

NIPS16一篇将变分自编码器迁移到图结构数据上的论文

解决的问题

在图结构数据上如何使用变分自编码器

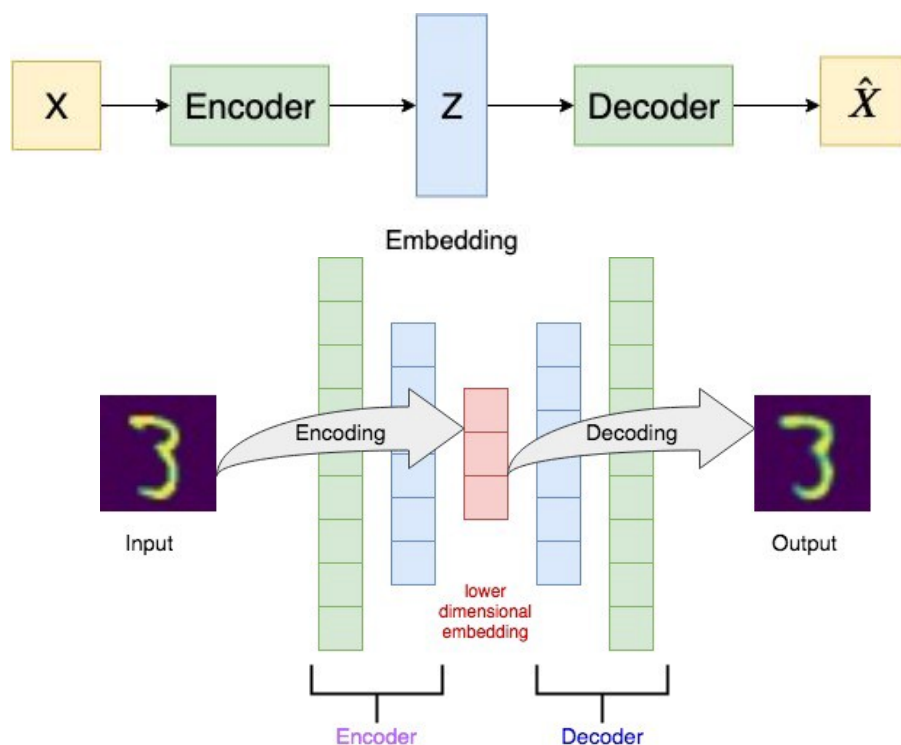
做法及创新

将已知的图进行编码（图卷积）得到图中顶点向量表示的一个分布，在分布中采样得到顶点的向量表示，然后进行解码重新构建图。

变分自编码器

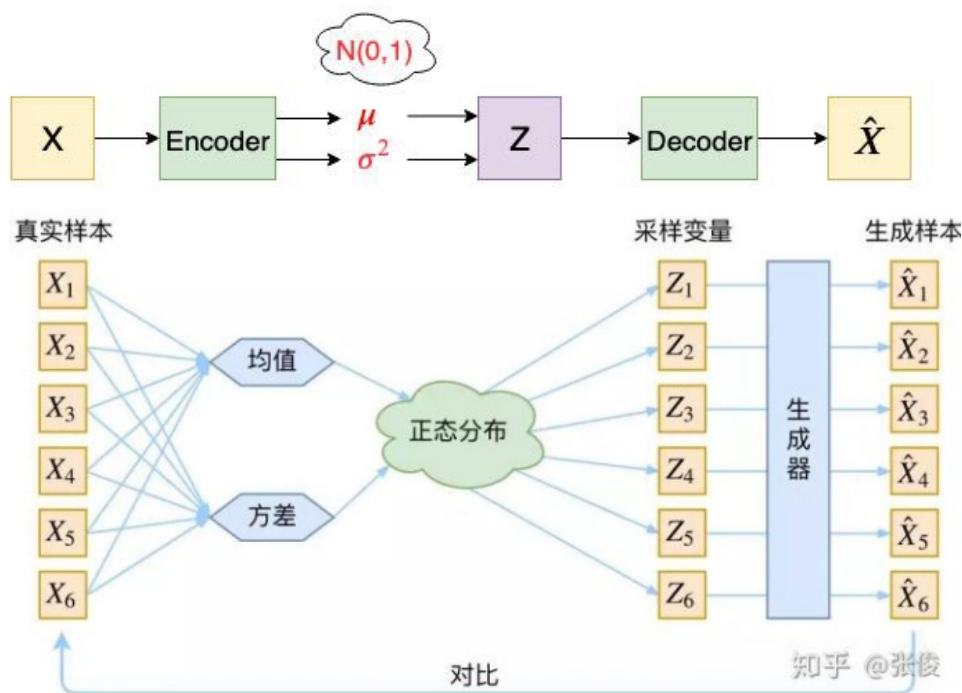
因为这篇论文做的是一个迁移的工作，变分自编码器的背景对于理解这篇论文来说十分重要，首先进行介绍。

变分自编码器是自编码器的一种，一个自编码器由编码器和解码器构成，编码器将输入数据转换为低维向量表示，解码器通过得到的低维向量表示进行重构。

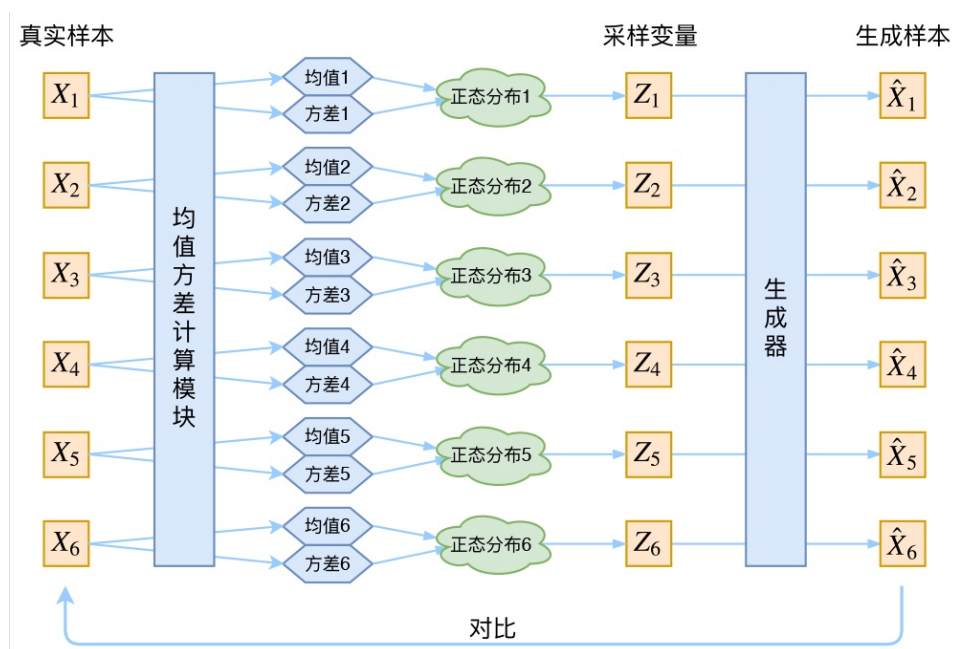


这种结构的不足之处在于，只能产生与输入数据相似的样本，而无法产生新的样本，低维向量表示必须是有真实样本通过编码器得到的，随机产生的低维向量经过重构几乎不可能得到近似真实的样本。而变分自编码器可以解决这个问题。

变分自编码器将输入数据编码为一个分布，而不是一个个低维向量表示，然后从这个分布中随机采样来得到低维向量表示。一般假设这个分布为正态分布，因此编码器的任务就是从输入数据中得到均值 μ 与方差 σ^2 。



然而，如果是将所有输入数据编码到同一个分布里，从这个分布中随机采样的样本 Z_i 无法与输入样本 X_i 一一对应，会影响模型的学习效果。所以，实际的变分自编码器结构如下图所示，为每一个输入样本学习一个正态分布：



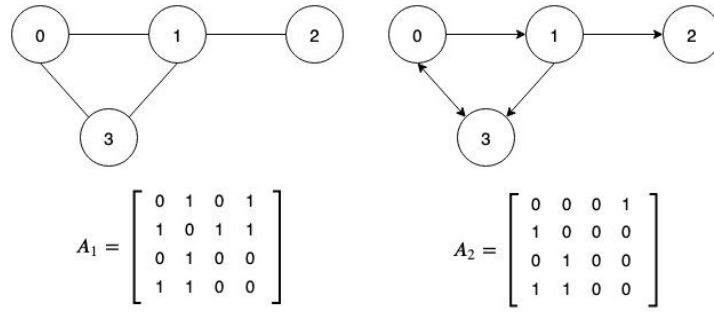
采样时常用"重参数"技巧(reparameterization trick)，从分布 $N(\mu, \sigma^2)$ 中采样一个 Z 相当于从 $N(0, 1)$ 中采样一个 ϵ 使得 $Z = \mu + \sigma * \epsilon$ 。

图变分自编码器

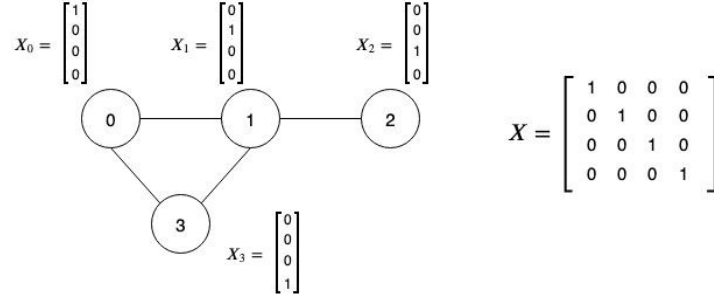
介绍完传统的变分自编码器，接下来就是介绍这篇论文的工作，如何将变分自编码器的思想迁移到图上。

针对图这个数据结构，输入的数据变为图的邻接矩阵 A 与特征矩阵 X ：

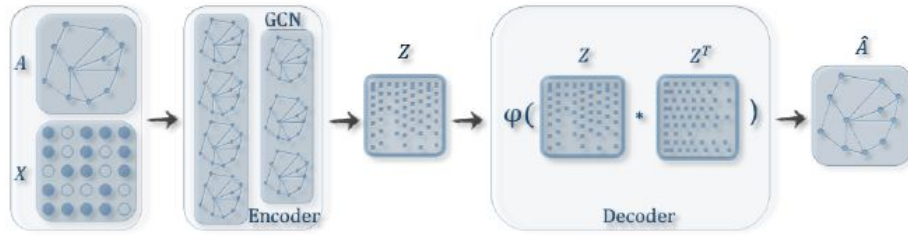
邻接矩阵 A ：



特征矩阵 X :



接下来的工作与变分自编码器相同，通过编码器（图卷积）学习图中顶点低维向量表示分布的均值 μ 与方差 σ^2 ，再通过解码器生成图。



编码器采用两层结构的图卷积网络，第一层产生一个低维的特征矩阵：

$$\bar{X} = \text{GCN}(X, A) = \text{ReLU}(\tilde{A}XW_0)$$

$$\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$$

第二层得到分布的均值 μ 与方差 σ^2 ：

$$\mu = \text{GCN}_\mu(X, A) = \tilde{A}\bar{X}W_1$$

$$\log \sigma^2 = \text{GCN}_\sigma(X, A) = \tilde{A}\bar{X}W_1$$

将两层网络的表达式合并可以得到编码器的表达式：

$$\text{GCN}(X, A) = \tilde{A}\text{ReLU}(\tilde{A}XW_0)W_1$$

同样地使用重参数技巧来得到低维向量表示 $Z = \mu + \sigma * \epsilon$ 。

编码器重构出图的邻接矩阵，从而得到一个新的图。之所以使用点积的形式来得到邻接矩阵，原因在于我们希望学习到每个顶点的低维向量表示 z 的相似程度，来更好地重构邻接矩阵。而点积可以计算两个向量之间的cosine相似度，这种距离度量方式不受量纲的影响。因此，重构的邻接矩阵可以学习到各个顶点之间的相似程度。

$$\hat{A} = \sigma(zz^T)$$

损失函数用于衡量生草样本与真是样本之间的差异，但如果只用距离度量作为损失函数，为了让编码器的效果最佳，模型会将方差的值学为0，这样从正态分布中采样出来的就是定值，有利于减小生成样本和真实样本之间的差异。但这样一来，就退化成了普通的自编码器，因此在构建损失函数时，往往还会加入各独立正态分布与标准正态分布的KL散度，来使得各个正态分布逼近标准正态分布：

$$L = E_{q(Z|X,A)}[\log p(A|Z)] - \text{KL}[q(Z|X,A)||p(Z)], \quad \text{where } p(Z) = N(0,1)$$

数据集

Cora、Citeseer、Pubmed