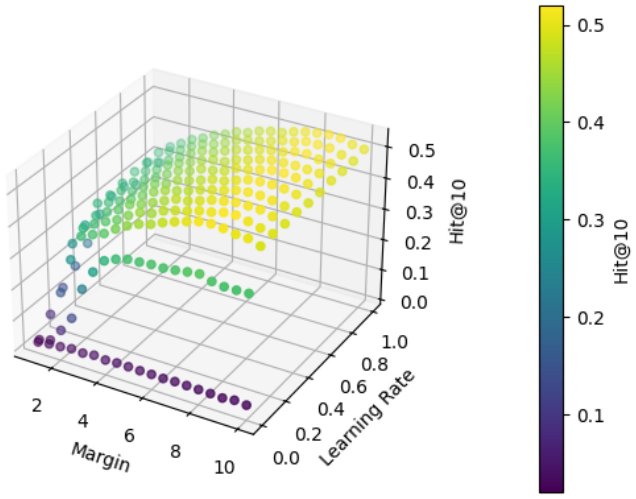


图 3: TransH

metric:	MRR	MR	hit@10	hit@3	hit@1
l(raw):	0.148681	2903.955078	0.436184	0.232291	0.008934
r(raw):	0.216224	2211.667236	0.531589	0.305999	0.063816
averaged(raw):	0.182453	2557.811035	0.483886	0.269145	0.036375
l(filter):	0.208516	2881.761230	0.486599	0.359604	0.026803
r(filter):	0.258744	2206.512207	0.552648	0.407786	0.072750
averaged(filter):	0.233630	2544.136719	0.519624	0.383695	0.049777

0.519624
0.5196235179901123

图 4: TransH 最佳参数链接预测结果

从图中可以看到，TransH 的模型对参数的偏好性质与 TransE 大致上相同，这可能是由于 TransH 本身是 TransE 的变种，因此两者有许多相似的性质和结构，从而导致对参数的偏好也很接近。最佳参数：margin 为 8.5，learning rate 为 0.6。

2 实验四

2.1 实验任务

1. 阅读 unKR 样例程序，了解数据集结构及模型参数意义，分别在 ppi5k 和 nl27k 两个数据集上训练满足以下条件的 UKGE 模型，并获取测试性能，要求：

   - 嵌入向量的维度为 256

- 最大 epoch 为 10

2. 其他参数可自行调整（注：所有的参数都可以通过对应的 yaml 文件设置）
3. 将 yaml 文件里所进行的参数设置操作, 以及最终的实验结果截图, 形成实验报告。操作步骤需要体现出使用 yaml 文件具体对哪些参数进行了设置，最终的实验结果需要包含 MAE，MSE，MRR，MR，WMRR，WMR 以及 Hits@N 的各项指标。

2.2 基本概念

2.2.1 UKG

UKG（Uncertain Knowledge Graph）是一种处理不确定性信息的知识图谱，旨在通过引入概率、模糊逻辑等方法，处理和表达现实世界中的不确定性。与传统知识图谱不同，UKG 不仅仅关心实体之间的确定性关系，还考虑到信息的不完全性、模糊性和潜在的变化性。UKG 在许多领域具有重要应用，如自然语言处理、语义搜索、推荐系统等，特别是在那些信息不完备或者存在模糊性的场景中，能够提供更加灵活和准确的推理能力。

2.2.2 unKR

定义

unKR(Uncertain Knowledge Gragh Reasoning) 是东南大学认知智能研究所 (seucoin) 基于 PyTorch_Lightning 开发的用于将不确定性知识图谱表示学习框架。unKR 是第一个用于不确定知识图谱表示学习的开源库，它是高度模块化和可扩展的，便于复现和定制不确定性知识图谱表示学习模型，unKR 在统一的工作流程下成功重新实现了近年来发布的九个不确定性知识图谱表示学习模型，包括 UKGE，PASSLEAF 等。

框架结构

- unKR 的核心代码都在 src 文件中，主要包含 Data Model LitModel Evaluator 四个模块
- Data 数据处理模块对输入的 UKG 数据集进行预处理，以便进行数据加载，主要包含 UKGData、UKGSampler 和 UKGDataModule
- Model 模型模块实现了两种类型的 UKG 表示学习模型，即 normal 和 few-shot 模型，每个模型都被封装到一个类中，例如 PASSLEAF 模型的 PASSLEAF 类
- LitModel 模块用于指导模型的迭代训练和评估。每个模型对应于一个继承自 BaseLitModel 类的派生类，其中 training_step 函数用于根据 Model Hub 模块中定义的评分函数计算批量数据的损失；validation_step 函数和 test_step 函数用于评估模型在验证集和测试集上的效果
- Evaluator unKR 在 eval_task 中实现了置信度预测和链接预测这两个不确定性知识图谱补全的子任务

文件组织结构

- 数据集在 dataset 文件夹下，具体包括 cn15k, nl27k, ppi5k 三个数据集
- 示例脚本在 demo 文件夹下，提供了每个模型的训练流程编写示例
- 参数文件使用 yaml，在 config 文件夹下，按数据集组织不同模型的对应参数设置

2.3 实验平台

- 系统版本：Ubuntu 22.04 LTS
- GPU：NVIDIA GeForce RTX 4060Ti 16GB
- CUDA：11.2

2.4 实验指标

- **MAE (Mean Absolute Error)**：MAE 是回归模型中常用的评估指标，表示预测值与实际值之差的绝对值的平均值。它反映了预测误差的平均水平，越小表示模型预测的准确度越高。
- **MSE (Mean Squared Error)**：MSE 是另一种常用的回归评估指标，计算的是预测值与实际值差的平方的平均值。由于平方的关系，MSE 对大误差较为敏感，能够有效惩罚较大的预测错误。
- **MRR (Mean Reciprocal Rank)**：MRR 常用于评估信息检索和推荐系统中，衡量模型对相关项的排序能力。它是查询的倒数排名的平均值。MRR 值越高，说明模型能够更好地将相关项排在前面。
- **MR (Mean Rank)**：MR 是评价模型排名能力的指标，计算模型对所有查询返回的所有项的平均排名。较低的 MR 值表示模型能够在较前的位置返回相关项。
- **WMRR (Weighted Mean Reciprocal Rank)**：WMRR 是对 MRR 的加权扩展，常用于推荐系统中，给不同的查询项或返回项赋予不同的权重，综合评估模型的排序效果。
- **WMR (Weighted Mean Rank)**：WMR 是对 MR 的加权扩展，它考虑了不同查询项的重要性，并根据给定的权重计算平均排名。该指标常用于需要对不同查询赋予不同优先级的场景。
- **Hits@N**：Hits@N 评估模型在前 N 个推荐项中是否包含正确答案。若在前 N 个预测项中至少有一个正确答案，则认为模型“命中”。该指标用于衡量模型在给定 N 个位置内的准确性。

2.5 可以优化的模型参数

1. **学习率 (lr)**:
   - 当前设置：$1.0e-03$
   - 改进建议：尝试试验不同的学习率，如 0.01, 0.0001 等，找到合适的值。

2. **批大小 (train_bs, eval_bs)**:
   - 当前设置：$train\_bs = 256$, $eval\_bs = 16$
   - 改进建议：调整批量大小，通常较大的批量大小有助于稳定训练,
3. **正则化 (weight_decay, regularization)**:
   - 当前设置：$weight\_decay = 0.01$
   - 改进建议：可以尝试增大或减小 $weight\_decay$，例如设置为 0.1 以增加正则化，从而防止过拟合，尤其是在数据集较小或模型较大时。
4. **优化器 (optim_name)**:
   - 当前设置：$optim\_name =$ `Adam`
   - 改进建议：Adam 优化器通常适用于很多任务,但也可以尝试 `AdamW`(带权重衰减的 Adam) 来提高收敛性。也可以尝试其他优化器。
5. **丢弃率 (dropout, hid_drop, inp_drop)**:
   - 当前设置：$dropout = 0$, $hid\_drop = 0.3$, $inp\_drop = 0.2$
   - 改进建议：适当增加丢弃率（尤其是 $dropout$）有助于防止过拟合，通常在 0.3 到 0.5 之间尝试效果较好。可以根据需要进行微调。
6. **数据增强 (filter_flag, inject_triple_percent)**:
   - 当前设置：$filter\_flag =$ `true`, $inject\_triple\_percent = 1.0$
   - 改进建议：增强数据可以帮助提高模型泛化能力。你以调整 $inject\_triple\_percent$（如从 0.5 到 1.5 之间），或者在数据预处理时引入更多的随机变换和数据增强方法。

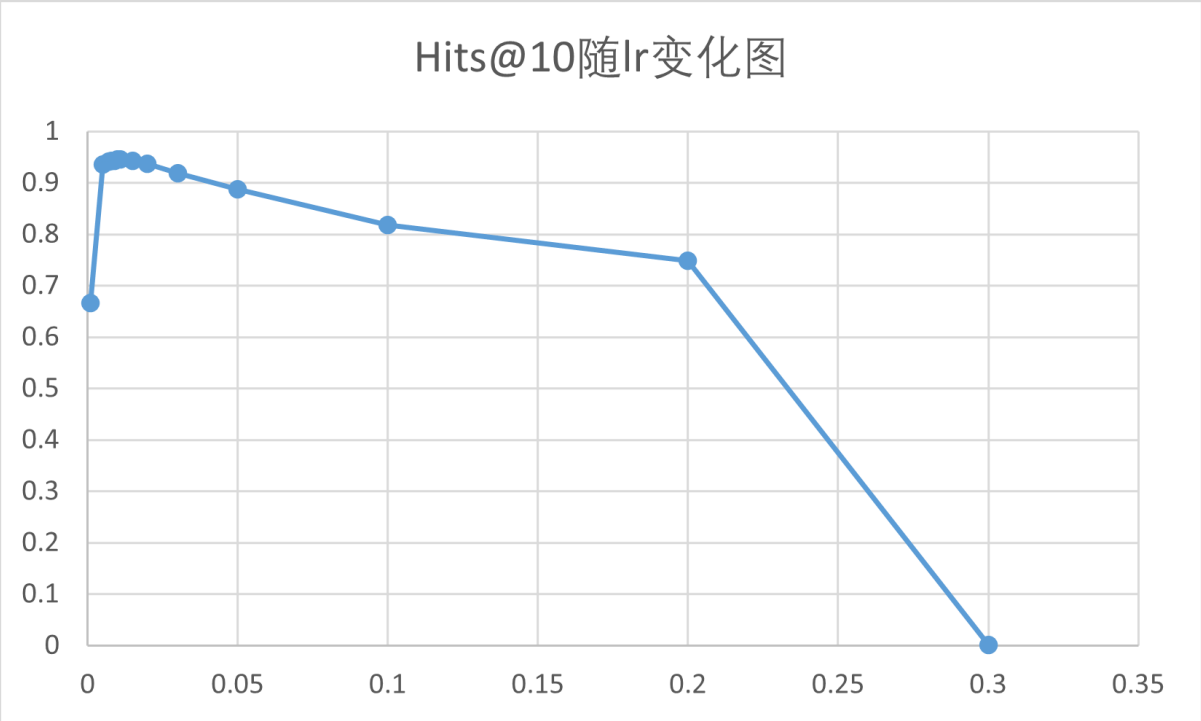
值得一提的是，在实验的一开始，我试图仿照实验三，修改 yaml 中的 margin 参数来优化模型的训练结果，但是事实上 margin 这一参数只在 Margin_Loss 函数中被调用，而在实验要求使用的 UKGE 模型当中，我们所使用的 loss 函数为 UKGE_Loss 函数，而这一函数并不含有 margin 参数，因此在本实验中，修改 margin 参数并不会对实验结果有任何的影响。

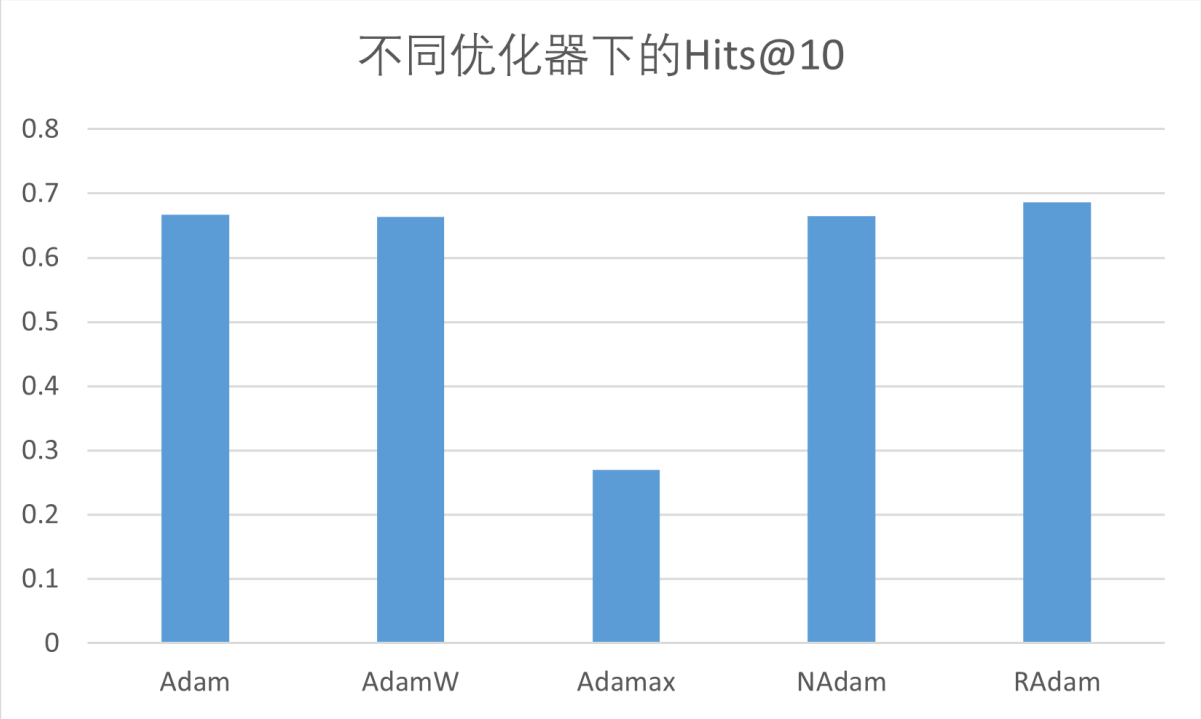
最后，综合考虑多方面因素，我选择了以下参数进行优化：

- **学习率 (lr)**
- **优化器 (optim_name)**

2.6 结果分析

2.6.1 ppi5k 数据集



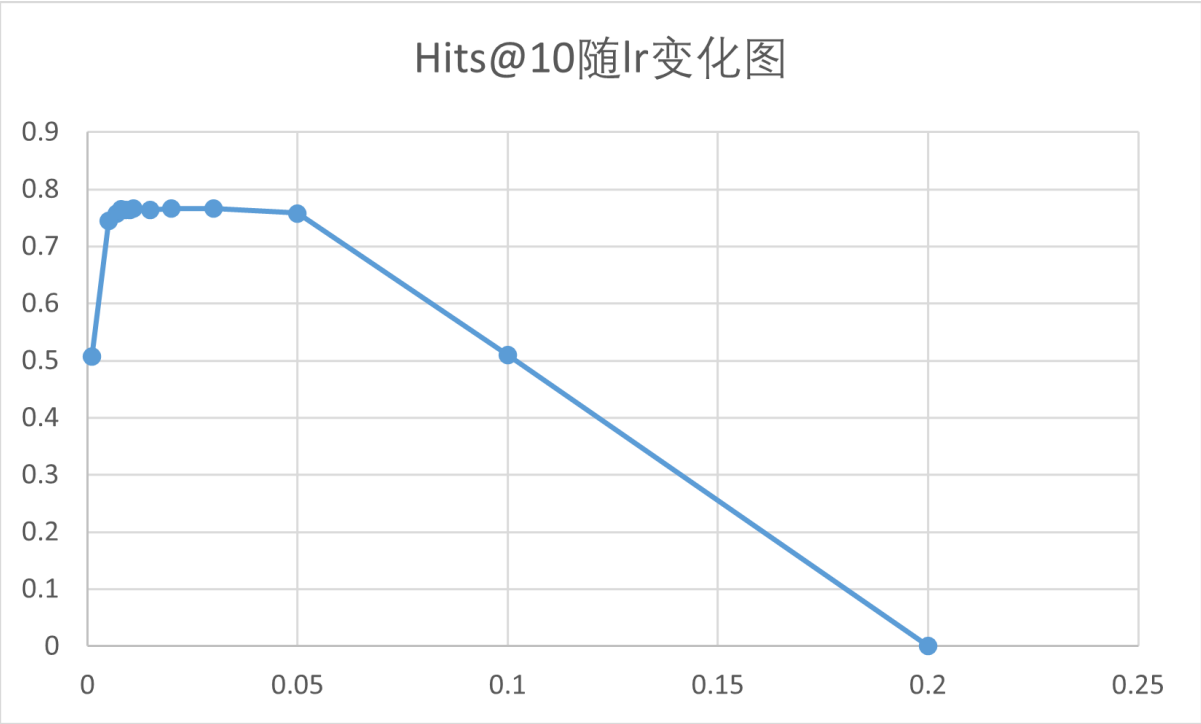


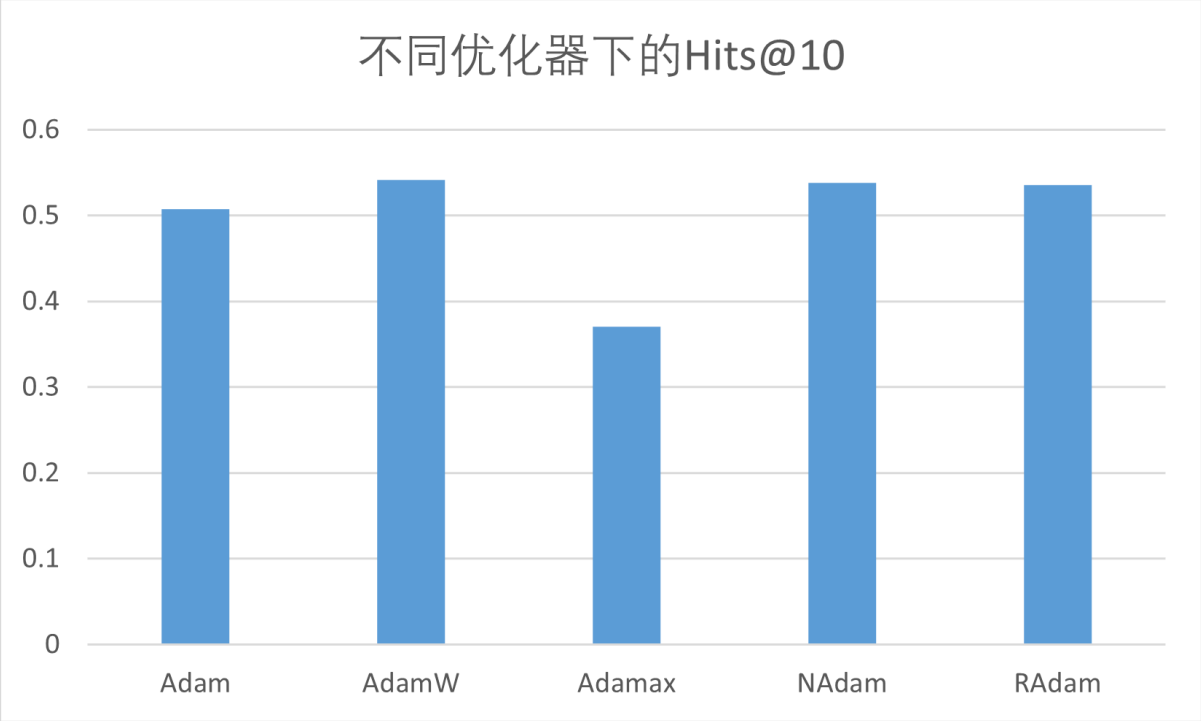
Test_MAE	0.030680000781201294
Test_MSE	0.0035379999317228794
Test_hits@1	0.4755989909172058
Test_hits@10	0.9485269784927368
Test_hits@3	0.7838860154151917
Test_hits@5	0.869337022304534 9
Test_mr	8.480156898498535
Test_mrr	0.6505380272865295
Test_raw_hits@1	0.026196999475359917
Test_raw_hits@10	0.21942900121212006
Test_raw_hits@3	0.0741709992289543 2
Test_raw_hits@5	0.1211329996585846
Test_raw_mr	41.4901008605957
Test_raw_mrr	0.09532199800014496
Test_raw_wmr	34.97789764404297
Test_raw_wmrr	0.11826200038194656
Test_wmr	6.9081878662109375
Test_wmrr	0.6705139875411987

图 5: ppi5k 最佳参数训练结果

从实验结果中可以看到，为了取得较好的训练结果，我们最好采用接近于 0.01 的 learning rate，而在优化器的选择方面，Adam、AdamW、NAdam 和 RAdam 都取得了非常优秀的成绩，其中 RAdam 效果最佳。

2.6.2 nl27k 数据集





{'Test_MAE': 0.0987870022654533 4,
'Test_MSE': 0.0328769981861114 5,
'Test_hits@1': 0.514607012271881 1,
'Test_hits@10': 0.754025995731353 8,
'Test_hits@3': 0.651988029479980 5,
'Test_hits@5': 0.697947978973388 7,
'Test_mr': 237.722961425781 25,
'Test_mrr': 0.601544976234436,
'Test_raw_hits@1': 0.351431995630264 3,
'Test_raw_hits@10': 0.599543988704681 4,
'Test_raw_hits@3': 0.486319005489349 37,
'Test_raw_hits@5': 0.53769397735595 7,
'Test_raw_mr': 288.088439941406 25,
'Test_raw_mrr': 0.440708994865417 5,
'Test_raw_wmr': 255.937042236328 12,
'Test_raw_wmrr': 0.464087992906570 43,
'Test_wmr': 208.596755981445 3,
'Test_wmrr': 0.645470023155212 4}

图 6: nl27k 最佳参数训练结果

从实验结果中可以看到，为了取得较好的训练结果，我们最好采用介于 0.005 到 0.05 之间的 learning rate，而在优化器的选择方面，AdamW、NAdam 和 RAdam 都取得了优秀的成绩，其中 AdamW 的效果最佳。