

知识图谱补全实验

实验环境

所有的实验都需要基于 `pytorch` 构建，不能使用现有的开源库。

指标

Hits@10: 前十命中率，如果预测结果中，正确的关系出现在预测结果排序前十就算该结果正确。

$$\text{Hits@10} = \frac{\text{正确答案排序前十总数}}{\text{测试集总数}}$$

Mean Rank: 正确答案平均排序位次

$$\text{Mean Rank} = \frac{\sum_i^n \text{第 } i \text{ 个答案的排序位次}}{\text{测试集总数}}$$

数据集

本次实验在数据集 WN18(WordNet) 以及 FB15K 上进行，数据集已经在压缩包中提供

参考

TransE 论文: https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf

TransH 论文: <https://ojs.aaai.org/index.php/AAAI/article/view/8870/8729>

TransR 论文: https://nlp.csai.tsinghua.edu.cn/~lyk/publications/aaai2015_transr.pdf

参考实现: <https://github.com/thunlp/OpenKE/tree/OpenKE-PyTorch/openke/module/model>

数据集来源: <https://github.com/ZhenfengLei/KGDatasets>

由于 TransE、TransH、TransR 训练流程相似，建议算法和训练评估流程分开写

一种 TransE 的实现

```
import torch
import torch.nn as nn
import torch.functional as F
class TransE(nn.Module):
    def __init__(self, ent_num, rel_num, device, dim=100, norm=1, margin=2.0,
alpha=0.01):
        super(TransE, self).__init__()
        self.ent_num = ent_num
        self.rel_num = rel_num
        self.device = device
        self.dim = dim
        self.norm = norm # 使用L1范数还是L2范数
        self.margin = margin
        self.alpha = alpha

# 初始化实体和关系表示向量
```

```

self.ent_embeddings = nn.Embedding(self.ent_num, self.dim)
torch.nn.init.xavier_uniform_(self.ent_embeddings.weight.data)
self.ent_embeddings.weight.data =
F.normalize(self.ent_embeddings.weight.data, 2, 1)

self.rel_embeddings = nn.Embedding(self.rel_num, self.dim)
torch.nn.init.xavier_uniform_(self.rel_embeddings.weight.data)
self.rel_embeddings.weight.data =
F.normalize(self.rel_embeddings.weight.data, 2, 1)

# 损失函数
self.criterion = nn.MarginRankingLoss(margin=self.margin)

def get_ent_resps(self, ent_idx): #[batch]
    return self.ent_embeddings(ent_idx) # [batch, emb]

def distance(self, h_idx, r_idx, t_idx):
    h_embs = self.ent_embeddings(h_idx) # [batch, emb]
    r_embs = self.rel_embeddings(r_idx) # [batch, emb]
    t_embs = self.ent_embeddings(t_idx) # [batch, emb]
    scores = h_embs + r_embs - t_embs

    # norm 是计算 loss 时的正则化项
    norms = (torch.mean(h_embs.norm(p=self.norm, dim=1) - 1.0)
              + torch.mean(r_embs ** 2) +
              torch.mean(t_embs.norm(p=self.norm, dim=1) - 1.0)) / 3

    return scores.norm(p=self.norm, dim=1), norms

def loss(self, positive_distances, negative_distances):
    target = torch.tensor([-1], dtype=torch.float, device=self.device)
    return self.criterion(positive_distances, negative_distances, target)

def forward(self):
    pass
    # todo

```

TransE

- 补充上述TransE的实现，基于该实现或参考其他实现方式，完成TransE的训练与评估
- TransE在WN18训练并测试关系预测指标，给出Hits@10以及Mean Rank，参考指标 Hits@10=0.75, Mean Rank=263，要求预测指标不低于参考指标的85%。如要求Hits@10 > 0.64 且 Mean Rank < 310。
- TransE在FB15K训练并测试关系预测指标，给出Hits@10以及Mean Rank，参考指标 Hits@10=0.34, Mean Rank=243，要求预测指标不低于参考指标的85%
 - 进一步测试FB15k中，预测头实体以及尾实体n-to-n的Hits@10

TransH

- 实现TransH（参考TransE），完成训练与评估
- TransH在WN18训练并测试关系预测指标，给出Hits@10以及Mean Rank，参考指标 Hits@10=0.75, Mean Rank=318，要求预测指标不低于参考指标的85%
- TransH在FB15K训练并测试关系预测指标，给出Hits@10以及Mean Rank，参考指标 Hits@10=0.42, Mean Rank=211，要求预测指标不低于参考指标的85%
 - 进一步测试FB15k中，预测头实体以及尾实体n-to-n的Hits@10，并与TransE进行对比，观察TransH是否在n-to-n预测时有所改善
 - （选做）TransH论文中给出了两种不同的负采样方法，*unif*和*bern*，分别使用两种负采样方法进行实验

TransR

- 实现TransR，完全训练与评估
- TransR在WN18训练并测试关系预测指标，给出Hits@10以及Mean Rank，参考指标 Hits@10=0.79, Mean Rank=238，要求预测指标不低于参考指标的85%
- TransH在FB15K训练并测试关系预测指标，给出Hits@10以及Mean Rank，参考指标 Hits@10=0.43, Mean Rank=226，要求预测指标不低于参考指标的85%
 - 进一步测试FB15k中，预测头实体以及尾实体n-to-n的Hits@10，并与TransE进行对比，观察TransR是否在n-to-n预测时有所改善

需要提交的文件

- 1.实验报告，参考 关系预测模板.docx
- 2.TransE.py，代码需要给出一定的注释说明
- 3.TransH.py，代码需要给出一定的注释说明
- 4.TransR.py，代码需要给出一定的注释说明