Variational Graph Autoencoder

Dominik Seitz

January 15, 2020

Table of Content

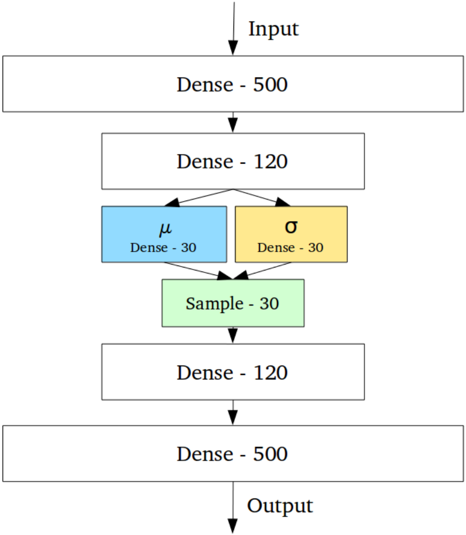
Recap

- Autoencoders are neural networks which condense down the feature space and reconstruct it again with minimal information loss but are known to overfit
- VAE, in contrast, return a distribution given a sample (not a point like normal AE) and hence are more robust

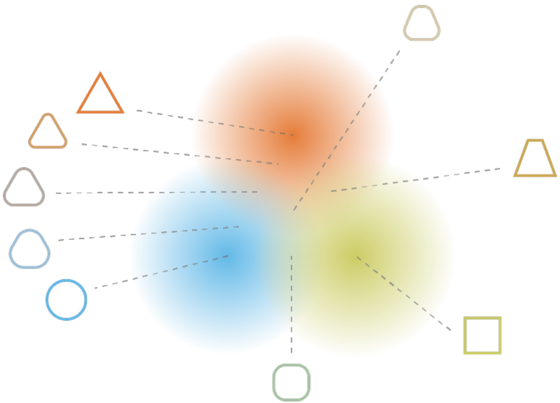
Recap: Variational Autoencoders

simple autoencoders	input **x**	encoding →	latent representation **z = e(x)**	decoding →	input reconstruction **d(z)**			
variational autoencoders	input **x**	encoding →	latent distribution **p(z|x)**	sampling →	sampled latent representation **z ~ p(z|x)**	decoding →	input reconstruction **d(z)**

VAE Architecture



Recap VAE



Computation graph run down

$$\begin{aligned}
\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x &= M(\mathbf{x}), \Sigma(\mathbf{x}) && \text{Push } \mathbf{x} \text{ through encoder} \\
\boldsymbol{\epsilon} &\sim \mathcal{N}(0,1) && \text{Sample noise} \\
\mathbf{z} &= \boldsymbol{\epsilon}\boldsymbol{\sigma}_x + \boldsymbol{\mu}_x && \text{Reparameterize} \\
\mathbf{x}_r &= p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) && \text{Push } \mathbf{z} \text{ through decoder}
\end{aligned}$$

$$\begin{aligned}
\text{recon. loss} &= \mathsf{MSE}(\mathbf{x}, \mathbf{x}_r) && \text{Compute reconstruction loss} \\
\text{var. loss} &= -\mathsf{KL}[\mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\sigma}_x) \| \mathcal{N}(0, I)] && \text{Compute variational loss} \\
\mathsf{L} &= \text{recon. loss} + \text{var. loss} && \text{Combine losses}
\end{aligned}$$

Variational Graph Autoencoder

- Apply the idea of VAE to graph-structured data.
- Generate new graphs or reason about graphs.

Problems

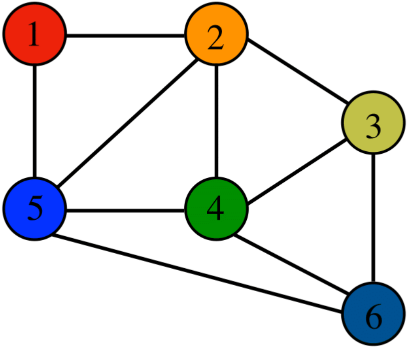
- Graph-structured data is irregular (variable size of unordered nodes / different number of neighbours)
- How to represent a graph in a way that a neural network can understand?

Basics for understanding VGAEs

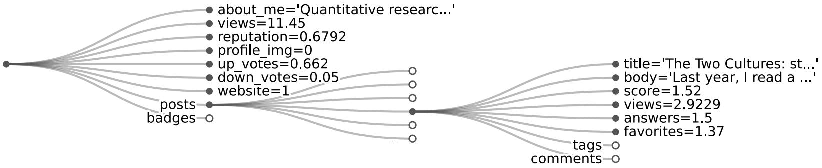
- Normal VAEs
- Graph-structured data
- Adjacency Matrices
- Feature Matrices
- Graph Convolutional Networks

Graphs

- Data structure consisting of vertices and edges $G = (V, E)$
- Can be directed or undirected / cyclic or acyclic



Graph structured data example

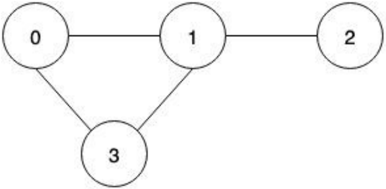


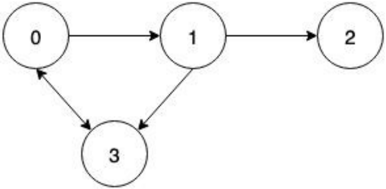
[1] Image taken from Janisch, Jaromír, Tomáš Pevný, and Viliam Lisý. "Deep Reinforcement Learning with Explicitly Represented Knowledge and Variable State and Action Spaces." arXiv preprint arXiv:1911.08756 (2019)

How to express relationships between nodes

1. Enumerate all nodes in a graph
2. Binary approach: Are node i and j connected by an edge?

Adjacency Matrix





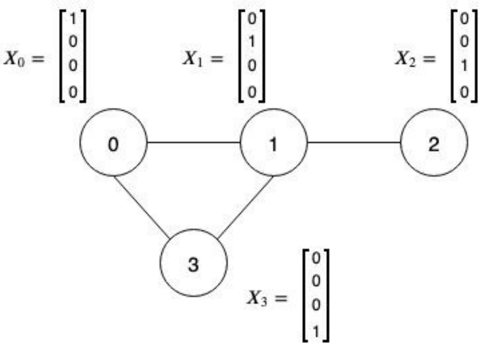
$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

How to express information about nodes

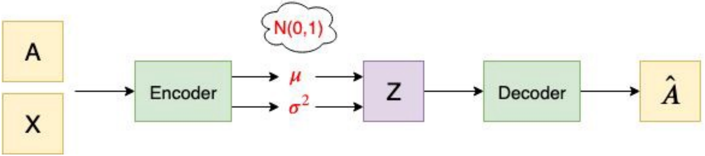
- Simplest example: Embed information about a node by a one-hot encoding vector

Feature Matrix

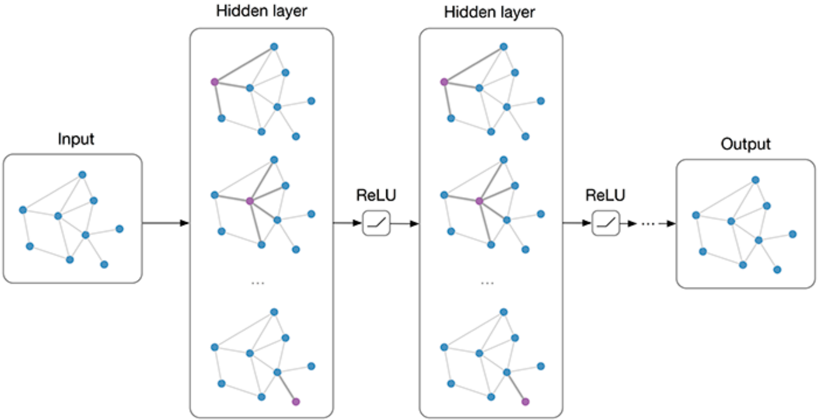


$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

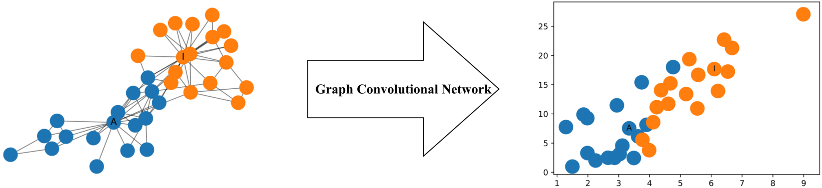
VGAE Architecture



Graph Convolutional Network (GCN): The Encoder of a VGAE



Objective of a GCN



GCN First Layer

Encoder takes an adjacency matrix A and a feature matrix X and generates the latent representation Z

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

where $\tilde{A}$ is the symmetrically normalized adjacency matrix using degree matrices of A:

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

GCN Second Layer

The second layer generates μ and $\log \sigma^2$

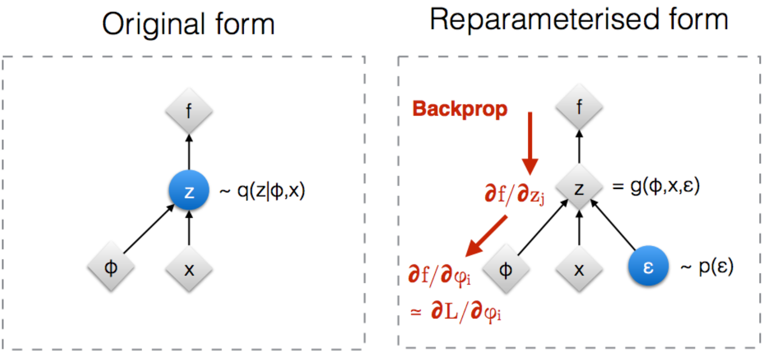
$$\mu = GCN_{\mu}(X, A) = \tilde{A}\bar{X}W_1$$
$$log\sigma^2 = GCN_{\sigma}(X, A) = \tilde{A}\bar{X}W_1$$

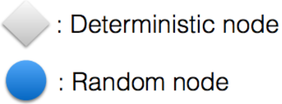
Both GCN layers combined

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

We end up with $Z = \mu + \sigma * \epsilon$ using the parameterization trick where $\epsilon \sim N(0, 1)$

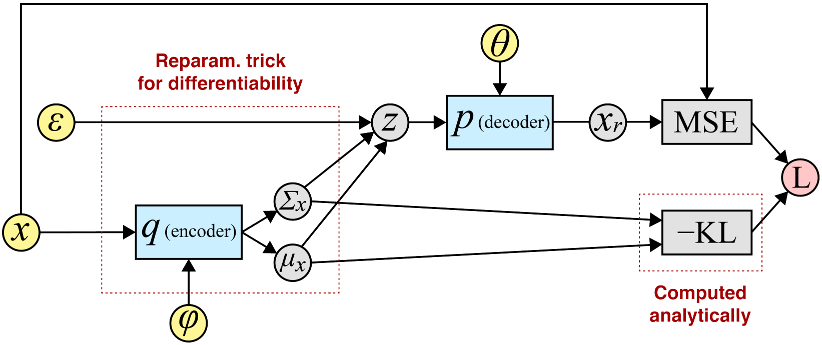
Reparameterization trick





[Kingma, 2013]
[Bengio, 2013]
[Kingma and Welling 2014]
[Rezende et al 2014]

Reparameterization trick



Decoder

The reconstructed adjacency matrix $\hat{A}$ is given by the inner product of the latent variable z:

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

VGAE Loss Function

$$L = E_{q(Z|X,A)}[logp(A|Z)] - KL[q(Z|X,A)||p(Z)]$$

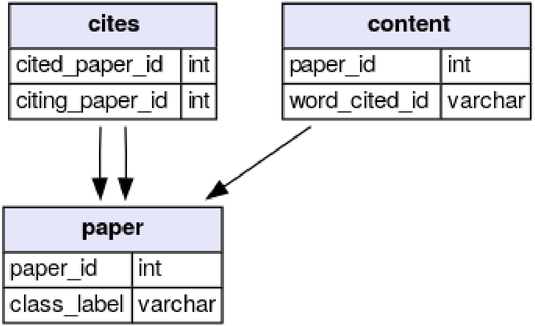
VGAE Applications

- Graph-structured data can be found in social networks, citations, links etc

Cora citation network dataset

The Cora dataset consists of 2708 scientific publications classified into one of seven classes. The citation network consists of 5429 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1433 unique words.

Cora citation network dataset illustration

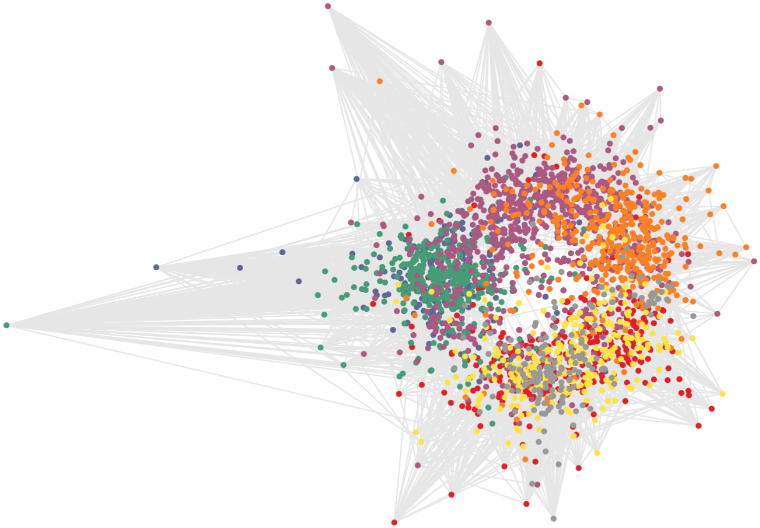


Results on Cora citation network dataset

Table 1: Link prediction task in citation networks. See [1] for dataset details.

Method	Cora		Citeseer		Pubmed	
	AUC	AP	AUC	AP	AUC	AP
SC [5]	84.6 ± 0.01	88.5 ± 0.00	80.5 ± 0.01	85.0 ± 0.01	84.2 ± 0.02	87.8 ± 0.01
DW [6]	83.1 ± 0.01	85.0 ± 0.00	80.5 ± 0.02	83.6 ± 0.01	84.4 ± 0.00	84.1 ± 0.00
GAE*	84.3 ± 0.02	88.1 ± 0.01	78.7 ± 0.02	84.1 ± 0.02	82.2 ± 0.01	87.4 ± 0.00
VGAE*	84.0 ± 0.02	87.7 ± 0.01	78.9 ± 0.03	84.1 ± 0.02	82.7 ± 0.01	87.5 ± 0.01
GAE	91.0 ± 0.02	92.0 ± 0.03	89.5 ± 0.04	89.9 ± 0.05	$\mathbf{96.4} \pm 0.00$	$\mathbf{96.5} \pm 0.00$
VGAE	$\mathbf{91.4} \pm 0.01$	$\mathbf{92.6} \pm 0.01$	$\mathbf{90.8} \pm 0.02$	$\mathbf{92.0} \pm 0.02$	94.4 ± 0.02	94.7 ± 0.02

Latent Space of VGAE on Cora citation network dataset



Summary

- Graph-structured data is becoming increasingly important in various research areas
- VGAEs apply the idea of a VAE to graph-structurd data
- They are used to reconstruct relationships nodes in graphs