# TransE算法及应用示例

## 一、内容简介

TransE算法是知识图谱表示学习算法中的首个基于Translation的方法，来源于2014年的论文《Translating Embeddings for Modeling Multi-relational Data》。知识图谱是一种用图结构的方式对真实世界进行建模的工具，其直观强大的表示能力使得人们能够很容易理解并进行应用，但是计算机只能处理0和1这样的数字，因此，知识图谱表示学习的目的就是将知识图谱中的元素，也即节点和边表示成可计算的向量形式，从而可供计算机进行计算。

TransE算法首次采用了基于Translation的方法，作为该类知识图谱表示学习方法的开山之作，也在后续许多工作中进行了应用。此外，后续许多针对于TransE算法的改进工作，如TransH、TransR等，共同构成了知识图谱表示学习的Trans系列算法。

## 二、理论基础

### 2.1 知识点一 知识图谱基本理论

2.1.1 什么是知识图谱

知识图谱是一种用图模型来描述知识和建模世界万物之间的关联关系的技术方法。知识图谱旨在利用图结构建模、识别和推断事物之间的复杂关联关系和沉淀领域知识，是实现人工智能的重要基石。

知识图谱中的基本的知识表示形式为（头实体、关系、尾实体）这样的三元组形式，一个三元组可以认为表示了一个具体的知识。

2.1.2 知识图谱的发展简史

二十世纪六十年代提出的语义网络，可以认为是知识图谱的雏形。如下图所示是一个简单的发展时间轴，概括来讲，知识的表示形式从最初的简单的图结构化的只是表示方式，逐渐演变成了如今具有识别、发现、推断事物与概念的复杂关系的可计算的模型，也即是现在的知识图谱。



### 2.2 知识点二 表示学习与知识图谱的表示学习

2.2.1 表示学习

表示学习，英文为Representation Learning，又被称为特征学习或表征学习。在深度学习领域，表示是指通过模型的参数，采用何种形式、何种方式来表示模型的输入观测样本，其能将原始数据转换成为能够被机器学习有效利用的一种形式，最常见的就是把一个原始数据样本表示为一个低维、连续空间中的向量。

表示学习得到的低维向量表示是一种分布式表示(distributed representation)。之所以如此命名，是因为孤立地看向量中的每一维，都没有明确对应的含义；而综合各维形成一个向量，则能够表示对象的语义信息。

2.2.2 知识图谱的表示学习

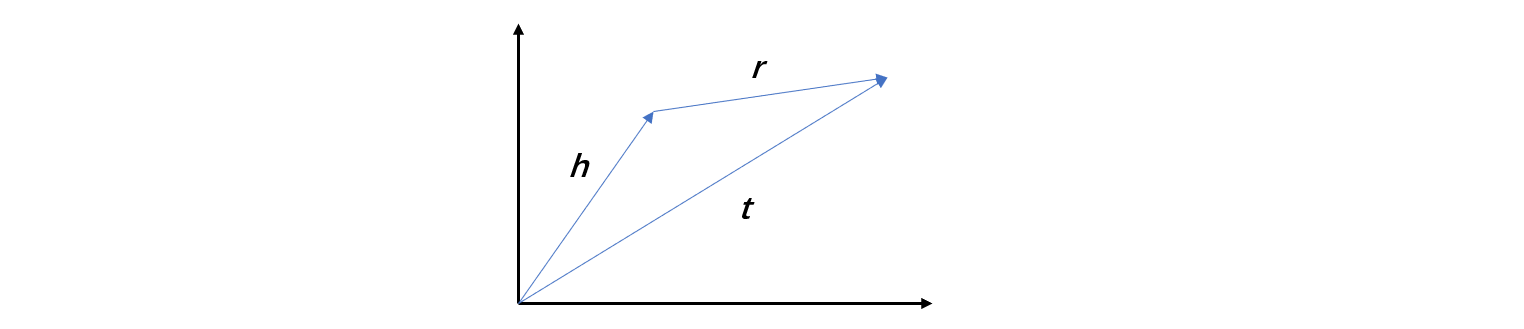
知识图谱的表示学习，也就是对图谱中所包含的知识（实体或关系）进行表示学习。通过将实体和关系投影到低维向量空间，就能够实现对实体和关系的语义信息的表示，可以高效计算实体、关系之间的复杂的语义关联。

常见的知识图谱表示学习的应用包括：（1）相似度计算：计算实体之间的相似度；（2）知识图谱补全：知识图谱的构建是一个动态丰富的过程，利用知识表示模型，可以预测两个实体之间的关系，进而补全未包含的知识；（3）NLP任务：如知识抽取、问答系统、实体链接等；（4）推荐系统：将商品和用户等实体进行表示学习从而利用知识图谱的语义信息进行推荐；

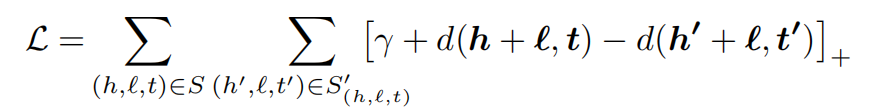
### 2.3 知识点三 TransE核心思想

前面已经提到知识图谱表示学习，通俗来讲就是把图谱中的实体和关系表示成低维、连续向量空间中的向量的形式，从而表示其语义并进行计算。

简单来讲，TransE算法就是将知识图谱中的每个三元组（head，relation，tail）表示为对应的向量***h，r，t***，但是需要满足一个约束条件，即***h*** + ***r*** ≈ ***t*。**也即头实体向量和关系向量的和要尽可能与尾实体向量相等，如果我们用一个二维平面坐标系来进行说明的话，即嵌入向量之间的关系应当如下图所示：



如果将实体和关系看作两个矩阵，实体矩阵为n×d，表示n个实体，每个实体的embedding维度为d维；关系矩阵维r×d，表示r个关系，每个关系的embedding维度为d维；因此，在TransE训练完成后，理想情况下，从实体矩阵和关系矩阵中抽取一对对应的向量，其相加得到的结果为该三元组的尾实体的向量，也即实体矩阵中的某一列向量。因此，在这里我们可知，算法的Translation的含义，可以理解为翻译、即实体之间通过关系向量的翻译建立连接，从而翻译语义；也可以理解为转移，即实体之间的关系通过关系向量进行转移，从而建立语义上的连接。因此，TransE算法所要优化的目标函数如下所示：

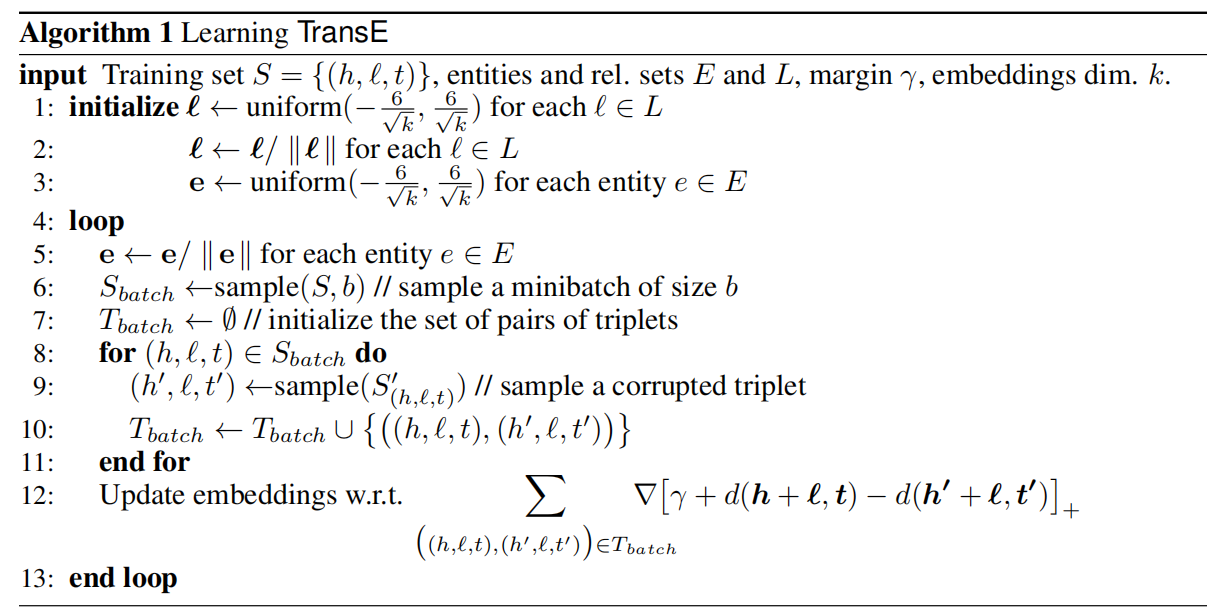


### 2.4 知识点四 TransE算法流程

2.4.1 总体算法流程

TransE算法的总体流程如下所示：

1. 首先为每个实体和关系采用Xavier初始化；
2. 从所有三元组样本中采样得到一个batch的训练样本；
3. 基于该batch的三元组样本，替换头实体或尾实体进而构造负样本三元组集合；
4. 通过SGD算法最小化损失函数从而更新embedding表示；



2.4.2 算法输入输出

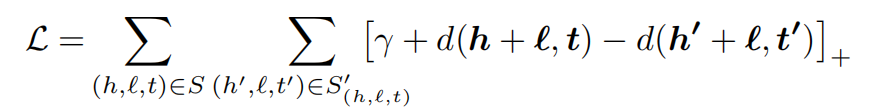
算法输入：三元组集合S；实体集合E、关系集合L；嵌入维度k；

算法输出：实体和关系的embedding向量表示；

2.4.3 负样本构造

由于负样本的构造是基于已有的三元组集合来进行头实体或尾实体的替换，从而构造出图谱中并不存在的三元组，并最大化正负三元组之间的差异；因此，实际应用中可以选择随机头实体或尾实体进行替换、也可以设置两个参数tph或hpt来选择更换头实体还是更换尾实体。

2.4.4 损失函数及优化



TransE算法要优化的损失函数如上所示，d(·)表示的即是三元组的头实体加关系与尾实体向量的关系，而h’和t’是替换后的负样本三元组中的头实体或尾实体。损失函数的优化的理想情况为对于正确的三元组，其d尽可能小，负样本三元组其d要尽可能大，这样整体的Loss趋近于0。因此，在SGD算法的参数更新过程中，正样本中的h和r逐渐减小，t逐渐增大；而负样本中h’和r逐渐增大，而t’则逐渐减小。

### 2.5 知识点五 TransE算法优缺点

2.5.1 TransE算法优点

（1）首次提出采用基于Translation的知识表示学习算法；

（2）容易训练，可以扩展到比较大的图谱当中；

（3）相较于以往模型，参数较少、计算复杂度低、能直接建立实体和关系之间的语义联系；

2.5.2 TransE算法缺点

（1）适用于处理一对一的单一关系型数据，对于复杂关系的表示学习效果不太理想；

（2）仅利用图谱结构信息，忽视了多源信息的融合；

## 三、演示验证

### 3.1 运行环境搭建

无非常规环境或特殊工具包。

### 3.2 实验任务一 数据集加载

步骤1.1 读取数据集

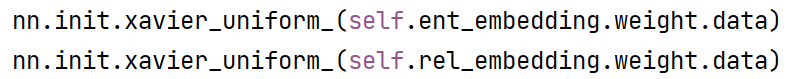
读取数据集的triples.txt（三元组文件）、entity2id.txt（实体文件）、relation2id.txt（关系文件）；

步骤1.2 提取数据信息

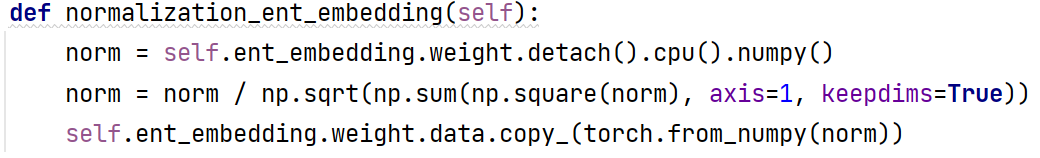
提取三元组信息、实体信息和关系信息、分别存储到三个list中；

### 3.3 实验任务二 TransE核心代码

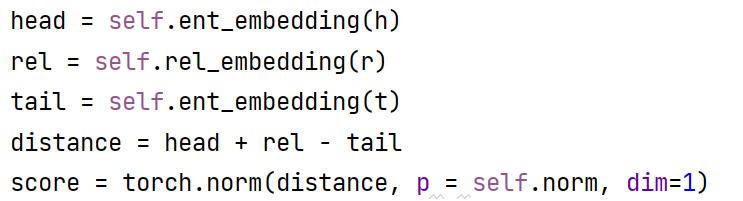
步骤2.1 初始化实体和关系的embedding



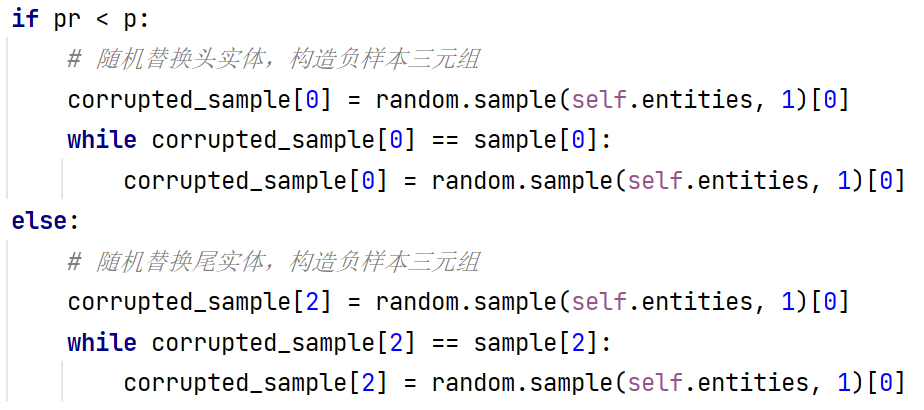
步骤2.2 对实体和关系的embedding进行规范化



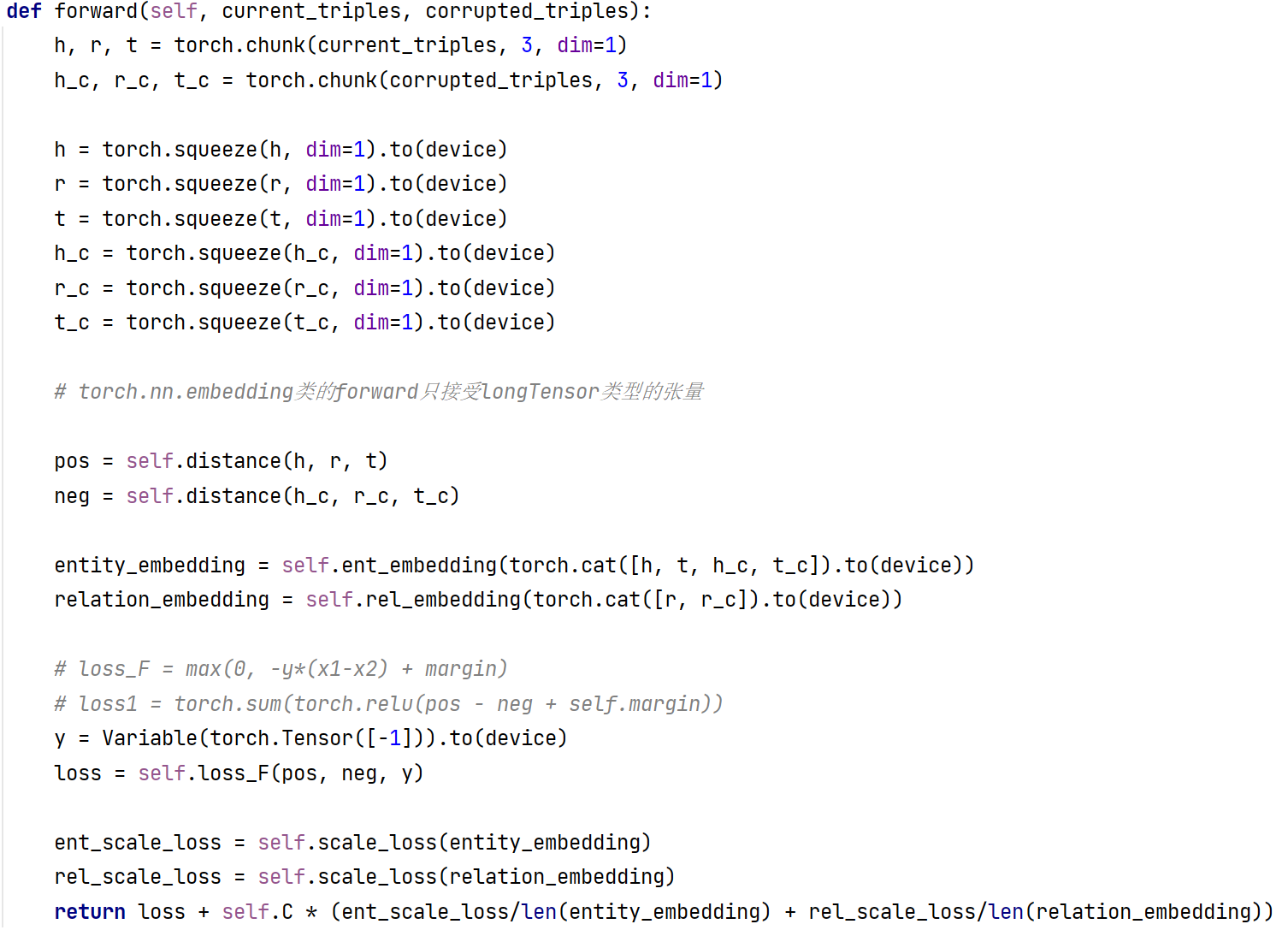
步骤2.3 定义目标函数



步骤2.4 构造负样本三元组



步骤2.5 定义模型的计算过程



### 3.4 实验任务三 训练及测试

步骤3.1 载入训练集进行模型训练

Dataloader()、Model()、Model.train()

步骤3.2 使用测试集进行测试

Dataloader()、Model.test()

步骤3.3 训练过程参数的可视化

Visulization()

步骤3.4 输出实体和关系的embedding结果

Output\_Embeddings()

## 四、实验环境

以下仅供参考：

操作系统：windows10

平台软件：Anaconda、torch、Python

## 五、界面设计

