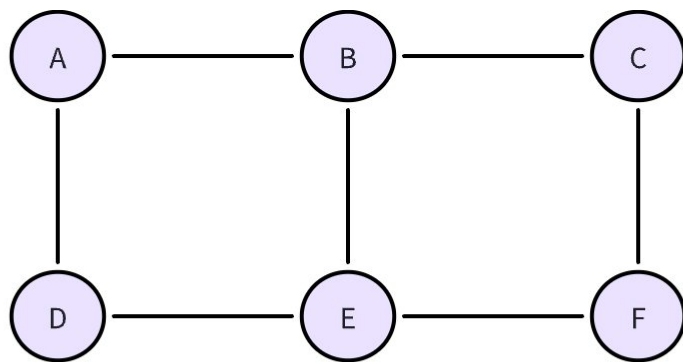


DeepWalk



1. 随机游走 (Random Walk)

我们从节点 A 开始，进行随机游走，假设游走的步数为 4，那么可能生成的一个随机游走序列为：

```
A -> B -> E -> D -> A
```

这个序列就像是一个句子，其中每个节点就是一个“单词”。

如果再从其他节点开始生成随机游走序列，比如从节点 C，可能生成的序列为：

```
C -> B -> A -> D -> E
```

通过这种方式，DeepWalk 会对图中的所有节点进行多次随机游走，生成多个类似于自然语言的“句子”

2. 使用 Word2Vec 训练节点表示

接下来，DeepWalk 会把这些节点序列（随机游走产生的“句子”）输入到 **Skip-Gram** 模型（**Word2Vec** 的一种变体）中，学习每个节点的低维向量表示。

例如，随机游走序列 `A -> B -> E -> D -> A` 中，可以使用 Skip-Gram 模型预测相邻节点的关系，假设上下文窗口为 2，模型会学到 A 与 B、D 关系密切，B 与 A、E 关系密切，依此类推。

3. 节点嵌入

训练完成后，每个节点都会得到一个低维向量表示。

节点 A 的表示可能是：

A: [0.1, 0.3, -0.2, 0.5]

节点 B 的表示可能是：

B: [-0.2, 0.4, 0.1, -0.3]

4. 应用

这些节点的嵌入向量可以用于不同的任务，例如：

- **节点分类**：根据每个节点的嵌入表示，我们可以训练分类器来预测每个节点的类别，例如在社交网络中预测用户的兴趣或职业。
- **节点相似性计算**：通过计算节点嵌入向量之间的余弦相似度，找到与某个节点最相似的其他节点。
- **社区发现**：通过嵌入向量的聚类，可以识别出图中相互紧密连接的节点群体。

总结

DeepWalk 的核心在于通过随机游走捕捉图的局部和全局结构，并使用 Word2Vec 模型学习每个节点的低维表示。这个方法有效解决了传统图分析中难以处理大规模复杂网络的问题。

Word2vec is a technique in natural language processing (NLP) for obtaining vector representations of words.

$$\vec{vec(Queen)} - \vec{vec(king)} \approx \vec{vec(Woman)} - \vec{vec(Man)}$$

