
Variational Autoencoders: A Review

Aditya Garg
160046

Arpit Gupta
160149

Hritvik Taneja
160300

Mayank Shrivastava
160393

Pranjal Pratik Lal
160496

Disclaimer

This is to certify that the work carried out in the project has not been re-used from any another course project at IITK or elsewhere.

1 Overview

Variational Autoencoders (VAEs) allow us to leverage neural networks to encode our data in an interpretable latent space in a probabilistic manner. The probabilistic formulation of the model allows us to sample and generate data from the learnt distribution. While initial models of VAEs, like most deep learning models, often resulted in latent representations which were not amenable for human interpretation, there have been recent advances on disentangling these representations, so that by perturbing each latent factors a meaning can be assigned to it. Besides this, other variations of the VAE model also include an architecture to learn representations from graphical data, and use it for various tasks like link prediction, and joint distribution models which have been used for tasks like image captioning. Through this report, we aim to survey some of the recent advances in this field, and test some of the state-of-the-art methods on our own.

2 Variational Autoencoders

Variational Autoencoders were introduced by Kingma & Welling (2013). VAE's incorporated variational inference to the traditional auto-encoders architecture.

2.1 Variational Inference

Let us consider a dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ consisting of N i.i.d. samples of some continuous or discrete variable \mathbf{x} . We assume that the data is generated, by some random process involving a latent variable \mathbf{z}_i corresponding to each data point \mathbf{x}_i . The generative story of any data point can be described as

- \mathbf{z}_i is sampled from prior $p_\theta(\mathbf{z})$
- \mathbf{x}_i is generated from some conditional $p_\theta(\mathbf{x}|\mathbf{z})$

We don't make any simplifying assumptions about the marginal or posterior probabilities (such as conjugacy). We wish to perform efficient inference and learning in directed probabilistic models.

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{\int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}$$

Since \mathbf{z} is continuous, and $p(\mathbf{x}|\mathbf{z})$ and $p(\mathbf{z})$ are not necessarily conjugate to each other, hence $\int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ is intractable. So, we cannot use the traditional EM algorithm here because it requires us to calculate the posterior distribution. For the purpose of solving the above problems, lets

introduce a recognition model $q_\phi(\mathbf{z}|\mathbf{x})$: this is an approximation of the intractable true posterior $p_\theta(\mathbf{z}|\mathbf{x})$.

$$\begin{aligned}
\log p_\theta(\mathbf{X}) &= \sum_{i=1}^N \log p_\theta(\mathbf{x}_i) \\
\log p_\theta(\mathbf{x}_i) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)} [\log p_\theta(\mathbf{x}_i)] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)} \left[\log \frac{p_\theta(\mathbf{x}_i, \mathbf{z}) q_\phi(\mathbf{z}|\mathbf{x}_i)}{p_\theta(\mathbf{z}|\mathbf{x}_i) q_\phi(\mathbf{z}|\mathbf{x}_i)} \right] \\
&= D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i) || p_\theta(\mathbf{z}|\mathbf{x}_i)) + \mathcal{L}(\theta, \phi; \mathbf{x}_i) \\
\Rightarrow \log p_\theta(\mathbf{x}_i) &\geq \mathcal{L}(\theta, \phi; \mathbf{x}_i) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)} [\log p_\theta(\mathbf{x}_i, \mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x}_i)] \\
\text{Since } p_\theta(\mathbf{x}_i, \mathbf{z}) &= p_\theta(\mathbf{x}_i|\mathbf{z}) p_\theta(\mathbf{z}) \text{ So,} \\
\mathcal{L}(\theta, \phi; \mathbf{x}_i) &= -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i) || p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)} [\log p_\theta(\mathbf{x}_i|\mathbf{z})]
\end{aligned}$$

Now we would like to differentiate and optimize the variational lower bound (ELBO) $\mathcal{L}(\theta, \phi; \mathbf{x}_i)$ with respect to the parameters θ and ϕ . The authors of the paper claim that the usual naive way of Monte Carlo estimator for the gradient with respect to ϕ yields a high variance, hence that method cannot be used. Instead, they introduce a practical estimator of the lower bound and its derivatives w.r.t. the parameters.

Under certain conditions for a chosen approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ we can reparameterize the random variable $\tilde{\mathbf{z}} \sim q_\phi(\mathbf{z}|\mathbf{x})$ using a differentiable transformation $g_\phi(\epsilon, \mathbf{x})$ (explained in more detail in the next subsection)

$$\tilde{\mathbf{z}} = g_\phi(\epsilon, \mathbf{x}) \text{ with } \epsilon \sim p(\epsilon) \text{ where } \epsilon \text{ is an auxiliary noise variable}$$

There is a one to one correspondence between each \mathbf{z} and ϵ . The Monte Carlo estimates of expectations of some function $f(\mathbf{z})$ w.r.t. $q_\phi(\mathbf{z}|\mathbf{x})$ as follows

$$\begin{aligned}
\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)} [f(\mathbf{z})] &= \mathbb{E}_{p(\epsilon)} [f(g_\phi(\epsilon, \mathbf{x}_i))] \\
&\simeq \frac{1}{L} \sum_{l=1}^L f(g_\phi(\epsilon_l, \mathbf{x}_i)) \text{ where } \epsilon_l \sim p(\epsilon)
\end{aligned}$$

Applying this estimation to our variational lower bound $\mathcal{L}(\theta, \phi; \mathbf{x}_i)$ we get an estimator $\tilde{\mathcal{L}}^A(\theta, \phi; \mathbf{x}_i)$ of the lower bound

$$\begin{aligned}
\tilde{\mathcal{L}}^A(\theta, \phi; \mathbf{x}_i) &= \frac{1}{L} \sum_{l=1}^L \log p_\theta(\mathbf{x}_i, \mathbf{z}_{il}) - \log q_\phi(\mathbf{z}_{il}|\mathbf{x}_i) \\
&\text{where } \mathbf{z}_{il} = g_\phi(\epsilon_{il}, \mathbf{x}_i) \text{ and } \epsilon_{il} \sim p(\epsilon)
\end{aligned}$$

In a lot of cases $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i) || p_\theta(\mathbf{z}))$ can be integrated analytically (eg: when both prior and the recognition model are Gaussian). So, this yields a second estimator $\tilde{\mathcal{L}}^B(\theta, \phi; \mathbf{x}_i)$ of $\mathcal{L}(\theta, \phi; \mathbf{x}_i)$ where we just have to estimate $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)} [\log p_\theta(\mathbf{x}_i|\mathbf{z})]$ because we have the closed form solution of the KL divergence term.

$$\begin{aligned}
\tilde{\mathcal{L}}^B(\theta, \phi; \mathbf{x}_i) &= -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i) || p_\theta(\mathbf{z})) + \frac{1}{L} \sum_{l=1}^L \log p_\theta(\mathbf{x}_i|\mathbf{z}_{il}) \\
&\text{where } \mathbf{z}_{il} = g_\phi(\epsilon_{il}, \mathbf{x}_i) \text{ and } \epsilon_{il} \sim p(\epsilon)
\end{aligned}$$

The KL-Divergence term can be thought of as a regularizer for ϕ and $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)} [\log p_\theta(\mathbf{x}_i|\mathbf{z})]$ can be thought of as the expected reconstruction error.

2.2 Reparametrization

In order to solve our problem we invoked an alternative way to generate samples from $q_\phi(\mathbf{z}|\mathbf{x})$. Let \mathbf{z} be a continuous random variable, such that $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ is some conditional. This reparameterization is useful, because it gives us a differentiable Monte Carlo estimate of the expectation w.r.t. ϕ .

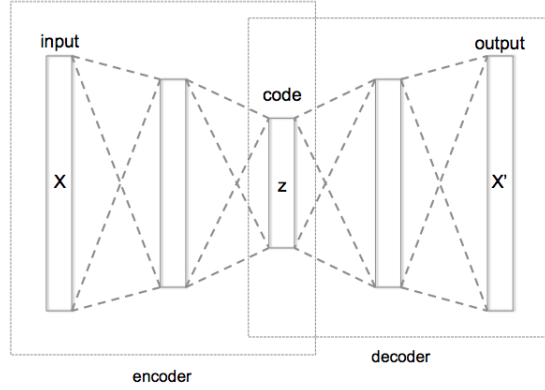


Figure 1: Basic autoencoder structure

It is often possible to express \mathbf{z} as a deterministic variable $\mathbf{z} = g_\phi(\epsilon, \mathbf{x})$, where ϵ is an auxiliary variable with independent marginal $p(\epsilon)$, and g_ϕ is some vector-valued function. Since there is a one to one correspondence between each \mathbf{z} and each ϵ . So

$$\begin{aligned}
 Pr(\mathbf{z} = \mathbf{z}_1) &= Pr(\epsilon = \epsilon_1) \text{ for some } \epsilon = \epsilon_1 \text{ and } \mathbf{z} = \mathbf{z}_1 \\
 \Rightarrow q_\phi(\mathbf{z}_1|\mathbf{x})d\mathbf{z}_1 &= p(\epsilon_1)d\epsilon_1 \\
 \Rightarrow \int_{\mathbf{z}_1} q_\phi(\mathbf{z}_1|\mathbf{x})f(\mathbf{z}_1)d\mathbf{z}_1 &= \int_{\epsilon_1} p(\epsilon_1)f(\mathbf{z}_1)d\epsilon_1 = \int_{\epsilon_1} p(\epsilon_1)f(g_\phi(\epsilon_1, \mathbf{x}))d\epsilon_1
 \end{aligned}$$

Using Monte Carlo approximation

$$\int_{\mathbf{z}_1} q_\phi(\mathbf{z}_1|\mathbf{x})f(\mathbf{z}_1)d\mathbf{z}_1 \simeq \frac{1}{L} \sum_{l=1}^L f(g_\phi(\epsilon_l, \mathbf{x})) \text{ where } \epsilon_l \sim p(\epsilon)$$

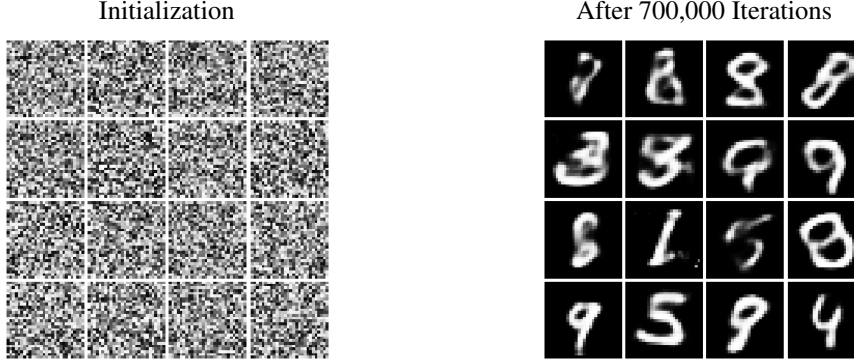
Example: for $z \sim p(z|x) = \mathcal{N}(\mu, \sigma^2)$, $z = \mu + \sigma\epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$ is a valid reparameterization. Author gives 3 criterion on $q_\phi(\mathbf{z}|\mathbf{x})$ for which we can choose such a differentiable transformation $g_\phi(\epsilon, \mathbf{x})$. Auto-encoders are just an encoder decoder model that tries to minimize the \mathcal{L}_2 reconstruction loss. We can add Variational Inference to this if we think of the encoder as $q_\phi(z|x)$ and decoder as $p_\theta(x|z)$.

2.3 Results

As a starting point we implemented a VAE model on MNIST data and obtained the following images and ELBO:

Epochs	ELBO
1000	146.9
2000	128.7
42000	12.8
46000	105.2

Table 1: VAE on MNIST



3 β -VAE

3.1 Need for Disentanglement

A disentangled representation is which separates the factors of variations, ensuring that change in one unit of a latent variable causes changes in only dimension of the data. This is an important step in improving performance of deep learning algorithms in areas where they currently struggle compared to humans, including tasks like transfer learning (by reusing learnt representations) and zero-shot learning (by recombining previously learnt factors to reason for new data). β -VAE was one of the first papers to introduce a scalable approach to disentangling the latent representations in an unsupervised manner in the VAE framework.

3.2 β -VAE

A modified version of the VAE was introduced by Higgins et al. (2016). A modified version of the VAE objective is used with a larger weight ($\beta > 1$) on the KL divergence between the variational posterior and prior.

$$\mathcal{L}(\theta, \phi; \mathbf{x}_i) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)}[\log p_\theta(\mathbf{x}_i|\mathbf{z})] - \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{z}))$$

The KL term can be shown as the sum of the mutual information term $\mathbb{I}(\mathbf{x}, \mathbf{z})$ and $D_{KL}(q_\phi(\mathbf{z})||p_\theta(\mathbf{z}))$. Penalizing both of these, while reducing the amount of information about \mathbf{x} stored in \mathbf{z} thus leading to poorer reconstruction quality, also pushes $q(\mathbf{z})$ towards prior $p(\mathbf{z})$, encouraging independence in the dimensions of \mathbf{z} as the prior has a diagonal co-variance matrix and hence already factorized dimensions.

3.3 Metrics for Disentanglement

The qualitative method of visual inspection by inspecting latent traversals remains the most popular one, where the changes in the reconstructions are visualized while traversing one dimension of the latent space. For datasets where the ground truth of the true generative factors (taken as \mathbf{v} subsequently) is not available, this still remains the only method.

The paper attempts to quantify disentanglement by defining a new metric.

1. Take a single factor v_k . Generate data by keeping this factor fixed and varying all other factors.
2. Take the pairwise difference of the corresponding learnt representations and average them out.
3. Train a linear classifier with each averaged representation as the input, and try to predict which of the factor was initially kept fixed.
4. The accuracy of this classifier will quantitatively define the amount of disentanglement.

Eastwood & Williams (2018) try to improve upon this metric by judging disentangled representations using three different criteria, *disentanglement*, *completeness* and *informativeness*. Since most of the unsupervised data that we have available to us does not include ground truth, developing a quantitative metric which does not need ground truth of the latent variables can be an important aim in future research work.

3.4 Further Work

Since the introduction of unsupervised disentanglement in this paper, there have been several other papers that have further build upon this work.

- **FactorVAE:** This paper attempts to improve the reconstruction tradeoff of β -VAE. Instead of modifying the VAE objective term, it augments it with a negative Total Correlation term that directly encourages independence. A major problem in the β -VAE metric is that it tends to give 100% accuracy if K-1 out of K ground truth variables have been disentangled. This paper tries to improve this by introducing a new metric based on a majority vote classifier. [Kim & Mnih (2018)]
- **HFVAE:** Here, the KL divergence term weighted by β in the original paper is decomposed into two parts, and two different weights are assigned to them. This decomposition also allows the authors to induce correlation between latent variables. The paper also considers an example beyond traditional image-based domains by training on textual data, leading to different latent factors for different topics. [Esmaeili et al. (2018)]
- Work is also being done to gain a better understanding of why applying pressure on the channel capacity term in models like β -VAE to make the latent variables independent induces disentanglement which aligns with the human intuition of data generation factors. The insights developed led to the authors proposing an alternate training regime for the β -VAE model where the encoding capacity of the latent variables is increased gradually instead of a fixed β , which is shown to have improved the reconstruction fidelity. [Burgess et al. (2018)]

3.5 Results

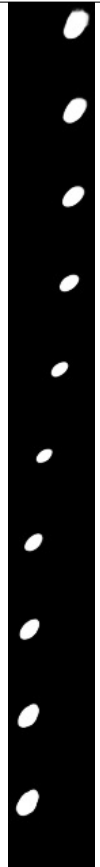
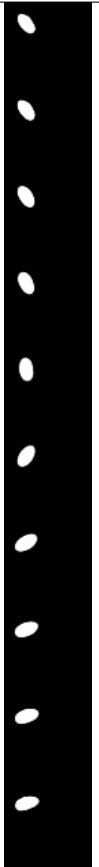
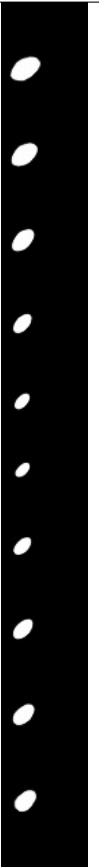


For implementing this paper, we used the dsprites dataset by Matthey et al. (2017), which is the standard dataset for evaluation of disentanglement in unsupervised learning methods. The dataset has been generated by taking 6 ground truth independent latent factors. By keeping the number of latent variables to ten, we were able to learn five disentangled representations of the possible six. The results obtained are shown below:

4 Learning Disentangled Representations with Semi-Supervised Deep Generative Models

4.1 Introduction

Similar to β -VAE this paper, authored by Narayanaswamy et al. (2017) focuses on learning disentangled representations. The model proposed in this paper learns disentangled representations for those axes of variations (and their dependencies) for which we have information about. For example, Suppose we have a fully unsupervised setting for MNIST dataset with 10 latent variables, then there is no guarantee that the 10 latent variables will recover the 10 digits. However, if the "digit" parameter is known for a subset of the data, the model proposed by this paper will learn a latent variable capturing only the "digit", independent of other factors. The paper achieves the goal of defining and learning Partially Specified Models in a Semi-Supervised Environment.

Table 2: β -VAE on dsprites Dataset

Varying z_o Position(x)	Varying z_1 Rotation	Varying z_6 Position(y)	Varying z_8 Size	Varying z_9 Shape
				

4.2 Objective Function

Loss Function for Unsupervised part is identical to the original VAE Paper

$$\mathcal{L}(\theta, \phi; \mathcal{D}, \mathcal{D}^{\text{sup}}) = \sum_{n=1}^N \mathcal{L}(\theta, \phi; \mathbf{x}^n) + \gamma \sum_{m=1}^M \mathcal{L}^{\text{sup}}(\theta, \phi; \mathbf{x}^m, \mathbf{y}^m)$$

Where the loss function for the Supervised part is

$$\mathcal{L}^{\text{sup}}(\theta, \phi; \mathbf{x}^m, \mathbf{y}^m) = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}^m, \mathbf{y}^m)} \log \frac{p_{\theta}(\mathbf{x}^m, \mathbf{y}^m, \mathbf{z})}{q_{\phi}(\mathbf{z} | \mathbf{x}^m, \mathbf{y}^m)} + \alpha \log q_{\phi}(\mathbf{y}^m | \mathbf{x}^m)$$

Importance sampling is used to estimate the above quantity. Where the conditional dependency structure is $q_{\phi}(\mathbf{z}, \mathbf{y}|\mathbf{x}) = q_{\phi_z}(\mathbf{z}|\mathbf{y}, \mathbf{x})q_{\phi_y}(\mathbf{y}|\mathbf{x})$.

4.3 Model

The graph contains three types of sub-graphs, corresponding to the three possibilities for supervision and gradient estimation:

- For the fully supervised variable likelihood is computed using a neural network under generative model.
- For the unobserved variable \mathbf{z} , both the prior probability, and the conditional probability are computed.

- For the partially observed variable y , probabilities $p(y)$ and $q_\phi(y|x)$ are computed. The value y is treated as observed when available, and sampled otherwise.

4.4 Results

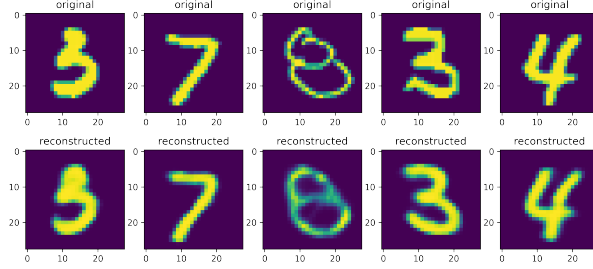


Figure 2: Reconstruction on MNIST dataset

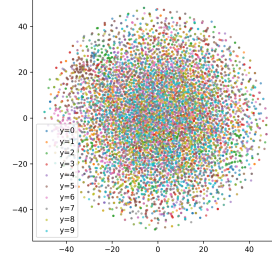


Figure 3: Latent variable embeddings in 2D

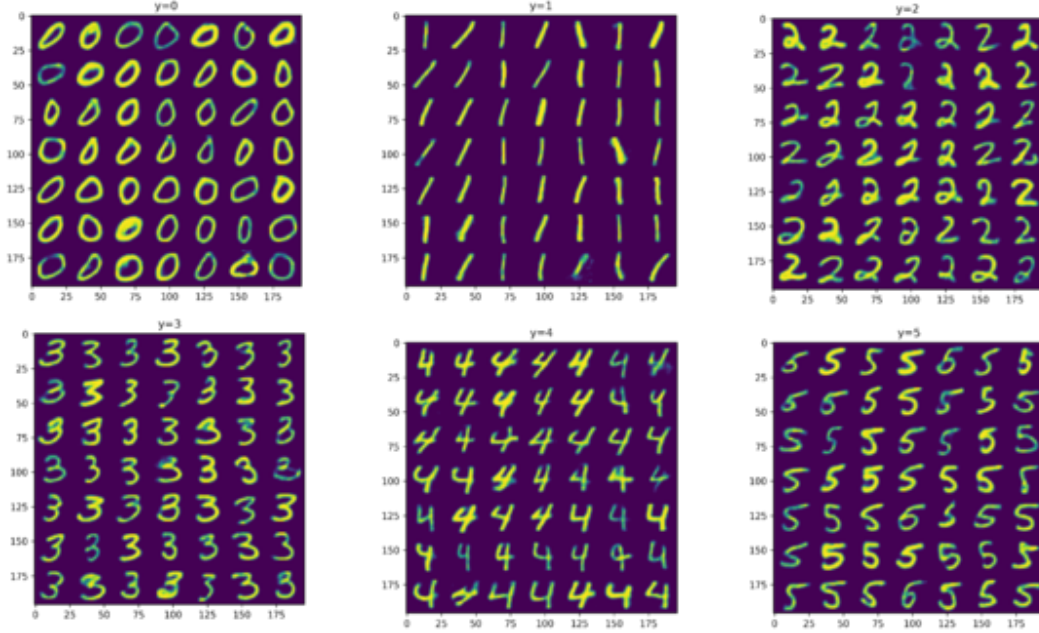


Figure 4: Samples obtained by keeping y fixed

5 Variational Graph Auto-Encoders

The variational graph autoencoder is a framework for unsupervised learning on graph-structured data based on VAEs. This was introduced by Kipf & Welling (2016). This models tries to learn latent variables given input features and tries to predict whether there is a edges between two nodes.

5.1 Inference model

We define a Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $N = |\mathcal{V}|$ are number of nodes, \mathbf{A} is the Adjacency Matrix, \mathbf{D} is the Degree Matrix and \mathbf{X} (size $N \times D$) are the node features. We define \mathbf{Z} (size $N \times F$) as the Stochastic Latent Variables. Now we will define the inference model. Let us define the conditional

	Original		Reproduced	
	AUC	AP	AUC	AP
GAE*	0.843	0.881	0.84611	0.88393
VGAE*	0.840	0.877	0.85273	0.88931
GAE	0.910	0.920	0.92091	0.92371
VGAE	0.914	0.926	0.92113	0.92824
*without Input features				

Table 3: Cora Dataset

	Original		Reproduced	
	AUC	AP	AUC	AP
GAE*	0.787	0.841	0.76928	0.82555
VGAE*	0.789	0.841	0.78661	0.82890
GAE	0.895	0.899	0.87490	0.88316
VGAE	0.908	0.920	0.90238	0.91601
*without Input features				

Table 4: Citeseer Dataset

probability of latent variables \mathbf{Z} as

$$q(\mathbf{Z}|\mathbf{X}, \mathbf{A}) = \prod_{i=1}^N q(\mathbf{z}_i|\mathbf{X}, \mathbf{A}) = \prod_{i=1}^N \mathcal{N}(\boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2))$$

Here $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ are the i^{th} row of $\boldsymbol{\mu}$ (size $N \times F$) and $\boldsymbol{\sigma}$ (size $N \times F$) respectively. Mean $\boldsymbol{\mu}$ and Variance $\boldsymbol{\sigma}$ are estimated using a Graph Convolution Network (GCN) as follows

$$\boldsymbol{\mu} = GCN_{\boldsymbol{\mu}}(\mathbf{X}, \mathbf{A})$$

$$\log(\boldsymbol{\sigma}) = GCN_{\boldsymbol{\sigma}}(\mathbf{X}, \mathbf{A})$$

A two layer GCN is defined as $GCN(\mathbf{X}, \mathbf{A}) = \tilde{\mathbf{A}} ReLU(\tilde{\mathbf{A}} \mathbf{X} \mathbf{W}_0) \mathbf{W}_1$. We define $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ as the symmetrically normalized adjacency matrix. Also $GCN_{\boldsymbol{\mu}}(\mathbf{X}, \mathbf{A})$ and $GCN_{\boldsymbol{\sigma}}(\mathbf{X}, \mathbf{A})$ share the first layer parameters \mathbf{W}_0 .

5.2 Generative model

This is the definition of the generative model that is used

$$p(\mathbf{A}|\mathbf{Z}) = \prod_{i=1}^N \prod_{j=1}^N p(A_{ij}|\mathbf{Z}) \text{ and } p(A_{ij} = 1|\mathbf{Z}) = \sigma(\mathbf{z}_i^T \mathbf{z}_j)$$

5.3 Loss Function

Now we define the ELBO loss for this model.

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{Z}|\mathbf{X}, \mathbf{A})} [\log p(\mathbf{A}|\mathbf{Z})] + KL(q(\mathbf{Z}|\mathbf{X}, \mathbf{A}) || p(\mathbf{Z}))$$

For a featureless approach, the dependence on \mathbf{X} simply dropped and \mathbf{X} is replaced with the identity matrix in the GCN. A Gaussian Prior is taken for \mathbf{Z} such that $p(\mathbf{Z}) = \prod_{i=1}^N p(\mathbf{z}_i) = \prod_{i=1}^N \mathcal{N}(\mathbf{z}_i|0, \mathbf{I})$. This model is known as the VGAE model. There also exists a non-probabilistic variant of the VGAE model known as the GAE model, here the predicted adjacency matrix is defined as follows

$$\hat{\mathbf{A}} = \sigma(\mathbf{Z} \mathbf{Z}^T) \text{ where } \mathbf{Z} = GCN(\mathbf{X}, \mathbf{A})$$

5.4 Future Work

Future work can search for better prior distributions, more flexible generative models and the application of a stochastic gradient descent algorithm for improved scalability.

6 Generative models of visually grounded imagination

The ability to create diverse images corresponding to novel concrete or partially specified (abstract) concepts, on both familiar and unseen concepts was first introduced in Vedantam et al. (2017). The authors extend the traditional VAE to the multi-modal setting where one can have an image and an attribute vector. The setting assumes a joint generative model of the form

$$p(\mathbf{x}, \mathbf{y}, \mathbf{z}) = p(\mathbf{z}) p(\mathbf{x}|\mathbf{z}) p(\mathbf{y}|\mathbf{z})$$

where $p(\mathbf{z})$ is the prior over the latent variable \mathbf{z} , $p(\mathbf{x}|\mathbf{z})$ is the image decoder and $p(\mathbf{y}|\mathbf{z})$ is the description decoder. The authors propose a product-of-experts inference network using a joint variational autoencoder model, with a new objective called triple evidence lower bound, or TELBO.

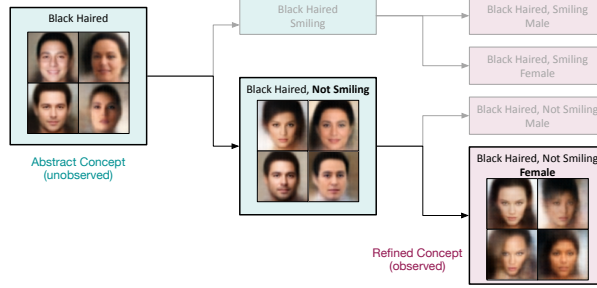


Figure 5: A compositional abstraction hierarchy for faces, derived from 3 attributes

6.1 Objective Function

The authors first define evidence lower bound (elbo) as follows:

$$\text{elbo}_{\lambda, \beta}(\mathbf{x}, \boldsymbol{\theta}, \phi) = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}, \phi)}[\lambda \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - \beta \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x})p_{\boldsymbol{\theta}}(\mathbf{z}))$$

$$\text{Standard VAE: } \mathcal{L}(\boldsymbol{\theta}, \phi) = \mathbb{E}_{\hat{p}(\mathbf{x})}[\text{elbo}(\mathbf{x}, \boldsymbol{\theta}, \phi)] \text{ , } \hat{p}(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{x}_n}(\mathbf{x}_n)$$

This is similar to the form described in β -VAE where β can be adjusted to make the features of latent variable disentangled. Here, λ is used to scale the likelihood in different dimensional spaces.

Since the paper will be modelling both the image decoder and the description decoder it uses the approach of Joint VAE (JVAE) to define the following

$$\begin{aligned} \text{elbo}_{\lambda_x, \lambda_y, \beta}(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_y, \phi) &= \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}, \phi)}[\lambda_x \log p_{\boldsymbol{\theta}_x}(\mathbf{x}|\mathbf{z}) + \lambda_y \log p_{\boldsymbol{\theta}_y}(\mathbf{y}|\mathbf{z})] \\ &\quad - \beta \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y})p_{\boldsymbol{\theta}}(\mathbf{z})) \end{aligned}$$

$$\text{Joint VAE: } \mathcal{L}(\boldsymbol{\theta}, \phi) = \mathbb{E}_{\hat{p}(\mathbf{x}, \mathbf{y})}[\text{elbo}(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}, \phi)] \text{ , } \hat{p}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{x}_n}(\mathbf{x}_n) \delta_{\mathbf{y}_n}(\mathbf{y}_n)$$

In this setting β is usually set to 1 and λ_y/λ_x is greater than 1 to scale up the likelihood from low dimensional attribute vector $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{z})$, to match $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$.

Once the the joint model is fixed, one can proceed to train inference networks $q_{\phi_x}(\mathbf{z}|\mathbf{x})$ and $q_{\phi_y}(\mathbf{z}|\mathbf{y})$. The fit for this is obtained by maximizing $\mathcal{L}(\phi_x|\boldsymbol{\theta})$ and $\mathcal{L}(\phi_y|\boldsymbol{\theta})$.

$$\begin{aligned} \mathcal{L}(\phi_x|\boldsymbol{\theta}) &= -\mathbb{E}_{\hat{p}(\mathbf{x})}[\text{KL}(q_{\phi_x}(\mathbf{z}|\mathbf{x}), p_{\boldsymbol{\theta}_x}(\mathbf{z}|\mathbf{x}))] \\ &= \mathbb{E}_{\hat{p}(\mathbf{x})}[\text{elbo}(\mathbf{x}, \boldsymbol{\theta}_x, \phi_x)] - \mathbb{E}_{\hat{p}(\mathbf{x})}[\log p_{\boldsymbol{\theta}_x}(\mathbf{x})] \end{aligned}$$

Note that last term in the above equation is constant w.r.t ϕ_x , hence can be dropped. Combining all the equation one get the TELBO (triple ELBO) objective

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}_x, \boldsymbol{\theta}_y, \phi_x, \phi_y, \phi) &= \mathbb{E}_{\hat{p}(\mathbf{x}, \mathbf{y})}[\text{elbo}_{1, \lambda, 1}(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_y, \phi) + \text{elbo}_{1, 1}(\mathbf{x}, \boldsymbol{\theta}_x, \phi_x) \\ &\quad + \text{elbo}_{\gamma, 1}(\mathbf{y}, \boldsymbol{\theta}_y, \phi_y)] \end{aligned}$$

In the above equation γ and λ scale the likelihood terms which can be set by cross validation.

6.2 Evaluation Metrics

The evaluation metrics introduced by the authors present a new way of evaluating the deep understanding of concepts for generative models in the case of abstract or unseen queries. They define the metrics in terms of 3Cs, i.e, correctness, coverage and compositionality.

- **Correctness** is defined as the fraction of attributes for each generated image that match those specified in the concept’s description.

$$\text{correctness}(\mathcal{S}, \mathbf{y}_{\mathcal{O}}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x} \in \mathcal{S}} \frac{1}{|\mathcal{O}|} \sum_{k \in \mathcal{O}} \mathbb{I}(\hat{y}(\mathbf{x})_k = y_k)$$

- **Coverage** compares the empirical distribution over values of attributes induced by the generated set to the true distribution for that attribute. This is done with the help of Jensen-Shannon divergence.

$$\text{coverage}(\mathcal{S}, \mathbf{y}_{\mathcal{O}}) = \frac{1}{|\mathcal{M}|} \sum_{k \in \mathcal{M}} (1 - \text{JS}(p_k, q_k))$$

- **Compositionality** is measured as the correctness of generated images in response to test concepts that differ in at least one attribute from training concepts.

6.3 Results

The experiments conducted (by authors) on different joint multimodal variational auto-encoder models differentiated by their objective functions, namely BiVCCA (Wang et al. (2016)), JMVAE (Suzuki et al. (2016)) and TELBO. **The results by TELBO stand out to others as shown in the research paper.** We are confident that the entirety of the paper could have been reproduced had we got more time and a more powerful hardware structure.

6.4 Future Work

This paper has shown models which can imagine compositionally concrete and abstract visual concepts. In the future we can extend this project to generate natural language description (given image) and generate attribute vectors (which can in-turn generate images) given natural language queries.

Software and tools used

All our model were either implemented in Tensorflow or Pytorch. These are automatic differentiation libraries, which have become really popular for Deep Learning models. Using these libraries, one can define the forward propagation in arbitrary computational graphs, and these libraries will compute the backpropagation itself. They also optimizes the graph for faster computation GPU, and provides various implementations of popular optimizers also.

Things we learnt

We learnt a lot from this project:

- We surveyed a lot of papers before deciding on a problem. This gave us a wholesome view of the field of generative modelling (though not complete yet!)
- We came up a lot of problems and possible approaches during the course of this project. However, (to our dismay sometimes!), these approaches had been tried out. Nonetheless, these approaches were more well-thought out and experimented with than we began with, thus giving us valuable lessons in research.
- We became more familiar with implementing deep learning models, going from an idea to implementation quickly. We also gained more proficiency in Tensorflow.

Acknowledgments

We would like to express my sincere gratitude to our instructor Prof. Piyush Rai for providing his invaluable guidance, comments and suggestions throughout the course of the project. We would also like to thank all the authors to provide their official repositories for their respective projects.

References

- Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G. & Lerchner, A. (2018), ‘Understanding disentangling in β -vae’, *arXiv preprint arXiv:1804.03599*.
- Eastwood, C. & Williams, C. K. (2018), ‘A framework for the quantitative evaluation of disentangled representations’.
- Esmacili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., Brooks, D. H., Dy, J. & van de Meent, J.-W. (2018), ‘Structured disentangled representations’, *stat* **1050**, 29.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S. & Lerchner, A. (2016), ‘beta-vae: Learning basic visual concepts with a constrained variational framework’.
- Kim, H. & Mnih, A. (2018), ‘Disentangling by factorising’, *arXiv preprint arXiv:1802.05983*.
- Kingma, D. P. & Welling, M. (2013), ‘Auto-encoding variational bayes’, *arXiv preprint arXiv:1312.6114*.
- Kipf, T. N. & Welling, M. (2016), ‘Variational graph auto-encoders’, *arXiv preprint arXiv:1611.07308*.
- Matthey, L., Higgins, I., Hassabis, D. & Lerchner, A. (2017), ‘dsprites: Disentanglement testing sprites dataset’, <https://github.com/deepmind/dsprites-dataset/>.
- Narayanaswamy, S., Paige, T. B., Van de Meent, J.-W., Desmaison, A., Goodman, N., Kohli, P., Wood, F. & Torr, P. (2017), Learning disentangled representations with semi-supervised deep generative models, in ‘Advances in Neural Information Processing Systems’, pp. 5925–5935.
- Suzuki, M., Nakayama, K. & Matsuo, Y. (2016), ‘Joint multimodal learning with deep generative models’, *arXiv preprint arXiv:1611.01891*.
- Vedantam, R., Fischer, I., Huang, J. & Murphy, K. (2017), ‘Generative models of visually grounded imagination’, *arXiv preprint arXiv:1705.10762*.
- Wang, W., Yan, X., Lee, H. & Livescu, K. (2016), ‘Deep variational canonical correlation analysis’, *arXiv preprint arXiv:1610.03454*.