实验报告

学号姓名：201250104 苏致成

Title：End-to-end Structure-Aware Convolutional Networks for Knowledge Base Completion

主题：用于知识库构建的端到端结构感知卷积网络

一、摘要与介绍翻译

1.1 摘要翻译

知识图谱的嵌入一直是知识库构建的一个活跃的研究课题，从最初的 $TransE$、$TransH$、$DistMult$ 等逐步发展到目前最先进的 $ConvE$。$ConvE$ 在嵌入和多层非线性特征上使用二维卷积来建模知识图。该模型可以有效地训练，并可扩展到大型知识图。

然而，$ConvE$ 的嵌入空间中没有结构强制。最近的图卷积网络通过成功地利用图的连通结构提供了另一种学习图节点嵌入的方法。在这项工作中，我们提出了一种新型的端到端结构化软件卷积网络（$SACN$），它能够同时吸纳 GCN 和 $ConvE$ 的优点。$SACN$ 由加权图卷积网络（$WGCN$）的编码器和卷积网络 $ConvTransE$ 的解码器组成。$WGCN$ 利用了知识图节点结构、节点属性和边缘关系类型。它具有可学习的权重，可以适应本地聚合中使用的邻居信息量，从而实现更精确的图形节点嵌入。

在 $WGCN$ 中，图中的节点属性表示为其他节点。解码器 $Conv-TransE$ 使最先进的 $ConvE$ 能够在实体和关系之间进行转换，同时保持与 $ConvE$ 相同的链路预测性能。我们在标准 $FB15k-237$ 和 $WN18RR$ 数据集上证明了提出的 $SACN$ 的有效性，并且从 $HITS@1, HITS@3$ 和$HITS@10$ 指标来看，它比最先进的 $ConvE$ 提高了约 10%。

1.2 介绍翻译

近年来，大型知识库如 $Freebase$（$Bollacker$ 等人，2008年）、$DBpedia$（$Auer$等人，2007年）、$NELL$（$Carlson$ 等人，2010年）和$YAGO3$（$Mahdisoltani$、$Biega$ 和 $Suchanek$，2013年）被建立以存储关于共同事实的结构化信息。知识库是多关系图，其节点表示实体，边表示实体之间的关系，边用不同的关系标记。这些关系以（s，r，o）三元组的形式组织（例如，实体 s=$AbrahamLincoln$，关系r=$DateOfBirth$，实体o= $02-12-1809$）。这些知识库广泛用于网络搜索、推荐和问答系统中。

尽管这些知识库已经包含了数百万个实体和三元组，但与现有事实和新增加的现实世界知识相比，它们还远远不够完整。因此，为了在现有三元组的基础上预测新的三元组，从而进一步扩展知识库，完成知识库非常重要。

知识库构建的最新活跃研究领域之一是知识图嵌入：它对连续低维向量空间中的实体和关系的语义进行编码（称为嵌入）。然后，这些嵌入用于预测新的关系。从一种简单有效的方法 $TransE$（$Bordes\ et\ al.2013$）开始，人们提出了许多知识图嵌入方法，如 $TransH$（$Wang\ et\ al.2014$）、$TransR$（$Lin\ et\ al.2015$）、$DistMult$（$Y\ ang\ et\ al.2014$）、$TransD$（$Ji\ et\ al.2015$）、$CompleEx$（$Trouillon\ et\ al.2016$）、$STransE$（$Nguyen\ et\ al.2016$）。一些调查（$Nguyen$ 2017；$Wang\ et\ al.2017$）给出了这些嵌入方法的详细信息和比较。

最新的 $ConvE$（$Dettmersetal.2017$）模型在嵌入和多层非线性特征上使用二维卷积，并在用于知识图链接预测的通用基准数据集上实现了最先进的性能。在 $ConvE$中， s 和 r 的嵌入被重新变形并连接到一个输入矩阵中，然后传送到卷积层。$n \times n$的卷积滤波器用于输出跨不同维嵌入项的特征映射。因此，$ConvE$ 不保留 $TransE$ 的平移属性，$TransE$是一种附加的嵌入向量操作：$es+er \approx eo$（$Nguyen$ 等人，2017）。在本文中，我们删除了 $ConvE$ 的变形步骤，并直接在s和r的相同维度上应用卷积滤波器。与原始的 $ConvE$ 相比，这个改动具有更好的性能，并且具有直观的解释，保持了嵌入三元组（es，er，eo）中s，r和o的全局学习度量相同。我们将此嵌入命名为 $ConvTransE$。

$ConvE$也没有将知识图中的连通结构合并到嵌入空间中。相比之下，图卷积网络已成为创建节点嵌入的有效工具，该节点嵌入可聚合每个节点的图邻域中的本地信息（$Kipf\ and\ Welling$ 2016b; $Hamilton$, $Ying$ 和 $Leskovec$ 2017a; $Kipf$ 和 $Welling$ 2016a; $Pham\ et\ al.$ 2017; $Shang\ et\ al.$ 2018）。GCN 模型还有其他优点（$Hamilton$、$Ying$和 $Leskovec$ 2017b），例如利用与节点相关的属性。它们还可以在计算每个节点的卷积时采用相同的聚合方案，这可以被视为一种正则化方法，并提高效率。尽管可扩展性最初是GCN模型的一个问题，但最新的高效数据 GCN，也就是$PinSage$（$Ying$ 等人，2018），能够处理数十亿级别的节点和边。

本文提出了一种端到端的图结构感知卷积网络（$SACN$），它将 GCN 和 $ConvE$ 的所有优点结合在一起。$SACN$ 由加权图卷积网络（$WGCN$）的编码器和卷积网络 $Conv-TransE$ 的解码器组成。

$WGCN$ 利用知识图节点结构、节点属性和关系类型。它具有可学习的权重来确定本地聚合中使用的邻居信息量，从而实现更精确的图节点嵌入。节点属性作为附加属性添加到 $WGCN$ 中以便于集成。$WGCN$ 的输出成为解码器 $ConvTransE$ 的输入。$ConvTransE$ 与 $ConvE$ 相似，但不同的是，$ConvTransE$ 保持了实体和关系之间的转化特性。

我们证明出 $ConvTransE$ 比 $ConvE$ 表现更好，并且在标准基准数据集中，我们的 $SACN$ 在 $ConvTransE$ 的基础上进一步提高。我们的模型和实验的代码是公开的。我们的贡献总结如下：

- 我们提出了一个端到端网络学习框架 $SACN$，它同时利用了 GCN 和$ConvTransE$ 。编码器GCN模型利用图形结构和图形节点的属性。解码器 $ConvTransE$ 通过特殊卷积简化了 $ConvE$ ，并保持了 $TransE$ 的转化特性和 $ConvE$ 的预测性能；
- 我们在标准 $FB15k-237$ 和 $WN18RR$ 数据集上证明了我们提出的 $SACN$ 的有效性，并在 $HITS@1, HITS@3$和$HITS@10$ 数据集上显示出和最先进的 $ConvE$ 相比大约有 10%的改进。

二、问题描述

2.1 传统模型问题

传统模型主要存在以下问题：

- 只对关系型三元组进行建模，而忽略了节点的属性。
- 没有将节点的结构化信息纳入考虑范围。

因此，随着图神经网络的提出与发展，使得 GCN 成为了获取图结构化信息的利器。因此，这里运用 $WGCN$ 模型，使得节点的结构化信息得以充分利用。并根据最新的用卷积解决知识图谱嵌入问题的研究，改进了 $ConvE$ 模型，使其保留了平移特性，构建了 $Conv-TransE$模型，将上述两个模型一个作为编码器，一个作为解码器，构建了整个 $SACN$ 模型，用于完善知识图谱。

三、输入、输出、模型算法描述

3.1 输入

3.1.1 整体输入

这里选取了三个知识图谱的基准数据集来评估链路预测的性能，分别为 $FB15k-237$、$WN18RR$、$FB15k-237-Attr$。

- $FB15k-237$： $FB15k$ 的子集，删除了逆向关系等。
- $WN18RR$：由 $WN18$ 创建而来，且 $WN18$ 是 $WordNet$ 的一个子集。
- $FB15k-237-Attr$：由 $FB15k-237$ 创建而来。

以下描述了数据集中的各指标数量：

Dataset	FB15k-237	WN18RR	FB15k-237-Attr
Entities	14,541	40,943	14,744
Relations	237	11	484
Train Edges	272,115	86,835	350,449
Val. Edges	17,535	3,034	17,535
Test Edges	20,466	3,134	20,466
Attributes Triples	—	—	78,334
Attributes	—	—	203

3.1.2 输入解析

以 $FB15k-237$ 为例，分析其数据结构。

- **entity2id.txt**
  - 实体和id对
  - 数据格式部分截图如下：

```
/m/034wx3	15
/m/0c5x_	16
/m/0fqpg6b	17
/m/03qcq	18
/m/01f_w	19
```

- **relation2id.txt**
  - 关系和id对
  - 数据格式部分截图如下：

```
/people/appointed_role/appointment./people/appointment/appointed_by	0
/location/statistical_region/rent50_2./measurement_unit/dated_money_value/currency	1
/tv/tv_series_episode/guest_stars./tv/tv_guest_role/actor	2
/music/performance_role/track_performances./music/track_contribution/contributor	3
/medicine/disease/prevention_factors	4
```

- **train.txt**
  - 训练集三元组（实体，实体，关系）
- **test.txt**
  - 测试集三元组（实体，实体，关系）

3.1.3 Conv-TransE的输入

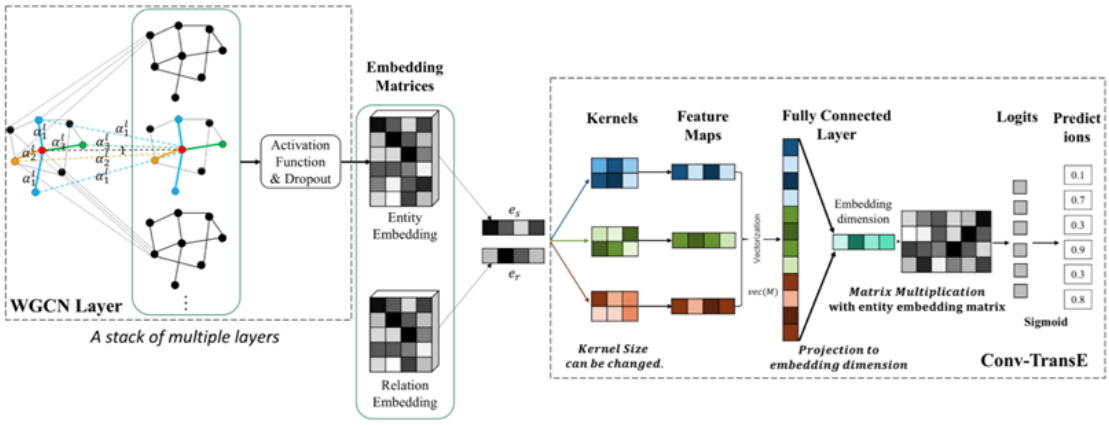


Figure 1: An illustration of our end-to-end Structure-Aware Convolutional Networks model. For encoder, a stack of multiple WGCN layers builds an entity/node embedding matrix. For decoder, e_s and e_r are fed into *Conv-TransE*. The output embeddings are vectorized and projected, and matched with all candidate e_o embeddings via inner products. A logistic sigmoid function is used to get the scores.

模型的整体框架为 $WGCN$ 模块提取实体 $embedding$ 表示，再将实体 $embedding$ 表示作为 $ConvTransE$ 模块的输入。

3.2 输出

输出为模型在不同数据集上正确率。详细结果参见后文。

3.2.1 ConvTransE

此数据是在 $FB15k-237$ 数据集上运行 $ConvTransE$ ，并将 $epoch$ 设置为300。原因如下：

1. 根据论文后续关于收敛性的分析，可以看出当 $epoch$ 数量达到300时， $Hits@n$ 和 MRR 都达到收敛。
2. 因为本模型运算量巨大，基于节省经费方面考虑。

```
2022-10-15 16:39:15.105409 (INFO): ########################################
2022-10-15 16:39:15.105421 (INFO):           COMPLETED EPOCH: 300
2022-10-15 16:39:15.105430 (INFO): train Loss: 0.0013213        99% CI: (0.0011913, 0.0014514), n=68
2022-10-15 16:39:15.105444 (INFO): ########################################
2022-10-15 16:39:15.105454 (INFO):

saving to saved_models/FB15k-237_ConvTransE_0.0_0.2.model
2022-10-15 16:39:15.119135 (INFO):
2022-10-15 16:39:15.119183 (INFO): --------------------------------------------------
2022-10-15 16:39:15.119191 (INFO): dev_evaluation
2022-10-15 16:39:15.119198 (INFO): --------------------------------------------------
2022-10-15 16:39:15.119206 (INFO):
2022-10-15 16:39:48.529280 (INFO): Hits left @1: 0.3427734375
2022-10-15 16:39:48.530673 (INFO): Hits right @1: 0.14723115808823528
2022-10-15 16:39:48.533184 (INFO): Hits @1: 0.24500229779411764
2022-10-15 16:39:48.534492 (INFO): Hits left @2: 0.42830882352941174
2022-10-15 16:39:48.535695 (INFO): Hits right @2: 0.21065027573529413
2022-10-15 16:39:48.538134 (INFO): Hits @2: 0.3194795496323529
2022-10-15 16:39:48.539448 (INFO): Hits left @3: 0.4802389705882353
2022-10-15 16:39:48.540749 (INFO): Hits right @3: 0.2552274816176471
2022-10-15 16:39:48.543233 (INFO): Hits @3: 0.3677332261029412
2022-10-15 16:39:48.544490 (INFO): Hits left @4: 0.5137293198529411
2022-10-15 16:39:48.545795 (INFO): Hits right @4: 0.28848805147058826
2022-10-15 16:39:48.548303 (INFO): Hits @4: 0.4011086856617647
2022-10-15 16:39:48.549687 (INFO): Hits left @5: 0.5404986213235294
2022-10-15 16:39:48.550940 (INFO): Hits right @5: 0.31784237132352944
2022-10-15 16:39:48.553492 (INFO): Hits @5: 0.42917049632352944
2022-10-15 16:39:48.554677 (INFO): Hits left @6: 0.5637637867647058
2022-10-15 16:39:48.555945 (INFO): Hits right @6: 0.3392118566176471
2022-10-15 16:39:48.558476 (INFO): Hits @6: 0.45148782169117646
2022-10-15 16:39:48.559732 (INFO): Hits left @7: 0.5827205882352942
2022-10-15 16:39:48.560935 (INFO): Hits right @7: 0.36046645220588236
2022-10-15 16:39:48.563365 (INFO): Hits @7: 0.47159352022058826
2022-10-15 16:39:48.564647 (INFO): Hits left @8: 0.5984604779411765
2022-10-15 16:39:48.565832 (INFO): Hits right @8: 0.37545955882352944
```

```
2022-10-15 16:39:48.529280 (INFO): Hits left @1: 0.3427734375
2022-10-15 16:39:48.530673 (INFO): Hits right @1: 0.14723115808823528
2022-10-15 16:39:48.533184 (INFO): Hits @1: 0.24500229779411764
2022-10-15 16:39:48.534492 (INFO): Hits left @2: 0.42830882352941174
2022-10-15 16:39:48.535695 (INFO): Hits right @2: 0.21065027573529413
2022-10-15 16:39:48.538134 (INFO): Hits @2: 0.3194795496323529
2022-10-15 16:39:48.539448 (INFO): Hits left @3: 0.4802389705882353
2022-10-15 16:39:48.540749 (INFO): Hits right @3: 0.2552274816176471
2022-10-15 16:39:48.543233 (INFO): Hits @3: 0.3677332261029412
2022-10-15 16:39:48.544490 (INFO): Hits left @4: 0.5137293198529411
2022-10-15 16:39:48.545795 (INFO): Hits right @4: 0.28848805147058826
2022-10-15 16:39:48.548303 (INFO): Hits @4: 0.4011086856617647
2022-10-15 16:39:48.549687 (INFO): Hits left @5: 0.5404986213235294
2022-10-15 16:39:48.550940 (INFO): Hits right @5: 0.31784237132352944
2022-10-15 16:39:48.553492 (INFO): Hits @5: 0.42917049632352944
2022-10-15 16:39:48.554677 (INFO): Hits left @6: 0.5637637867647058
2022-10-15 16:39:48.555945 (INFO): Hits right @6: 0.3392118566176471
2022-10-15 16:39:48.558476 (INFO): Hits @6: 0.45148782169117646
2022-10-15 16:39:48.559732 (INFO): Hits left @7: 0.5827205882352942
2022-10-15 16:39:48.560935 (INFO): Hits right @7: 0.36046645220588236
2022-10-15 16:39:48.563365 (INFO): Hits @7: 0.47159352022058826
2022-10-15 16:39:48.564647 (INFO): Hits left @8: 0.5984604779411765
2022-10-15 16:39:48.565832 (INFO): Hits right @8: 0.37545955882352944
2022-10-15 16:39:48.568233 (INFO): Hits @8: 0.48696001838235294
2022-10-15 16:39:48.569529 (INFO): Hits left @9: 0.6121323529411765
2022-10-15 16:39:48.570820 (INFO): Hits right @9: 0.3890165441176471
2022-10-15 16:39:48.573246 (INFO): Hits @9: 0.5005744485294118
2022-10-15 16:39:48.574511 (INFO): Hits left @10: 0.6222426470588235
2022-10-15 16:39:48.575760 (INFO): Hits right @10: 0.4055606617647059
2022-10-15 16:39:48.578212 (INFO): Hits @10: 0.5139016544117647
2022-10-15 16:39:48.580328 (INFO): Mean rank left: 159.90286075367646
2022-10-15 16:39:48.581527 (INFO): Mean rank right: 304.7682674632353
2022-10-15 16:39:48.583578 (INFO): Mean rank: 232.33556410845588
2022-10-15 16:39:48.584817 (INFO): Mean reciprocal rank left: 0.4370035373135073
2022-10-15 16:39:48.585966 (INFO): Mean reciprocal rank right: 0.23189599400674485
2022-10-15 16:39:48.588208 (INFO): Mean reciprocal rank: 0.33444976566012613
```

3.2.2 SACN

此数据是在 $FB15k-237$ 数据集上运行 $SACN$ ，并将 $epoch$ 设置为300。原因如下：

1. 根据论文后续关于收敛性的分析，可以看出当 $epoch$ 数量达到300时， $Hits@n$ 和 MRR 都达到收敛。
2. 因为本模型运算量巨大，基于节省经费方面考虑。且方便与上述模型对比。

```
2022-10-16 02:07:28.740210 (INFO): ######################################
2022-10-16 02:07:28.740221 (INFO):           COMPLETED EPOCH: 300
2022-10-16 02:07:28.740234 (INFO): train Loss: 0.0015994        99% CI: (0.0012731, 0.0019257), n=68
2022-10-16 02:07:28.740243 (INFO): ######################################
2022-10-16 02:07:28.740252 (INFO):

saving to saved_models/FB15k-237_SACN_0.0_0.2.model
2022-10-16 02:07:28.797228 (INFO):
2022-10-16 02:07:28.797287 (INFO): ------------------------------------------------
2022-10-16 02:07:28.797295 (INFO): dev_evaluation
2022-10-16 02:07:28.797302 (INFO): ------------------------------------------------
2022-10-16 02:07:28.797309 (INFO):
2022-10-16 02:08:12.174538 (INFO): Hits left @1: 0.3759375
2022-10-16 02:08:12.175786 (INFO): Hits right @1: 0.17878676470588236
2022-10-16 02:08:12.178096 (INFO): Hits @1: 0.27736213235294118
2022-10-16 02:08:12.179265 (INFO): Hits left @2: 0.4577389705882353
2022-10-16 02:08:12.180430 (INFO): Hits right @2: 0.23427849264705882
2022-10-16 02:08:12.182582 (INFO): Hits @2: 0.3460087316176471
2022-10-16 02:08:12.183804 (INFO): Hits left @3: 0.5054756433823529
2022-10-16 02:08:12.185017 (INFO): Hits right @3: 0.27701746323529413
2022-10-16 02:08:12.187200 (INFO): Hits @3: 0.39124655330882354
2022-10-16 02:08:12.188361 (INFO): Hits left @4: 0.5404678905643224
2022-10-16 02:08:12.189525 (INFO): Hits right @4: 0.30522288602941174
2022-10-16 02:08:12.191658 (INFO): Hits @4: 0.4226114430147059
2022-10-16 02:08:12.192822 (INFO): Hits left @5: 0.5649885110294118
2022-10-16 02:08:12.209090 (INFO): Hits right @5: 0.33250919117647056
2022-10-16 02:08:12.211354 (INFO): Hits @5: 0.4487488511029412
2022-10-16 02:08:12.212496 (INFO): Hits left @6: 0.5850367647058824
2022-10-16 02:08:12.213667 (INFO): Hits right @6: 0.35422334558823353
2022-10-16 02:08:12.215805 (INFO): Hits @6: 0.4696300551470588
2022-10-16 02:08:12.216988 (INFO): Hits left @7: 0.6008340992647058
2022-10-16 02:08:12.218161 (INFO): Hits right @7: 0.37162913602941174
2022-10-16 02:08:12.220394 (INFO): Hits @7: 0.4862316176470588
2022-10-16 02:08:12.221619 (INFO): Hits left @8: 0.6167463235294118
2022-10-16 02:08:12.222820 (INFO): Hits right @8: 0.3874264705882353
2022-10-16 02:08:12.225017 (INFO): Hits @8: 0.50208639705882354
```

```
2022-10-16 02:09:03.107730 (INFO): Hits left @1: 0.34832154088050314
2022-10-16 02:09:03.109180 (INFO): Hits right @1: 0.1567433176100629
2022-10-16 02:09:03.111741 (INFO): Hits @1: 0.25253242924528303
2022-10-16 02:09:03.113113 (INFO): Hits left @2: 0.4508687106918239
2022-10-16 02:09:03.114448 (INFO): Hits right @2: 0.23265919811320756
2022-10-16 02:09:03.117052 (INFO): Hits @2: 0.34176395440251574
2022-10-16 02:09:03.118416 (INFO): Hits left @3: 0.50049528301886794
2022-10-16 02:09:03.119782 (INFO): Hits right @3: 0.27324488993710693
2022-10-16 02:09:03.122409 (INFO): Hits @3: 0.3868700864779874
2022-10-16 02:09:03.123766 (INFO): Hits left @4: 0.53405463836477986
2022-10-16 02:09:03.125128 (INFO): Hits right @4: 0.3047897012578616
2022-10-16 02:09:03.127708 (INFO): Hits @4: 0.41942216981132076
2022-10-16 02:09:03.129083 (INFO): Hits left @5: 0.5595066823899371
2022-10-16 02:09:03.130410 (INFO): Hits right @5: 0.3318632075471698
2022-10-16 02:09:03.132909 (INFO): Hits @5: 0.44568494496855345
2022-10-16 02:09:03.134247 (INFO): Hits left @6: 0.582747641509434
2022-10-16 02:09:03.135553 (INFO): Hits right @6: 0.3453989779874214
2022-10-16 02:09:03.138065 (INFO): Hits @6: 0.46907330974842767
2022-10-16 02:09:03.139378 (INFO): Hits left @7: 0.5994044811320755
2022-10-16 02:09:03.140704 (INFO): Hits right @7: 0.3721049528301887
2022-10-16 02:09:03.143271 (INFO): Hits @7: 0.48575471698113206
2022-10-16 02:09:03.144588 (INFO): Hits left @8: 0.6152260220125787
2022-10-16 02:09:03.145902 (INFO): Hits right @8: 0.38640330188679247
2022-10-16 02:09:03.148420 (INFO): Hits @8: 0.5008146619496855
2022-10-16 02:09:03.149752 (INFO): Hits left @9: 0.6272150157232704
2022-10-16 02:09:03.151076 (INFO): Hits right @9: 0.3996698113207547
2022-10-16 02:09:03.153615 (INFO): Hits @9: 0.5134424135220126
2022-10-16 02:09:03.154948 (INFO): Hits left @10: 0.6376316823899371
2022-10-16 02:09:03.156256 (INFO): Hits right @10: 0.41136399371069184
2022-10-16 02:09:03.158767 (INFO): Hits @10: 0.5244978380503145
2022-10-16 02:09:03.160944 (INFO): Mean rank left: 145.39293435534591
2022-10-16 02:09:03.162154 (INFO): Mean rank right: 271.6278989779874
2022-10-16 02:09:03.164338 (INFO): Mean rank: 208.51041666666666
2022-10-16 02:09:03.165595 (INFO): Mean reciprocal rank left: 0.4395735660010316
2022-10-16 02:09:03.166814 (INFO): Mean reciprocal rank right: 0.23495350803535349
2022-10-16 02:09:03.169165 (INFO): Mean reciprocal rank: 0.33726353701819254
```

结果如下：

模型	Hits@10	Hits@3	Hits@1	MRR
ConvTransE	0.514	0.368	0.245	0.334
SACN	0.524	0.387	0.253	0.337

论文中原数据如下：

模型	Hits@10	Hits@3	Hits@1	MRR
ConvTransE	0.51	0.37	0.24	0.33
SACN	0.54	0.39	0.26	0.35

- 数据基本一致，可以看到在 $epoch$ 数量为300时，两模型均接近收敛。因为 $SACN$ 后续仍有小幅增长（查看关于收敛性分析的章节），因此和论文中数值有小幅出入。
- 因为时间成本及经济成本等原因，对其他模型和其他数据集暂时不予复现。

3.3 模型算法描述

3.2.1 GCN回顾

GCN 的定义的核心思想为：每个节点依次作为聚合的中心节点，对于每个中心节点，聚合邻居节点的本层特征来作为中心节点下一层特征的表示。在 GCN 中前向迭代公式为：

$$h_i^{l+1} = \sigma(\sum_{j \in N_i} g(h_i^l, h_j^l))$$

其中，N_i表示 i 及 i 的邻居节点的集合。h^l 为该中心节点在第 l 层的向量表示。$g(par1, par2)$ 表示信息传递函数，定义为：

$$g(h_i^l, h_j^l) = h_j^l W^l$$
$$h_j^l \in R^{F^l}, W^l \in R^{F^l \times F^{l+1}}$$

上述操作将线性变换后邻居节点的向量传递到中心节点，实现 GCN 层中的聚合操作。 GCN 层堆叠次数越多，中心节点聚合的邻居节点的范围越广。

3.2.2 WGCN改进

$WGCN$ 与 GCN 的主要区别在于，在上述聚合过程中对不同的关系以不同的权重，定义权重为 α_t ，$1 \leq t \leq T$,其中 T 为关系总数，α_t 为可学参数。因此在 $WGCN$ 中前向迭代公式为：

$$h_i^{l+1} = \sigma(\sum_{j \in N_i} \alpha_t^l g(h_i^l, h_j^l))$$

其中信息传递参数与上述一致。并且此处可以将中心节点和邻居节点进行分离，将上述公式写作：

$$h_i^{l+1} = \sigma(\sum_{j \in N_i} \alpha_t^l h_j^l W^l + h_i^l W^l)$$

将其转换为矩阵形式：

$$A^l = \sum_{t=1}^{T} (\alpha_t^l A_t) + I$$

上述表达式中，A_t 表示第 t 个关系构成的 $0-1$ 邻接矩阵，0表示没有直连边，1表示有直连边。因此，可以转为类似原始的 GCN 递推公式:

$$H^{l+1} = \sigma(A^l H^l W^l)$$

根据上述的转换可以将多关系图转为多个具有不同强弱关系的单关系图。如下图所示：

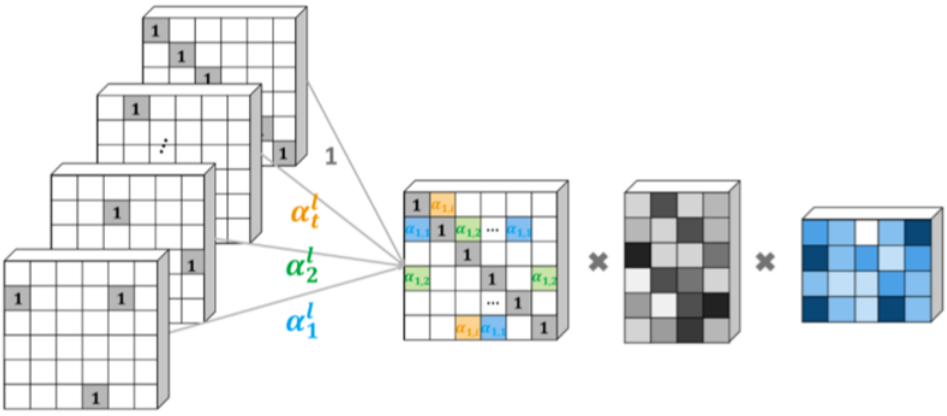


Figure 2: A weighted graph convolutional network (*WGCN*) for entity embedding.

3.2.3 属性节点

因为每个节点的属性数量少，而且各不相同，因此，属性向量十分稀疏。在本工作中，将节点的属性同样作为图的节点，这样相同属性的节点可以共享信息。

3.2.4 Conv-TransE

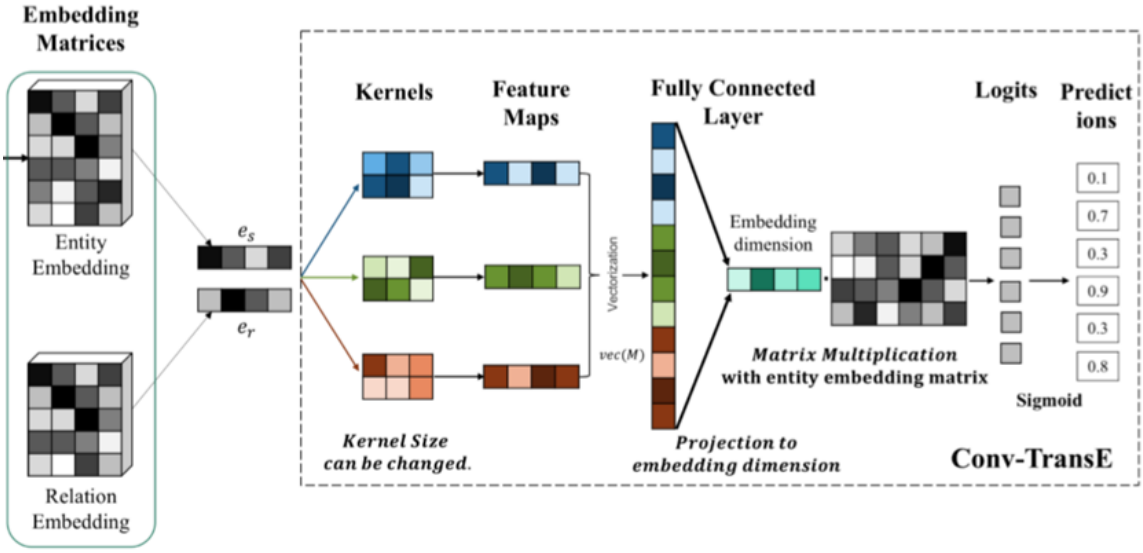
$Conv-TransE$ 类似于 $ConvE$ ，通过对 (s,r) 和 o 进行相似度打分以预测关系是否成立。不同之处在于此处省略了 $ConvE$ 的 $reshape$ 操作。

以下是知识图谱经典的打分函数。

Table 1: Scoring function $\psi(e_s, e_o)$. Here $\bar{e_s}$ and $\bar{e_r}$ denote a 2D reshaping of e_s and e_r.

Model	Scoring Function $\psi(e_s, e_o)$
TransE	$\|\|e_s + e_r - e_o\|\|_p$
DistMult	$< e_s, e_r, e_o >$
ComplEx	$< e_s, e_r, e_o >$
ConvE	$f(vec(f(concat(\bar{e_s}, \bar{e_r}) * \omega))W)e_o$
ConvKB	$concat(g([e_s, e_r, e_o] * \omega))\beta$
SACN	$f(vec(\mathbf{M}(e_s, e_r))W)e_o$

$Conv-TransE$ 中总体流程如图所示：



步骤如下：

1. 将 $WGCN$ 得到的实体 s 的嵌入式向量表示和预训练的关系 r 的嵌入式向量表示进行连接操作，转化成一个 $2*n$ 维的矩阵。
2. 对上述矩阵进行卷积操作，通过多个相同尺寸的卷积核，得到图中所示的特征图。
3. 将特征图拼接形成一个向量，通过全连接层进行降维。
4. 将上述得到的向量与 $WGCN$ 生成的所有向量分别进行点积操作，计算 (s,r) 和所有待选的 o 的相似度。
5. 相似度通过 $Sigmoid$ 函数后，取相似度最高的作为预测的实体 o 。

公式补充说明：

- 整个网络的损失可以定义为 (s,r) 和待选的 o 构成的三元组是否成立的二分类交叉熵。公式为：

$$\mathcal{L}(p,t)=-\frac{1}{N}\sum_i(t_i\cdot log(p_i)+(1-t_i)\cdot log(1-p_i))$$

- $decoder$ 中的卷积公式如下：

$$m_c(e_s,e_r,n)=\sum_{r=0}^{K-1}\omega_c(\tau,0)\hat{e_s}(n+\tau)+\omega(\tau,1)\hat{e_r}(n+\tau)$$

  其中，ω_c 是第 c 个内核的参数，K是内核大小。

- 打分函数的公式为:

$$\psi(e_s,e_o)=f(vec(M(e_s,e_r))W)e_o$$

  其中，$M(var1,var2)$ 表示卷积操作，$vec(var1)$ 表示拉直操作，$f(var1)$ 表示激活函数。

- 将打分函数通过 $Sigmoid$ 函数后，得到 (s,r) 和待选的 o 构成的三元组成立的概率。公式为：

$$p(e_s,e_r,e_o)=\sigma(\psi(e_s,e_o))$$

四、评价指标及其计算公式

4.1 计算公式

本公式为对 **评估方案** 中的 $rank_i$ 函数进行说明。

本过程在算法描述中已有涉及。首先通过打分函数和 $Sigmoid$ 函数进行计算(s,r) 和 o 的概率：

$$p(e_s,e_r,e_o)=\sigma(\psi(e_s,e_o))$$

对上述概率进行排序，概率最高的即为预测的结果。根据测试集中的正确结果即可计算出其正确率。根据上述打分函数，也可以知道排序的结果（$rank_i$）。

4.2 评估方案

4.2.1 Hits@n

该指标是指在链接预测中排名小于 n 的三元组的平均占比。具体的计算公式为：

$$HITS@n=\frac{1}{|S|}\sum_{i=1}^{|S|}\mathrm{II}(rank_i\le n)$$

其中, $\mathrm{II}()$ 表示若预测成功为1，反之为0。S 是三元组集合，$|S|$ 是三元组集合个数。$rank_i$ 指第 i 个三元组的链接预测排名。

一般而言，n 取1、3、10。本实验中亦选择1、3、10 作为指标。

4.2.2 MRR

实验使用正确实体的比例排名前1、3、10的平均倒数排名 (MRR) 作为度量值。同时，由于知识图中存在一些损坏的三元组，因此在排名之前首先过滤掉所有无效的三元组。具体计算公式如下：

$$MRR=\frac{1}{|S|}\sum_{i=1}^{|S|}\frac{1}{rank_i}=\frac{1}{|S|}(\frac{1}{rank_1}+\frac{1}{rank_2}+\ldots\frac{1}{rank_{|S|}})$$

其中，上述的参数在 $Hits@n$ 中已进行说明。

五、对比方法及引用出处

5.1 对比方法

$SACN$ 是在 $Conv-TransE$ 的基础上衍生出来的，因此首先与其进行对比。同时，这里使用更多的知识图谱嵌入工具与 $SACN$ 进行对比。对比方法为利用知识图谱和 $SACN$ 对同一知识图谱进行训练以及链接预测工作，计算其准确率 $(Hits@n)$ 以及 MRR。

下图为对比结果：

Table 3: Link prediction for FB15k-237, WN18RR and FB15k-237-Attr datasets.

	FB15k-237				**WN18RR**			
	Hits				Hits			
Model	@10	@3	@1	MRR	@10	@3	@1	MRR
DistMult (Yang et al. 2014)	0.42	0.26	0.16	0.24	0.49	0.44	0.39	0.43
ComplEx (Trouillon et al. 2016)	0.43	0.28	0.16	0.25	0.51	0.46	0.41	0.44
R-GCN (Schlichtkrull et al. 2018)	0.42	0.26	0.15	0.25	—	—	—	—
ConvE (Dettmers et al. 2017)	0.49	0.35	0.24	0.32	0.48	0.43	0.39	0.46
Conv-TransE	0.51	0.37	0.24	0.33	0.52	0.47	0.43	0.46
SACN	0.54	0.39	0.26	0.35	**0.54**	**0.48**	**0.43**	**0.47**
SACN using **FB15k-237-Attr**	**0.55**	**0.40**	**0.27**	**0.36**	—	—	—	—
Performance Improvement	**12.2%**	**14.3%**	**12.5%**	**12.5%**	**12.5%**	**11.6%**	**10.3%**	**2.2%**

详细分析见结果部分。

5.2 引用出处

以下是几种与 $SACN$ 对比模型的出处。

DistMult：Embedding Entities and Relations for Learning and Inference in Knowledge Bases

ComplEx：Complex Embeddings for Simple Link Prediction

R-GCN：Modeling Relational Data with Graph Convolutional Networks

ConvE：Convolutional 2D Knowledge Graph Embeddings

六、结果

6.1 链接预测

6.1.1 结果展示

在链接预测任务的实验中，结果如图所示。

Table 3: Link prediction for FB15k-237, WN18RR and FB15k-237-Attr datasets.

Model	FB15k-237				WN18RR			
	Hits				Hits			
	@10	@3	@1	MRR	@10	@3	@1	MRR
DistMult (Yang et al. 2014)	0.42	0.26	0.16	0.24	0.49	0.44	0.39	0.43
ComplEx (Trouillon et al. 2016)	0.43	0.28	0.16	0.25	0.51	0.46	0.41	0.44
R-GCN (Schlichtkrull et al. 2018)	0.42	0.26	0.15	0.25	—	—	—	—
ConvE (Dettmers et al. 2017)	0.49	0.35	0.24	0.32	0.48	0.43	0.39	0.46
Conv-TransE	0.51	0.37	0.24	0.33	0.52	0.47	0.43	0.46
SACN	0.54	0.39	0.26	0.35	**0.54**	**0.48**	**0.43**	**0.47**
SACN using **FB15k-237-Attr**	**0.55**	**0.40**	**0.27**	**0.36**	—	—	—	—
Performance Improvement	**12.2%**	**14.3%**	**12.5%**	**12.5%**	**12.5%**	**11.6%**	**10.3%**	**2.2%**

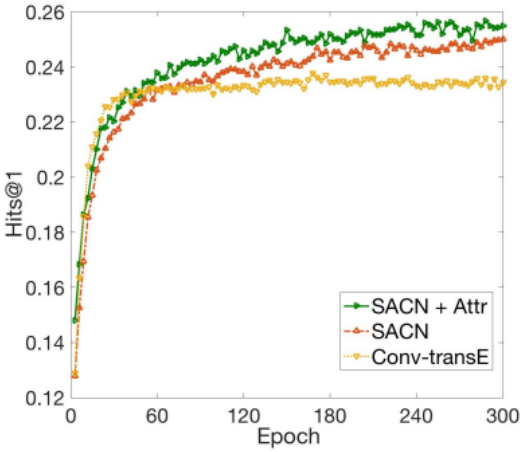
6.1.2 结论

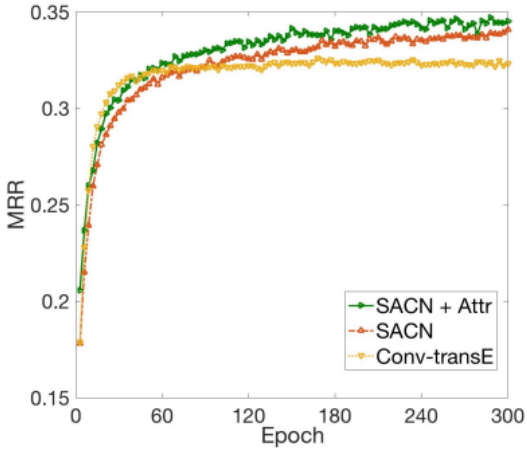
1. 若不考虑 $SACN$ 。
   - 在 $FB15k-237$ 数据集中，$Conv-TransE$ 模型的$Hits@10$ 比 $ConvE$ 提高了4.1%， $Hits@3$ 提高了 5.7% 。
   - 在 $WN18RR$ 数据集中，$Conv-TransE$ 模型的$Hits@10$ 比 $ConvE$提高了8.3%， $Hits@3$ 提高了 9.3% 。
   - 得出结论：使用神经网络的 $Conv-TransE$ 保持了实体和关系之间的平移特性，展示出更好的性能。
2. 若考虑 $SACN$。
   - 在 $FB15k-237$ 数据集中，$SACN$ 模型的$Hits@10$ 比 $ConvE$提高了10.2%， $Hits@3$ 提高了 11.4%， $Hits@1$提高了 8.3% ， MRR 的值提高了 9.4%。
   - 在 $WN18RR$ 数据集中，$SACN$ 模型的$Hits@10$ 比 $ConvE$提高了12.5%， $Hits@3$ 提高了 11.6%， $Hits@1$提高了 10.3% ， MRR 的值提高了 2.2%。
3. 若考虑 $SACN$，并且将节点属性添加到 $SACN$ 模型中。
   - 在 $FB15k-237-Attr$ 数据集中，$SACN$ 模型的$Hits@10$ 比 $ConvE$提高了12.2%， $Hits@3$ 提高了 14.3%， $Hits@1$提高了 12.5% ， MRR 的值提高了 12.5%。
   - 在 $FB15k-237-Attr$ 数据集中，$SACN$ 模型的$Hits@10$ 比没有使用属性的 $SACN$提高了 1.9%， $Hits@3$ 提高了 2.6%， $Hits@1$提高了 3.8% ， MRR 的值提高了 2.9%。

6.2 收敛分析

6.2.1 结果展示

若使用 $FB15k-237-Attr$, $SACN$ 、$Conv-TransE$、$SACN+Attr$ 三者的 $Hits@1$ 和 MRR 收敛性如下：





6.2.2 结论

1. $Conv-TransE$ 在 120 个 $Epoch$ 后性能达到最优、停止增长，但 $SACN$ 仍保持增长。
2. 当使用 $FB15k-237-Attr$ 数据集，$SACN+Attr$ 的模型效果总是比 $SACN$ 高。

6.3 卷积核大小的选择

6.3.1 结果展示

针对参数的调节部分，这里尝试了不同大小的卷积核，此处给出针对不同数据集的卷积核大小的结果。

Table 4: Kernel size analysis for FB15k-237 and FB15k-237-Attr datasets. "*SACN+Attr*" means the *SACN* using FB15k-237-Attr dataset.

Model	Kernel Size	FB15k-237			
		Hits			
		@10	@3	@1	MRR
Conv-TransE	2 × 1	0.504	0.357	0.234	0.324
Conv-TransE	2 × 3	0.513	0.365	0.240	0.331
Conv-TransE	2 × 5	0.512	0.361	0.239	0.329
SACN	2 × 1	0.527	0.379	0.255	0.345
SACN	2 × 3	0.536	0.384	0.260	0.351
SACN	2 × 5	0.536	0.385	0.261	0.352
SACN+Attr	2 × 1	0.535	0.384	0.260	0.351
SACN+Attr	2 × 3	0.543	0.394	0.268	0.360
SACN+Attr	2 × 5	**0.547**	**0.396**	**0.268**	**0.360**

6.3.2 结论

1. 如果增加核大小，$Hits@1$、$Hits@3$、$Hits@10$、MRR 指标均能增加。
2. 最佳内核大小取决于具体任务。

6.4 度对性能的影响

6.4.1 结果展示

这里比较了 $Conv-TransE$ 和 $SACN$ 在不同的度下的性能表现。

Table 5: Node indegree study using FB15k-237 dataset.

	Conv-TransE		**SACN**	
	Average Hits		Average Hits	
Indegree Scope	@10	@3	@10	@3
[0,100]	0.192	0.125	0.195	0.134
[100,200]	0.441	0.245	0.441	0.253
[200,300]	0.696	0.446	0.705	0.429
[300,400]	0.829	0.558	0.806	0.577
[400,500]	0.894	0.661	0.868	0.663
[500,1000]	0.918	0.767	0.891	0.695
[1000, maximum]	0.992	0.941	0.981	0.922

6.4.2 结论

- 在度较低的节点中，$SACN$ 的表现优于 $Conv-TransE$，因为邻居节点直接能共享更多的信息。
- 在度较高的节点中，$SACN$ 的表现不如 $Conv-TransE$，因为较多的邻居节点使得较为重要的邻居的信息被过度稀释，因此不如 $Conv-TransE$。