在图数据上做机器学习，应该从哪个点切入？

AI科技大本营 1 week ago

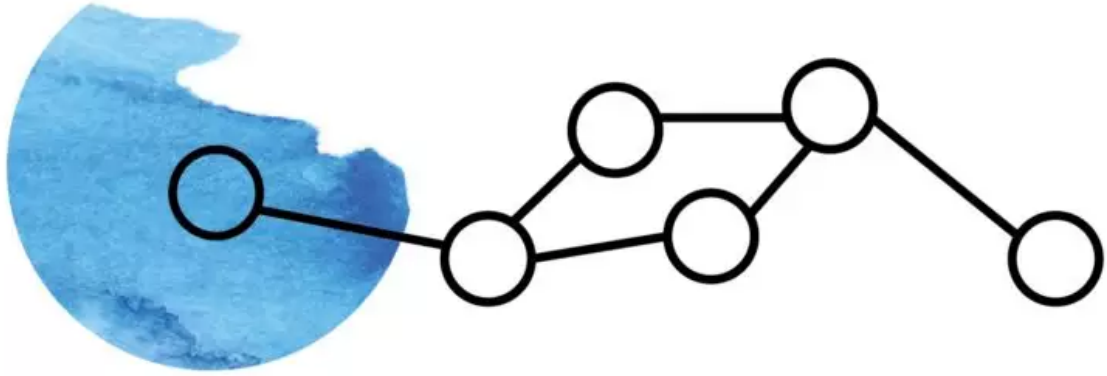
Reposted from Official Account AI公园，Author ronghuaiyang

作者 | David Mack
编译 | ronghuaiyang
来源 | AI公园（ID:AI_Paradise）

> 【导读】很多公司和机构都在使用图数据，想在图上做机器学习但不知从哪里开始做，希望这篇文章给大家一点启发。

自从我们在伦敦互联数据中心(Connected Data London)的演讲以来，我已经与许多拥有图数据的研究团队进行了交谈，他们希望对图进行机器学习，但不确定从哪里开始。

在本文中，我将分享一些资源和方法，帮助你开始学习图机器学习。



什么是图数据？

通过与研究团队的交谈，我们可以清楚地看到图数据的广泛性和普适性——从疾病检测、遗传学和医疗保健到银行业和工程学，图正成为解决难题的一种强大的分析范式。

简单地说，图是节点(例如人)和节点之间关系的集合(例如Fatima是Jacob的朋友)。通常，这些节点具有某些特征(例如Fatima是23岁)。

将这些数据存储在数据库中是很常见的。一个流行的数据库是Neo4j，用他们自己的话说，“世界领先的图数据库，具有本地图存储和处理功能。”
Neo4j允许你使用Cypher查询数据库，这相当于SQL。在上面的例子中，我们可以看到Fatima的朋友列表如下：

```
MATCH (n1)-[:IS_FRIEND_OF]-(n2)WHERE n1.name = "Fatima"RETURN n2.name
```

图是一种非常灵活和强大的表示数据的方法。传统的关系数据库具有固定的模式，因此很难

> 术语“关系”和“边”在本文中可互换使用。Neo4j使用前者，很多图论使用后者。

为什么要在图数据上使用机器学习？

首先，为什么要使用机器学习？关于这个问题的一篇很棒的文章是Benedict Evans的“Ways to think about machine Learning”：https://www.ben-evans.com/benedictevans/2018/06/22/ways-to-think-about-machine-learning-8nefy，涵盖了公司开始思考和实际使用ML的方法。

从Ben的论点中提炼出图机器学习，它有两种主要的用处：

机器学习可以自动化一些功能，这些功能对于人类来说很容易做到，但是对于计算机来说却很难描述

真实世界的数据是有噪声的，有许多复杂的子结构。像“在这幅图中勾勒出人物轮廓”这样的任务对人类来说很容易，但很难转化为单个算法。
深度学习允许我们将大量的数据转换为某种函数，从而实现特定任务的自动化。
对于图来说，这是双重的事实——由于开放式关系结构，图与图像或向量之间的差异可能呈指数级增长。使用图机器学习，我们可以创建函数来发现重复出现的模式。

机器学习可以在人类无法进行的尺度内转换信息

计算机的双刃剑是，它们会完全按照我们告诉它们的去做——不多也不少(偶尔的bug除外!)

这意味着他们将执行我们的确切指示，没有偏离也没有即兴创作。不管我们让它们跑多久，它们都会重复执行。

因此，计算机可以处理人类无法处理的数据量(由于所需的时间或精力)。这使得新的分析成为可能，比如分析数十亿个交易网站的指纹欺诈。

什么是图机器学习？

我们的定义就是“将机器学习应用于图数据”。这是广泛和包容的。在本文中，我将侧重于神经网络和深度学习方法，因为它们是我们自己的重点，但是在可能的情况下，我将包括到其他方法。

在本文中，我不打算讨论“传统的”图分析——这是众所周知的算法技术，如PageRank、社群识别、最短路径等。这些功能非常强大，由于它们的性质很好理解，并且在开源库中有大量的实现，所以应该将它们视为一个调用的接口。

使用图机器学习的挑战是什么？

虽然图机器学习是一个充满希望的令人兴奋的领域，但它仍然是一项新兴技术。
在机器学习的主流领域，出现了许多广泛应用的技术(例如，使用ResNet处理图像或BERT处理文本)，并使开发人员可以使用这些技术(例如TensorFlow、PyTorch、FastAI)。然而，没有同样简单、通用的技术，也没有任何流行的机器学习库支持图数据。

类似地，像Neo4j这样的图形数据库并不提供对其数据运行机器学习算法的方法(尽管Neo4j正在考虑如何使这成为可能!)

深度学习库缺乏图支持的原因之一在数据结构的灵活性(例如，任何节点可以拥有任意数量的其他节点)，之间的关系不适合固定的计算图，在深度学习库和GPU厂商中，流行使用固定大小的张量。

更简单地说，很难将稀疏图表示为矩阵并对其进行操作。不是不可能，但肯定比处理向量、文本和图像更困难。

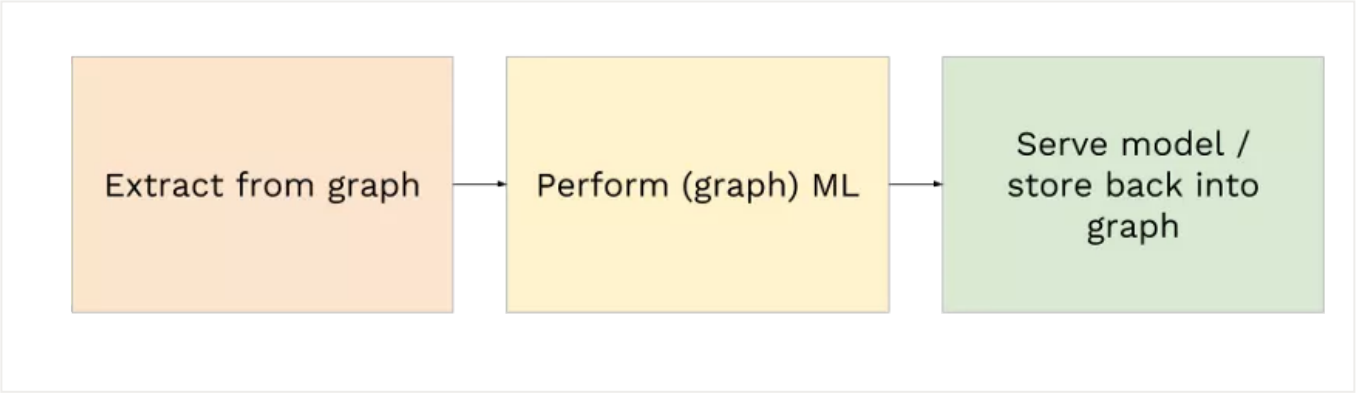
然而，尽管如此，人们对图机器学习的兴趣还是激增。我个人预测，这个领域将成为主流，并成为我们分析许多行业数据的基础工具。
注意，与任何其他继续学习技术一样，大多数图机器学习需要大量的训练数据。

图机器学习系统是个什么样的东西？

虽然这个问题的答案可能会随着任务和数据集的不同而有很大的不同，但是概括一下应该期望遇到的情况是很有帮助的。

几乎可以肯定的是，你将自己编写这个系统，因为图机器学习的高级工具还不存在。你很可能将使用Python和类似TensorFlow或PyTorch的机器学习库来构建系统。根据你的规模，你可能正

在一台机器上训练模型，或者使用分布式集群(有趣的是，许多图学习算法天然适合分布式)。



你可能首先需要从图中提取数据—可能存储在CSV文件、Neo4j之类的图数据库或其他格式中。

然后将这些数据输入机器学习库。在我目前的工作中(涉及数百万个小图)，我将每个图预编译成一个TFRecord，用特征向量存储节点、关系和邻接矩阵。所有节点属性和文本都使用公共字典进行标记。

这适用于小图，但对于较大的图，你需要某种方案将图划分为较小的训练样本(例如，你可以在小块上进行训练，或者在单个节点-边缘-节点三元组上进行训练)。
注意，有些方法在数据到达机器学习库之前将其制表。Node2Vec就是一个很好的例子，它使用随机游走将每个节点转换成一个向量。然后将这些向量作为列表输入给机器学习模型。

一旦数据被接收，实际的建模和学习就开始了。这里有很多种可能的方法。
最后，需要以某种方式使用模型或提供服务。在某些情况下，模型会计算一个新的节点/边/图属性，并将其添加到原始的数据存储中。

在其他情况下，会生成一个用于在线预测的模型。在此设置中，需要建立一个系统来，给模型输入满足要求的图数据，然后进行预测(可能需要再一次从图数据库中取数据)，最后，得到的预测可以送到用户手里，或给到后续的系统。

让我们在图上来做机器学习

好了，让我们来看看你可以采取的一些方法来对图进行机器学习。

我将在这里概述这些方法，指出它们的一些优缺点。为了时间和空间，我不得不在这里牺牲一些细节。

尽管这是一个年轻的领域，研究人员已经提出了一系列令人眼花缭乱的方法和变化。虽然我已经试着涵盖了本文的主要领域，但遗憾的是，没有办法使这个列表完全详尽。

你要完成什么任务？

图机器学习有各种各样的起点和总体方法。因此，通过思考要完成的一般任务是什么来缩小这些起点和方法的范围是很有帮助的。

与任何学习系统一样，缩小你想要达到的目标范围，对你的成功和开发工作都有很大的帮助。通过提出一个最小的、清晰的目标，你的模型和数据集可以简化为更容易处理的东西。

最后一点，图数据库非常强大，它鼓励我们朝着宏大的"全能"目标前进：由于数据库几乎可以代表任何东西，因此试图构建通用的智能是很有诱惑力的。

我们将讨论的任务类型：

1. 预测两个节点之间是否存在关系
2. 节点、边缘和整个图的评分和分类

本文旨在作为你自己研究的起点。与任何数据科学一样，方法也需要根据你的具体情况进行调整。由于许多图机器学习还处于早期研究阶段，所以在找到一种有效的方法之前，你应该尝试许多方法。

基本方法

在开始构建图机器学习系统之前(可能需要在基础设施方面进行大量投资)，重要的是考虑是否可以使用更简单的方法。

有几种方法可以简化这个问题：

- 你能把你的数据制表吗？是否可以使用传统的ML方法(例如线性回归、前馈网络)？
- 你可以过滤数据集让数据集变得更小吗(例如删除某些节点)？

- 你能把这个图分成子图并把它们当作表格吗？
- 是否可以使用传统的图度量(例如PageRank)，并可能使用传统机器学习进行扩充(例如，对计算出的节点属性应用线性回归对节点进行分类)

在下面的部分中，我将在特别适用的地方回顾其中的一些方法。

通用图机器学习方法

一些图机器学习方法可以用于多个任务。我这里包括了它们的完整描述。在后面的部分中，我将引用这个部分并突出显示一些特定于任务的细节。
同样，不可能公正地对待这些大的工作领域，我们在这里能做的最好的事情就是给你提供进一步探索的指导。

节点嵌入

节点嵌入是图机器学习的早期发展之一，由于其简单、健壮性和计算效率，一直很受欢迎。

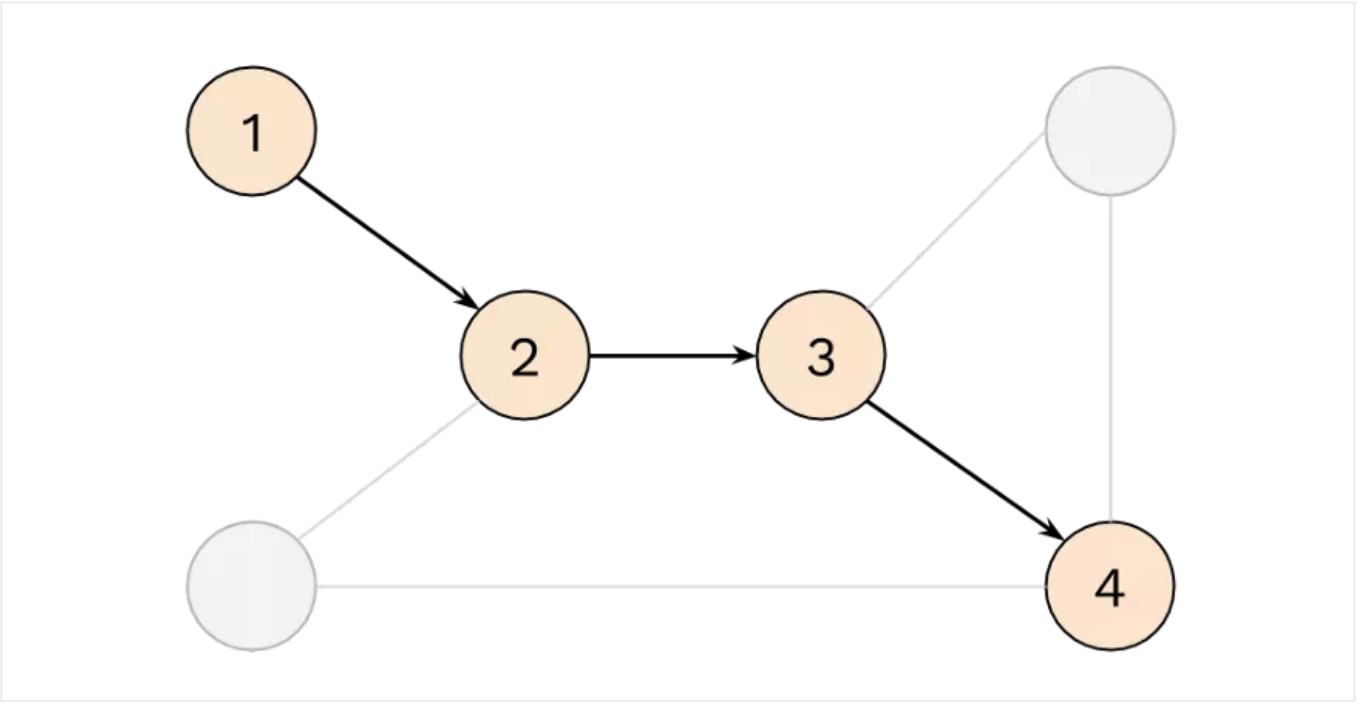
节点嵌入仅仅意味着为图中的每个节点计算一个向量。计算向量是为了得到有用的属性，例如任意两个节点的嵌入的点积可以告诉你它们是否来自同一个社区。
通过这种方式，嵌入将图数据简化为更易于管理的东西：向量。

节点嵌入通常是通过将许多图结构合并在一起来计算的(稍后将详细介绍)。
他们的权衡是必须丢弃信息。一个固定长度的向量很少能表示一个节点周围的所有的图结构。它可以合并节点和关系属性，也可以不合并。

但是，只要稍加创新，就可以将节点嵌入与其他图机器学习方法结合使用。在这个设置中，嵌入成为一个节点属性，可以用作其他技术的助推器，这些技术可能不会像嵌入生成那样深入到图结构中。

这里我将重点介绍一些主要的嵌入方法。

随机游走



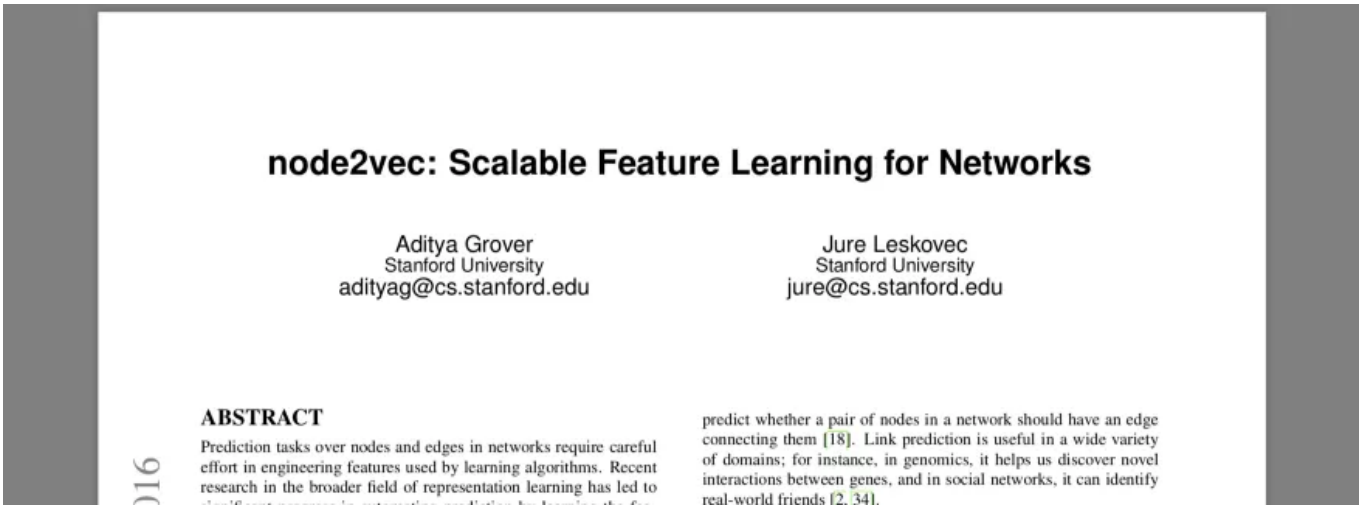
随机游走是一种功能强大且简单的图分析技术，有悠久的数学理论作后盾。
随机游走是从图中的一个节点开始，随机选择一条边，然后遍历它。然后重复这个过程，直到产生足够长的路径。

随机游走的天才之处在于它将一个多维不规则的东西(图)转化为一个简单的矩阵(固定长度路径的列表，每个路径由它的节点组成)。

在足够大的体量下，理论上有可能从随机游走重构出基本的图结构。而随机游走发挥了机器学习的巨大优势：从大量数据中学习。

利用随机游走计算节点嵌入的方法有很多。在接下来的文章中，我将重点介绍一些主要的方法。

Node2Vec



node2vec: Scalable Feature Learning for Networks

Aditya Grover
Stanford University
adityag@cs.stanford.edu

Jure Leskovec
Stanford University
jure@cs.stanford.edu

ABSTRACT

Prediction tasks over nodes and edges in networks require careful effort in engineering features used by learning algorithms. Recent research in the broader field of representation learning has led to

predict whether a pair of nodes in a network should have an edge connecting them [18]. Link prediction is useful in a wide variety of domains; for instance, in genomics, it helps us discover novel interactions between genes, and in social networks, it can identify real-world friends [2, 34].

Node2Vec是一种使用随机游走的流行且相当通用的嵌入技术。

将这些随机游走转化为嵌入的方法有一个聪明的优化目标。首先为每个节点分配一个随机嵌入(例如长度为N的高斯向量)，然后对每个遍历中的每一对源-邻居节点，通过调整它们的嵌入，使它们的嵌入的点积最大。最后，我们同时最小化随机节点对的点积。这样做的结果是，我们学习了一组嵌入，它们趋向于为相同游走中节点提供高点积，例如在相同的社区/结构中。

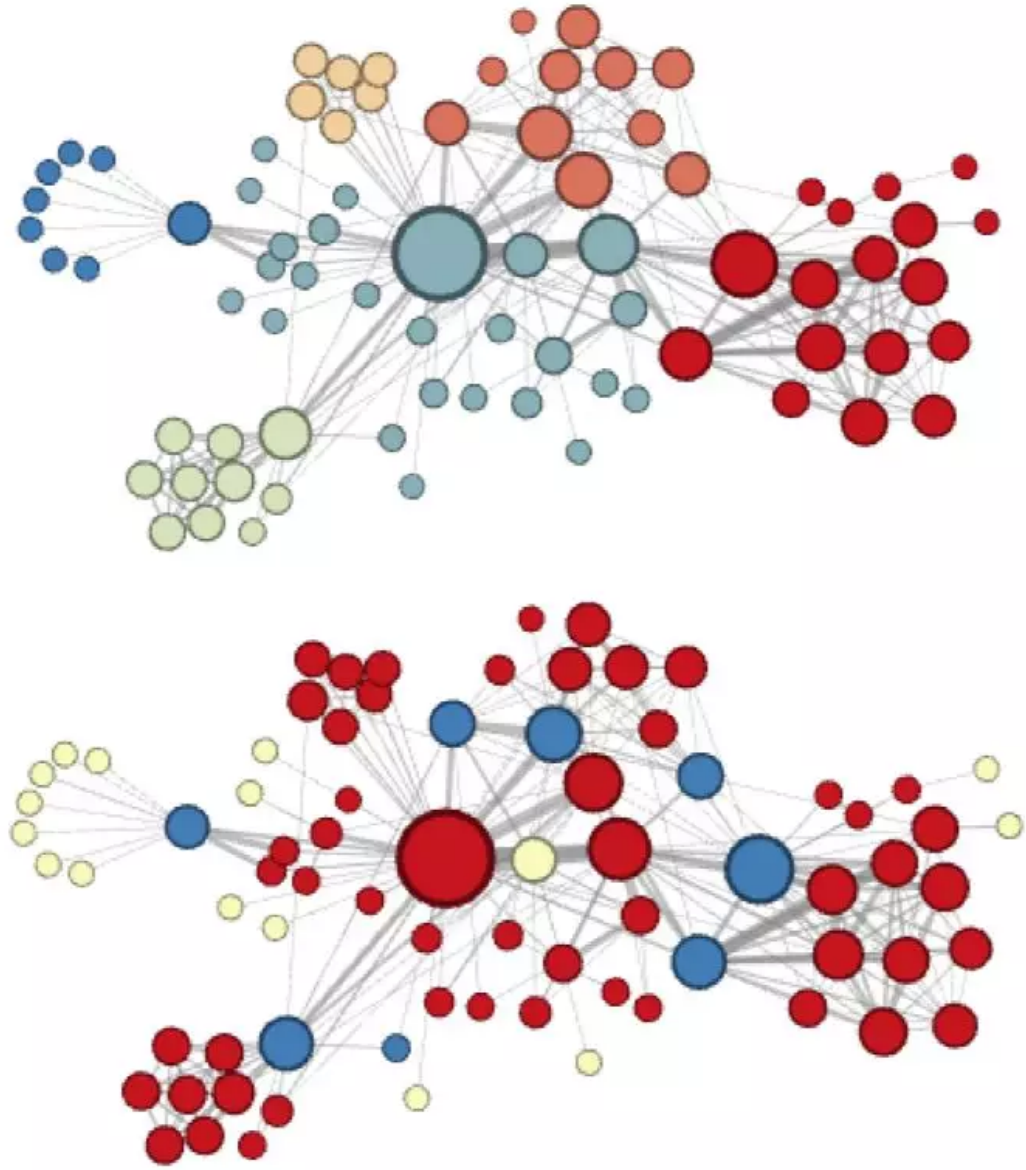


Figure 3: Complementary visualizations of Les Misérables coappearance network generated by *node2vec* with label colors reflecting homophily (top) and structural equivalence (bottom).

Node2Vec的最后一点是，它有参数来形成随机游动。使用“in -out”超参数，你可以确定游走的优先级是集中在小的局部区域(例如，这些节点是否在相同的小社区中?)，还是游走在图的广泛分布中(例如，这些节点是否在相同类型的结构中?)

Node2Vec扩展

Node2Vec的优点是简单，但这也是它的缺点。标准算法不包含节点属性或边缘属性以及其他需要的信息。

然而，扩展Node2Vec以包含更多信息是非常简单的。简单地改变损失函数。例如：

- 不要在节点嵌入之间做点积，尝试一个不同的/可学习的函数
- 不要只使用节点嵌入，还要合并它们的属性

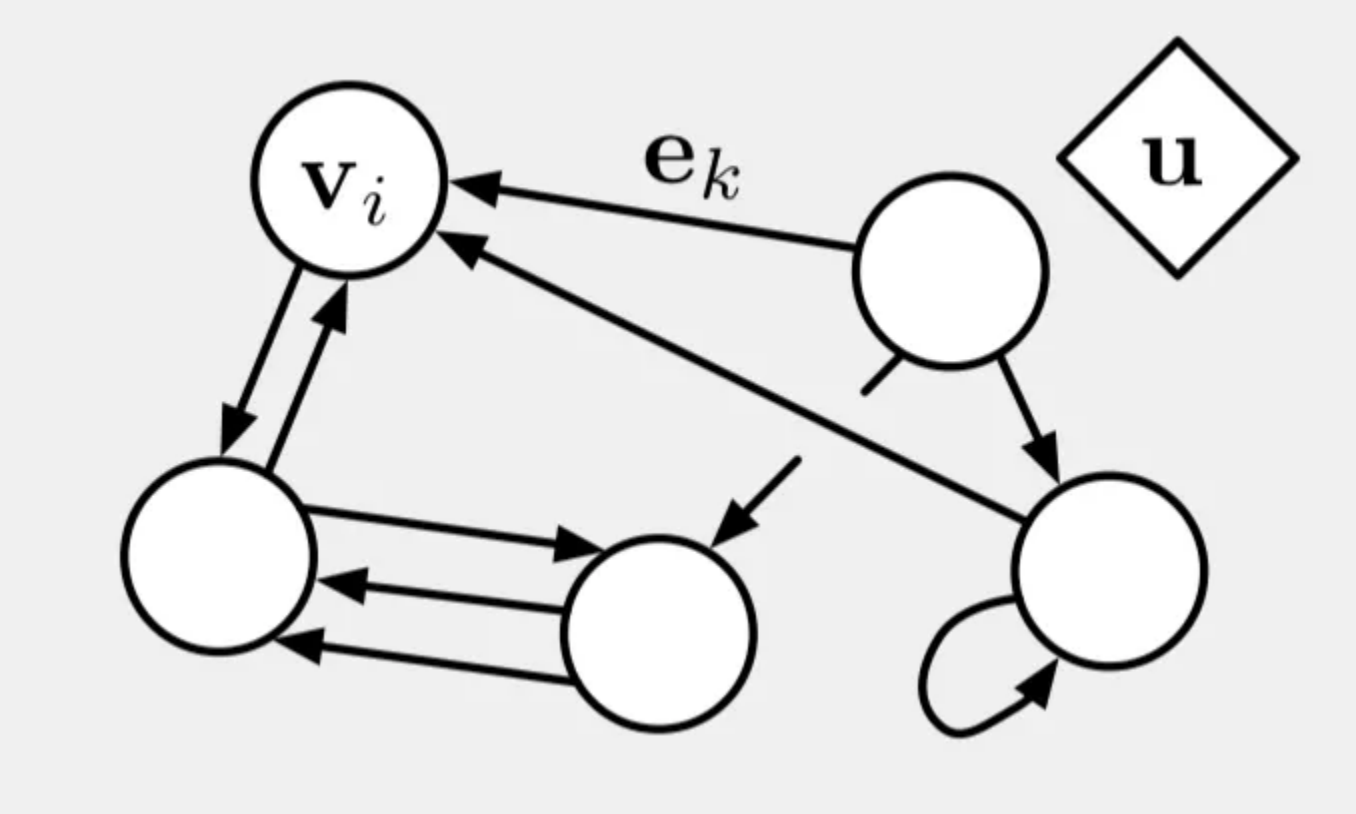
使用随机游走做协同过滤

使用随机游走的一个非常简单的例子是解决协同过滤问题，例如，给定用户对产品的评论，用户还会喜欢其他哪些产品？

这大致遵循与node2vec相同的方案，尽管已经进一步简化。你可以在我们的文章中看到整个实现和解释：https://medium.com/octavian-ai/review-predicing-with-neo4j-and-tensorflow-1cd33996632a。

图网络（也就是图卷积网络）

图网络是图机器学习中一个丰富而重要的领域，其基本前提是将神经网络嵌入到图结构本身中：



通常，这包括为每个节点存储其状态，并使用邻接矩阵将这些状态传播到该节点的邻居。

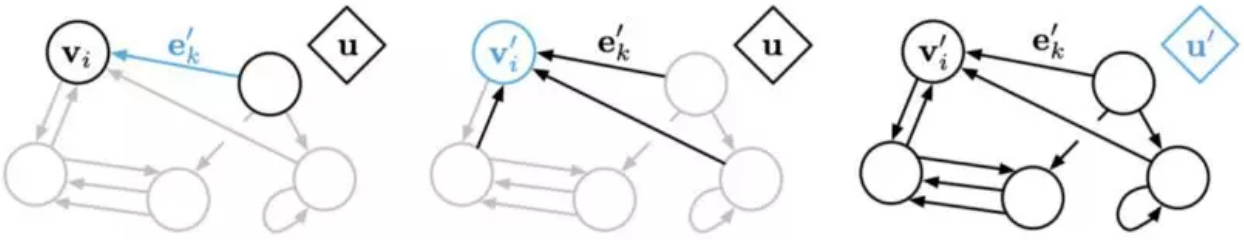
有一篇很好的综述性论文，来自DeepMind的“Relational inductive biases, deep learning, and graph networks”，它既调查了这个子领域的历史，也提出了一种统一的方法来比较不同的

图网络和一般的神经网络。

在上面的文章中，图网络被认为是一组函数的集合，用于传播状态和跨节点、边缘和整个图的状态聚合。通过这种方式，比较了文献中许多不同的架构。下面是这些功能的摘录：

A GN block contains three "update" functions, ϕ, and three "aggregation" functions, ρ,

$$\begin{aligned} \mathbf{e}'_k &= \phi^e\left(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}, \mathbf{u}\right) & \bar{\mathbf{e}}'_i &= \rho^{e\to v}\left(E'_i\right) \\ \mathbf{v}'_i &= \phi^v\left(\bar{\mathbf{e}}'_i, \mathbf{v}_i, \mathbf{u}\right) & \bar{\mathbf{e}}' &= \rho^{e\to u}\left(E'\right) \\ \mathbf{u}' &= \phi^u\left(\bar{\mathbf{e}}', \bar{\mathbf{v}}', \mathbf{u}\right) & \bar{\mathbf{v}}' &= \rho^{v\to u}\left(V'\right) \end{aligned} \tag{1}$$



(a) Edge update (b) Node update (c) Global update

Figure 3: Updates in a GN block. Blue indicates the element that is being updated, and black indicates other elements which are involved in the update (note that the pre-update value of the blue element is also used in the update). See Equation 1 for details on the notation.

一个图网络有许多可能的输出：

- 节点状态
- 边缘状态
- 全图状态

然后，这些可以像嵌入一样用于分类、评分和关系预测等任务。

图网络是非常通用和强大的——他们被用于分析很多事情，自然语言，3d场景，生物学。我们最近的工作已经表明，它们可以实现许多常见的传统图算法。
预测两个节点之间是否存在关系（关联预测）

这是一项常见的任务，而且研究得很充分。基本公式为：

节点A与节点B有关系的概率p(A,R,B)是多少？

例如知识图谱的完善(例如，如果米开朗基罗是出生在托斯卡纳的画家，他是意大利人吗？)和预测蛋白质相互作用]。这既可以用来预测新的未知事物(例如，哪些药物可能有效?)，也可以用来改进现有的不完善数据(例如，这项任务属于哪个项目?)

> 关于许多方法的更多信息可以在前面的“通用图机器学习方法”一节中找到

节点嵌入和随机游走

节点嵌入(通常使用随机游动生成)经常用于连接预测。

嵌入通常是这样生成的：图中邻近的节点具有类似的嵌入张量。因此，度量(例如点积或欧氏距离)提供了连接的可能性。像Node2Vec这样的一些方法实际上直接训练嵌入连接的存在与否。

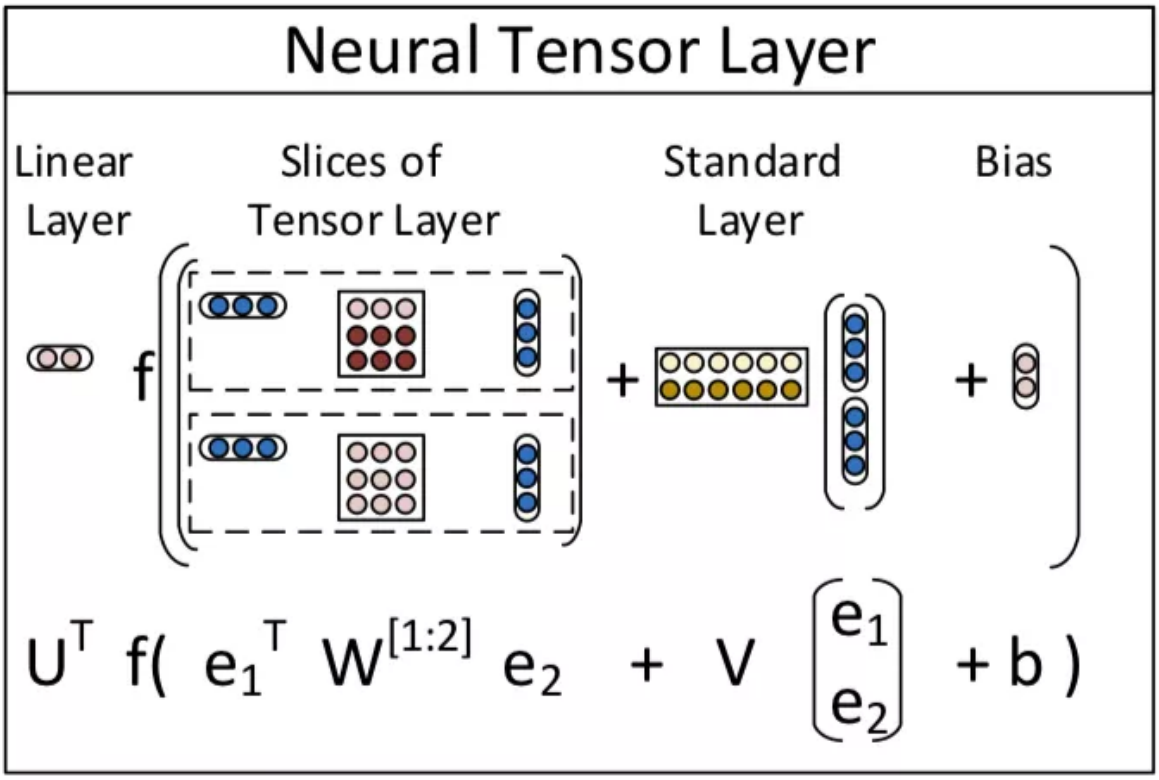
图网络可用于生成节点嵌入，用于连接预测。在这种情况下，将连接预测功能合并到网络的损失函数中。

使用图特征来做深度学习

这意味着将图数据制表，然后在其上运行传统的前馈网络。

例如，每个节点都可以用它的属性表示(连接成一个张量)。每个训练实例都有两个节点和关系类型作为特征，边缘的存在作为标签。记住要平衡label类。
当许多图结构反映在属性中时(例如，街道图和每个节点都有其GPS位置)，这种简单的方法可以很好地工作。

神经张量网络



斯坦福大学的这个有趣的方法本质上是将图形存储为张量和矩阵。“我们的模型优于之前的模型，能够对WordNet和FreeBase中不可见的关系进行分类，准确率分别为86.2%和90.0%。”

对节点，边和图打分和分类

另一个常见的任务是对图的一部分进行分类或打分。例如，试图找出蛋白质与某一特定基因的相关性。或者试图将学生按他们的关系分组。

分类表示输出一个跨越潜在标签的概率分布，得分表示输出一个标量，这个标量可能用于与其他标量进行比较。两者在概念上是相似的，分类涉及到更多的维度。

形式上，任务是定义以下函数之一，其中Output是可能的输出类分布集或可能的输出分数集：

f(n:Node)→r∈Output
g(e:Edge)→r∈Output
h(g:Graph)→r∈Output

大多数实现这一点的方法有两个步骤：

1. 对图执行一些计算，可能将其节点和边的多个元素组合成存储在节点、边和/或整个图中的状态

2. 提取、聚合并将状态转换为所需的输出

步骤1，可以使用许多不同的方法执行，我将在下面列出。

步骤2，通常使用前馈神经网络 (FFN)执行。提取和聚合要么使用手工编写的函数(例如读出特定的节点，将特定的边求和)，要么使用学习函数(例如注意力用于提取，卷积用于聚合)。

这两个步骤的选择是数据科学和实验的问题，还没有出现任何明确的“一刀切”解决方案。

节点嵌入和随机游走

节点嵌入为分类和评分提供了丰富的节点状态源。

当使用embeddings时，通常被检查的节点会让他们的embeddings通过一个小的FFN来产生想要的输出。根据用例，节点的属性也可以包含在FFN的输入中。
如果节点创建嵌入时使用随机游走(例如使用Node2Vec)他们将把本地的结构信息(例如，节点属于哪个社区，或者这个节点属于哪个超结构的一部分)这可能和分类或评分有关(例如，不同的子图的聚类)。

图网络

图网络是一种通用的方法，可以将神经网络嵌入到图中。

图形网络计算节点、边缘和图形状态(尽管根据应用程序可以省略其中一些)。
然后可以将这些状态转换为最终的输出。例如，可以通过FFN传递图的状态来创建图的总体分类。

在图网络的文献中有许多不同的例子，请参阅介绍性部分以获得对它们的简要概述。

在节点属性或子图上做传统深度学习

将问题简化为一个表格数据集，这样可以使用许多更好的研究方法(例如前馈和卷积神经网络)。

一种方法是将每个节点及其属性作为一个训练样本。这可能涉及手工生成额外的属性，相信这将有助于分类/评分。

另一种将图制成表的方法是提取固定大小的子图。在这个模型中，一个节点、它的边以及它的邻居被提取到一个固定大小的表中。固定的大小意味着可以存储最大数量的边和节点，如果表中存在更多的边和节点，则必须随机采样。此外，如果节点和边缘比固定表所能存储的少，则需要用指定的空值填充。最后，必须选择如何选择子图——一个简单的模型是提取每个节点或边缘的子图。

表格化丢弃了潜在的有价值的网络信息，但简化了工程和模型研究。

英文原文：https://medium.com/octavian-ai/how-to-get-started-with-machine-learning-on-graphs-7f0795c83763

*(*本文为AI科技大本营转载文章，转载请联系作者)*

推荐阅读

- Dropout、梯度消失/爆炸、Adam优化算法，神经网络优化算法看这一篇就够
- AI换脸软件ZAO刷屏，可我却不敢用了
- 只给测试集不给训练集，要怎么做自己的物体检测器？
- 还在抱怨pandas运行速度慢？这几个方法会颠覆你的看法
- 没有光芯片，何谈 5G 与 AI！
- 30 岁的程序员，我没有活成理想的模样，失败吗？
- 看懂“大数据”，这一篇就够了！
- 别让分析公司卖了你：一文读懂比特币的私密性及隐私保护

你点的每个“在看”，我都认真当成了喜欢

Read more