Learning disentangled representations with Variational Autoencoders

Technical University of Cluj-Napoca - Faculty of Automation and Computers - Department of Computers - Master in Computer Vision and Deep Learning

Florian Moga – Research Activity 1st year 1st semester

1. Network Architectures

1.1. Autoencoder (AE)

AEs are a type of artificial neural networks that are used to learn efficient encodings of unlabeled data. They are an unsupervised learning technique that achieve lossy compression of the input data through an encoder and a decoder function. The encoder compresses the data (x) from a higher-dimensional space to a lower-dimensional space (also called the latent space), while the decoder tries to reconstruct the original data (x̂). Typically, they are used for dimensionality reduction, but can be used for other tasks, like denoising, image inpainting, anomaly detection etc. Ideally x and x̂ are identical, but due to the nature of the compression it is impossible to achieve such results without overfitting. AEs use a MSE loss (mean squared error) shown in Equation 1 that is minimized during network training. If linear activations are used, or only a single sigmoid hidden layer, then the optimal solution to an autoencoder is strongly related to principal component analysis (PCA).

$$\frac{1}{n}\sum_{i=1}^{n}(xi - \hat{x}i)^2$$

Equation 1 AE loss function

A high level AE architecture is illustrated in Figure 1. It illustrates best the main components, which are influenced by the task the network is designed for. The encoder and decoder network must be a good fit to the data, e.g. fully connected networks when handling tabular data, convolutional neural networks or visual transformers when handling images, long-short term memory networks or transformers when handling sequential data etc. The design of the latent space z is also very important. If it is too small, too much information may be lost and the reconstruction capabilities of the network may be reduced. If it is too big, downstream tasks such as clustering may suffer from it.

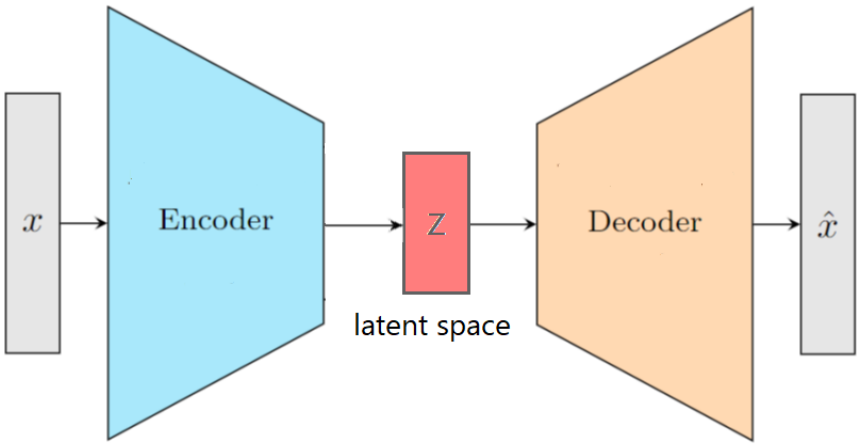

Figure 1 AE architecture

AEs have a couple of drawbacks that make them impractical for modern deep learning tasks:

- The loss function focuses only on the input data and reconstructed output, which leads to 2.
- The latent space is not meaningful. Data has many ground truth generative factors that ideally are captured and ordered in the latent space. By not constraining the latent space in any way, it will be unregularized and uninterpretable.
- It learns to capture as much information as possible rather than as much relevant information as possible. Relevant information may be lost and irrelevant features can be captured.
- Overfitting on the training data. While a task-specific AE may yield great results when inputting data similar to the training data, it will fail to generalize when inputting different data.

1.2. Variational Autoencoder (VAE)

VAEs were introduced in 2013 in the paper ”Auto-Encoding Variational Bayes” [1] and have a different mathematical foundation in comparison to AEs. They are probabilistic generative models that require neural networks as only a part of their overall structure. Instead of being just a lower-dimensional mapping of the input data, the latent space represents the parameters of a multivariate distribution. The VAE architecture has 3 important layers (or nodes) in addition to the AE architecture Figure 2. Instead of directly mapping the decoder to the latent space, it maps 2 layers, μ and σ, which have learnable parameters. Their names are chosen this way, because the values of these layers are used to sample from a normal distribution with μ mean and σ^2 variance - $N(\mu, \sigma^2)$. Because this computation would require the nodes to be stochastic and backpropagation would not be possible, the authors introduced the “Reparameterization trick” that allows backpropagation. Instead of using just the μ and σ layers to compute $N(\mu, \sigma^2)$, they added a new layer $\epsilon \sim N(0, 1)$ (which is stochastic and has no learnable parameters) with the same dimension as μ and σ that samples from a standard uniform distribution and showed that $N(\mu, \sigma^2) = \epsilon * \sigma^2 + \mu$. It is important to mention that μ, σ, ϵ and z must have the same dimensions, because $[z_0, ..., z_n] = [\epsilon_0 * \sigma^2_0 + \mu_0, ..., \epsilon_n * \sigma^2_n + \mu_n]$.

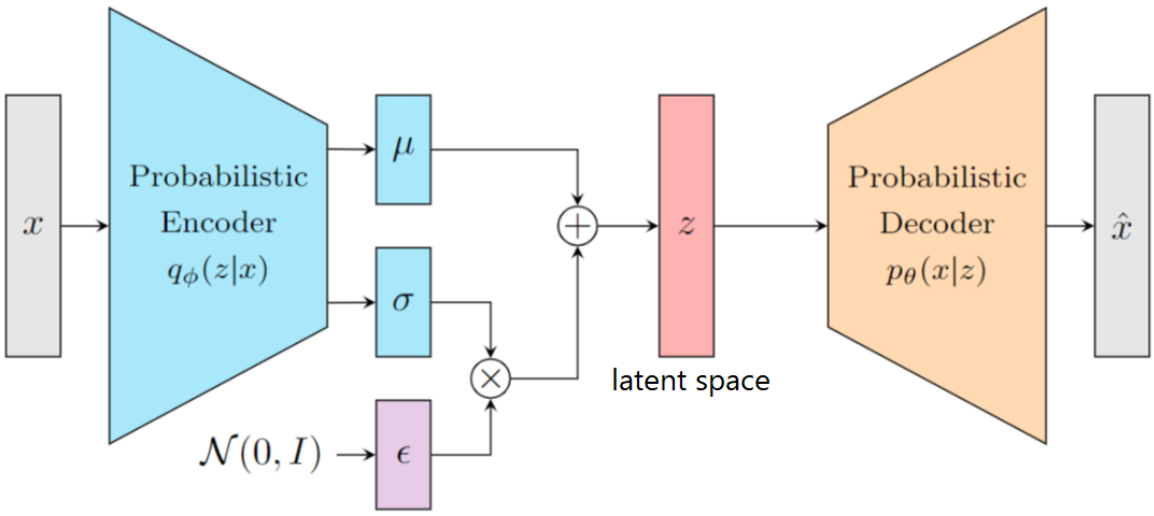

Figure 2 VAE architecture

The goal of a VAE is to deduce the probability distribution of the latent space, p(z), given the probabilistic decoder function (or the probability distribution that projects the data into latent space), p(z|x). By using Bayes' theorem, we conclude that such a computation is not possible, because:

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) \mathrm{d}z$$

The equation is intractable, because we would need to evaluate this integral over all latent variables ***z*** and

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

we don't have a ground truth value for p(z|x), which is also our goal.

In order to compute a good approximation for the intractable integral, we substitute the intractable true posterior with an approximate posterior q parametrized by ϕ: pθ(z|x) ≈ qϕ(z|x). The Kullback-Leibler (KL) divergence formula illustrated in Equation 2 is used to measure how close the approximated posterior is to the true posterior. It is a non-symmetric and always positive. It is equal to 0 only when pθ = qϕ.

$$DKL(p\theta||q\phi) = \sum p\theta(x)\, log\, \frac{p\theta(x)}{q\phi(x)}$$

Equation 2 KL divergence formula

To derive the initial formula, we derive the initial probability distribution of the data

$$\log p_\theta(\boldsymbol{x}) = \log p_\theta\ (\boldsymbol{x})$$

$$= \log p_\theta\ (\boldsymbol{x}) \int q_\varphi(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z}$$

$$= \int \log p_\theta\ (\boldsymbol{x})\, q_\varphi(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z}$$

$$= E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta\ (\boldsymbol{x})]$$

$$= E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})}\right]$$

$$= E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z}) q_\varphi(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z}|\boldsymbol{x}) q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\right]$$

$$= E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\right] + E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z}|\boldsymbol{x})}\right]$$

$$= E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\right] + D_{KL}\left(q_\varphi(\boldsymbol{z}|\boldsymbol{x}) \| p_\theta(\boldsymbol{z}|\boldsymbol{x})\right).$$

The resulted formula consists of 2 terms, the first is the Evidence Lower Bound (ELBO) and the second is the KL divergence, which is always positive. We can infer that

$$\log p_\theta(\boldsymbol{x}) \geq E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\right].$$

By further deriving the resulted formula

$$E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\right] = E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z})}{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\right]$$

$$= E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] + E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z})}{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}\right]$$

$$= E_{q_\varphi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}\left(q_\varphi(\boldsymbol{z}|\boldsymbol{x}) \| p_\theta(\boldsymbol{z})\right)$$

we achieve its final form. In order to maximize the ELBO, the first term, the reconstruction likelihood of the decoder, needs to be maximized, and the second term, the distance between the learned distribution and the prior belief over the latent variable, needs to be minimized. This objective achieves an accurate generative model and an accurate discriminative model. To convert this final form to a loss function that needs to be minimized, the ELBO must be inverted, so that the objective is to minimize the inverted ELBO.

The final loss function is illustrated in Equation 3 and the final computable loss function is illustrated in Equation 4.

$$\begin{aligned} L_{\mathrm{VAE}}(\theta, \phi) &= -\log p_\theta(\mathbf{x}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}|\mathbf{x})) \\ &= -\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z})) \\ \theta^*, \phi^* &= \arg\min_{\theta, \phi} L_{\mathrm{VAE}} \end{aligned}$$

Equation 3 VAE loss function

```
def loss_function(x, x_hat, mean, log_var):
        reconstruction_loss = nn.functional.binary_cross_entropy(x_hat, x, reduction='sum')
        KLD = -0.5 * torch.sum(1 + log_var - mean.pow(2) - log_var.exp())

        return reconstruction_loss + KLD
```

Equation 4 VAE PyTorch loss function

1.2.1. Training VAEs

Being an unsupervised learning task, data is uncategorized and is not required to have annotations. Given that, the only necessary set of data is the training dataset, on which the loss is evaluated. Because the loss is composed of 2 terms, different weighing produces different results (i.e. if a better regularization of the latent space is wanted, the KL divergence term should have a weight greater than the reconstruction loss). In addition, VAE can be used as a stand-alone generative model, or can be used for various downstream tasks. It is important to strike a good balance between the VAE loss function and other loss functions that represent the downstream tasks. The training process represents a very complex task that is very context dependent and that is influenced by the desired output.

2. Disentangled Representations

Disentangled representation learning is a concept in machine learning that refers to the process of learning a representation of data where the individual factors of variation in the data are captured by separate, distinct elements of the representation. In our case, the representation is portrayed by the latent space z of the VAE. Real-world data is often complex and high-dimensional, with many underlying factors that can vary independently. Disentangled representations aim to separate these factors, making it easier for models to understand the structure of the data and how different factors contribute to it. This can lead to more interpretable models, better generalization to new scenarios, and improved performance in tasks such as clustering, transfer learning and data generation. By carefully designing and constraining the VAE's architecture and loss function, it's possible to encourage the model to learn a disentangled representation. An illustration of a perfect disentangled representation that features samples from the 3dshapes dataset is illustrated in Figure 3.

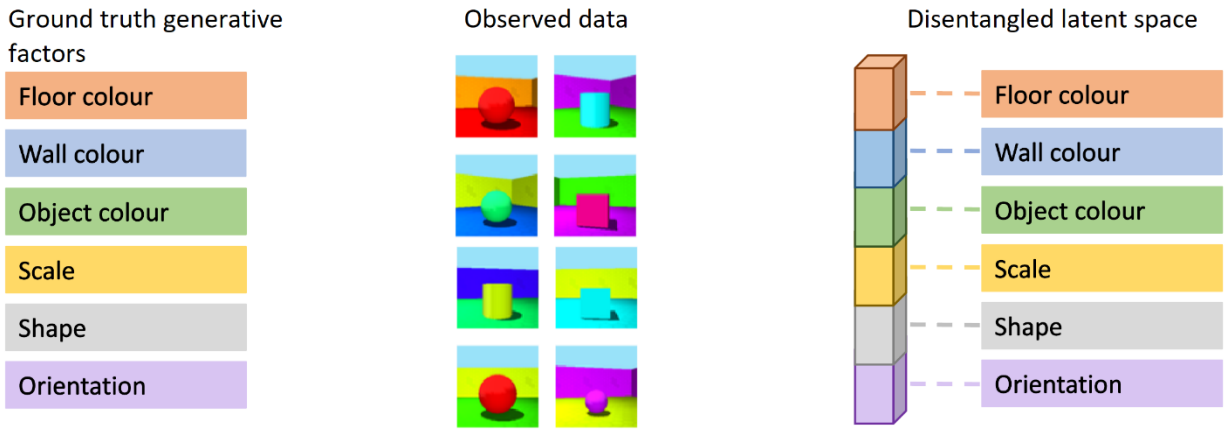

Figure 3 A disentangled latent space that is correlated to the ground truth generative factors of the observed data

2.1. How disentangled representations are measured

Because there is no universally accepted definition of disentanglement, many past metrics fail to generalize and focus only on 1 or neither of the 2 important fundamental properties of disentanglement that are illustrated in Figure 4.

- Property 1. A metric gives a high score to all representations that satisfy the characteristic that the metric reflects.

- Property 2. A metric gives a low score for all representations that do not satisfy the characteristic that the metric reflects.

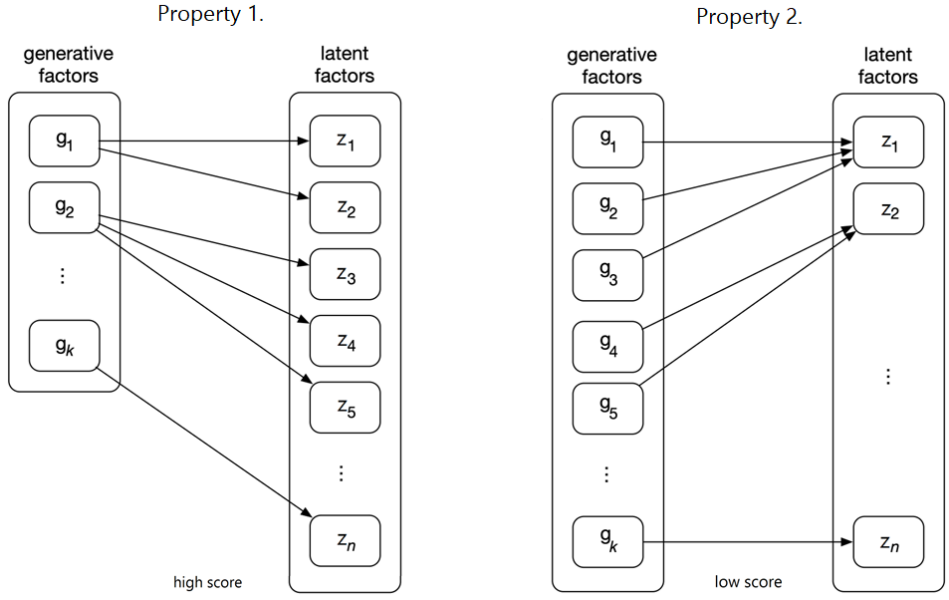

Figure 4 Properties of disentanglement

After analyzing current disentanglement metrics, the authors of the paper "Hot to Not Measure Disentanglement" [2] discovered that the quantitative evaluations do not correspond to the qualitative overview. BetaVAE, FactorVAE, DCI and SAP metrics do not satisfy either property, whereas MIG satisfies both. They further proposed 3CharM, a new metric, but is has not been referenced yet in recent papers.

2.2. Mutual Information Gap (MIG) [3]

The general idea of this metric is to measure the difference between the top two latent variables z with highest mutual information (MI) in Equation 5 in respect to the entropy of their ground truth generative factors v in Equation 6.

$$I_n(z_j; v_k) = \mathbb{E}_{q(z_j, v_k)} \left[\log \sum_{n \in \mathcal{X}_{v_k}} q(z_j|n) p(n|v_k) \right] + H(z_j)$$

Equation 5 normalized MI formula

$$H(v_k) = \mathbb{E}_{p(v_k)}[-\log p(v_k)]$$

Equation 6 Entropy formula

A high MI implies that a latent variable is strongly bound to a ground truth generative factor. The MI is maximized when inverting the roles of z and v gives the same output. Because a single ground truth generative factor can have high MI with multiple latent variables, axis-alignment is enforced by measuring the difference between the top two latent variables with highest MI. K is the number of known ground truth generative factors in the final MIG formula illustrated in Equation 7. The MIG is bounded by 0 and 1 and the objective is to maximize this metric. To calculate the MIG, a full pass through the chosen dataset is recommended.

$$\frac{1}{K} \sum_{k=1}^{K} \frac{1}{H(v_k)} \left(I_n(z_{j^{(k)}}; v_k) - \max_{j \neq j^{(k)}} I_n(z_j; v_k) \right)$$

Equation 7 MIG formula

2.3. Datasets

2.3.1. Datasets used for quantitative and qualitative evaluation

Objective generative factors are hard to be determined in real-life scenarios, because a local state of an object cannot be determined without a full understanding of its global state. In addition, people who annotate data can have a subjective view on how the data should be annotated, inducing bias and making the task imprecise. Given that, current datasets that are subject to quantitative evaluation are synthetically computer generated. The top featured datasets are:

- 3dshapes is a dataset of 3D shapes procedurally generated from 6 ground truth independent latent factors published in "Disentangling by Factorising" [4]. These factors are floor colour (10 values), wall colour (10 values), object colour (10 values), scale (8 values), shape (4 values) and orientation (15 values). All possible combinations of these latent variables are present exactly once, generating 480000 total images.

- 3dface is a dataset including the pair of 2D face image and its corresponding 3D face geometry model published in “CNN-based Real-time Dense Face Reconstruction with Inverse-rendered Photo-realistic Face Images” [5]. The ground truth factors are azimuth, elevation and lighting.
- 3dcars is a dataset of 199 car models published in “Advances in Neural Information Processing Systems” [6]. For each car model, the authors generated color renderings from 24 rotation angles each offset by 15 degrees, as well as from 4 different camera elevations.
- dSprites is a dataset of 2D shapes published as miscellaneous data [7]. The shapes are procedurally generated from 6 ground truth independent latent factors. These factors are color, shape, scale, rotation, x and y positions of a sprite. All possible combinations of these latents are present exactly once, generating 737280 total images. A sprite is a type of "stand-alone" computer graphic element that has evolved along with modern computer graphics technologies. A sprite is defined as a two-dimensional image or animated image that plays a specific role, often independently manipulated, within a larger image environment.

2.3.2. Datasets used for qualitative evaluations

Although the goal of learning tasks is to improve upon current solutions by quantitative comparisons, the particular task of visual disentanglement learning also depends on how disentanglement visually affects images. All of the above-mentioned datasets are synthetically generated and serve the goal to be a stepping stone for future in-the-wild disentangled datasets, in my opinion. For qualitative results that are closer to real-world data with more complex generative ground truth factors, the celebA dataset is used. It was launched in “Deep Learning Face Attributed in the Wild” [8]. It is a large-scale face attributes dataset with more than 200000 celebrity images of 10177 different identities. The images are not annotated with ground truth generative factors but they cover large pose variations and background clutter.

3. Successful attempts at achieving disentanglement

3.1. “Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework” [9]

Published as a conference paper at ICLR 2017, Google DeepMind engineers introduced a new state-of-the-art framework that automatically discovers interpretable factorized latent representations from raw data in a fully unsupervised manner. They demonstrate that by appropriately weighing the KL divergence term of the loss function with of a VAE with beta > 1 it qualitatively outperforms VAE (beta = 1), as well as other state-of-the art models, like unsupervised InfoGAN model and semi-supervised DC-IGN. For beta > 1, the constraint on the latent representation is higher, a higher beta encourages more efficient encodings but it reduces the representation capacity of z. In addition to the qualitative improvements, the model also allows for stable training, by only needing the beta hyperparameter tuned. The weighing of the beta term depends on the training datasets (i.e. beta=250 for celebA, beta=20 on 3D faces and beta=5 for 3D Chairs [10]). The qualitative results are obtained by choosing latent variables and traversing them over a given range. The discovered generative factors for the celebA dataset are rotation, smile, fringe, hair parting, background, skin colour, saturation, age/gender, sunglass/smile. The final loss function is illustrated in Equation 8. A disentanglement metric was also introduced in this paper, but it has been proven that is biased and does not satisfy the 2 disentanglement properties.

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

Equation 8 beta-VAE loss function

3.2. Beta-TCVAE –" Isolating Sources of Disentanglement in Variational Autoencoders" [3]

Published in NeurIPS 2018, the authors of this paper further decompose the KL divergence term that can be used to explain success of beta-VAE and obtain a new state-of-the-art framework. In addition, they propose a stochastic minibatch weighted training schedule and introduce the MIG metric. The decomposition of the KL divergence term comes as follows and illustrated in Equation 9:

i. Index-Code MI – MI between the data variable and latent variable
ii. Total Correlation (TC) – the dependence between variables, forces the model to find statistically independent factors in the data distribution
iii. Dimension-wise KL – mainly prevents individual latent dimensions from deviating too far from their corresponding priors

$$\mathbb{E}_{p(n)}\Big[\mathrm{KL}\big(q(z|n)||p(z)\big)\Big] = \underbrace{\mathrm{KL}\left(q(z,n)||q(z)p(n)\right)}_{\text{(i) Index-Code MI}} + \underbrace{\mathrm{KL}\big(q(z)||\prod_j q(z_j)\big)}_{\text{(ii) Total Correlation}} + \underbrace{\sum_j \mathrm{KL}\left(q(z_j)||p(z_j)\right)}_{\text{(iii) Dimension-wise KL}}$$

Equation 9 KL divergence decomposition

The TC term is the most important term in this decomposition and is the only penalized term in the end loss function in Equation 10. Ablation experiments with different values for α, β and γ were conducted, but the best results were produced with α = γ = 1 and β > 1.

$$\mathcal{L}_{\beta-\mathrm{TC}} := \mathbb{E}_{q(z|n)p(n)}[\log p(n|z)] - \alpha I_q(z;n) - \beta\,\mathrm{KL}\big(q(z)||\prod_j q(z_j)\big) - \gamma\sum_j \mathrm{KL}\left(q(z_j)||p(z_j)\right)$$

Equation 10 beta-TCVAE loss function

In addition to the beta-VAE discovered generative factors for the celebA dataset, the paper further discovers following: dramatic masculinity, contrast, glasses/mustache, shadow/smile, gender, skin colour, brightness, bangs, hue, face width, eye shadow. By using the MIG metric, the quantitative improvement upon beta-VAE is remarkable.

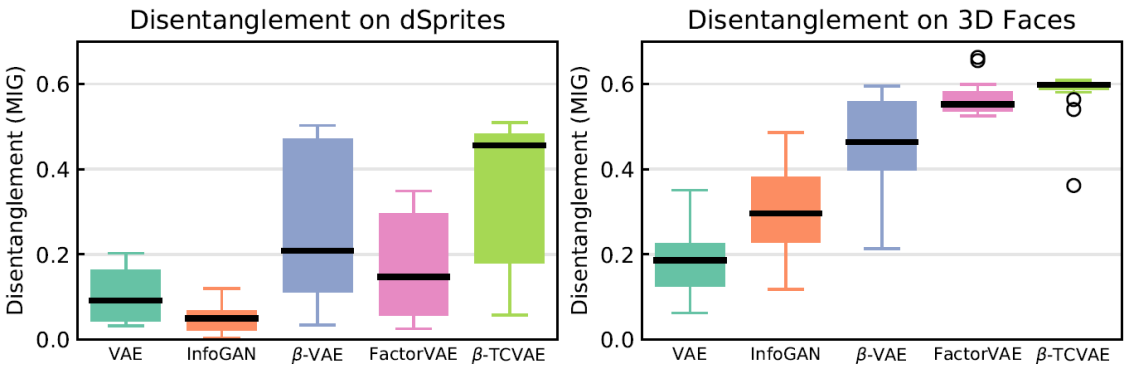

Figure 5 Distribution of disentanglement score (MIG) for different modeling algorithms

4. Downstream tasks where disentangled representations help - examples

- Image generation – get rid of the encoder, sample from a normal distribution, manipulate the latent space
- Anomaly detection – feeding data that differs from the training data would result in an unusual distribution of latent variables
- Dimensionality reduction for downstream tasks
- Generative models for reinforcement learning – I studied “Learning disentangled skills for hierarchical reinforcement learning through trajectory autoencoder with weak labels” [11] for my “Systems of Intelligent Agents” class
- Stable diffusion (+ U-net + an option text encoder)

References

[1] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *ICLR,* 2014.

[2] S. Anna, K. Julia and d. R. Maarten, *How to Not Measure Disentanglement,* ArXiv, 2021.

[3] D. R. T. Q. Chen, X. Li, R. Grosse and D. Duvenaud, *Isolating Sources of Disentanglement in Variational Autoencoders,* arXiv, 2019.

[4] H. Kim and A. Mnih, "Disentangling by Factorising," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[5] G. Yudong, Z. Juyong, C. Jianfei, J. Boyi and Z. Jianmin, "CNN-based Real-time Dense Face Reconstruction with Inverse-rendered Photo-realistic Face Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 41, no. 6, pp. 1294-1307, 2019.

[6] S. E. Reed, Y. Zhang, Y. Zhang and H. Lee, "Deep Visual Analogy-Making," in *Advances in Neural Information Processing Systems*, 2015.

[7] M. Loic, H. Irina, H. Demis and L. Alexander, *dSprites: Disentanglement testing Sprites dataset,* https://github.com/deepmind/dsprites-dataset/: Google Deepmind, 2014.

[8] Z. Liu, P. Luo, X. Wang and X. Tang, "Deep Learning Face Attributes in the Wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[9] H. Irina, M. Loic, P. Arka, B. Christopher, G. Xavier, B. Matthew, M. Shakir and L. Alexander, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *International conference on learning representations*, 2018.

[10] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell and J. Sivic, "Seeing 3D Chairs: Exemplar Part-Based 2D-3D Alignment Using a Large Dataset of CAD Models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[11] W. Song, S. Jeon and H. Choi, "Learning disentangled skills for hierarchical reinforcement learning through trajectory autoencoder with weak labels," *Expert Systems with Applications,* vol. 230, no. 120625, 2023.