PVAE: Learning Disentangled Representations with Intrinsic Dimension via Approximated L_0 Regularization

Chengzhi Shi* CSHI19@JHU.EDU
Whiting School of Engineering, Johns Hopkins University, Baltimore, USA

Ben Glocker B.GLOCKER@IMPERIAL.AC.UK
Daniel C. Castro DC315@IMPERIAL.AC.UK
Department of Computing, Imperial College London, London SW7 2AZ, UK

Abstract

Many models based on the Variational Autoencoder are proposed to achieve disentangled latent variables in inference. However, most current work is focusing on designing powerful disentangling regularizers, while the given number of dimensions for the latent representation at initialization could severely influence the disentanglement. Thus, a pruning mechanism is introduced, aiming at automatically seeking for the intrinsic dimension of the data while promoting disentangled representations. The proposed method is validated on MPI3D and MNIST to be advancing state-of-the-art methods in disentanglement, reconstruction, and robustness. The code is provided on the https://github.com/WeyShi/FYP-of-Disentanglement.

Keywords: Disentanglement, Pruning, Intrinsic Dimension, Variational Autoencoders

1. Introduction

To advance disentanglement, models based on the Variational Autoencoder (VAE) (Kingma and Welling, 2014) are proposed in terms of additional disentangling regularizers. However, in this paper, we introduce an orthogonal mechanism that is applicable to most state-of-the-art models, resulting in higher disentanglement and robustness for model configurations—especially the choice of dimensionality for the latent representation.

Intuitively, both excessive and deficient latent dimensions can be detrimental to achieving the best disentangled latent representations. For excessive dimensions, powerful disentangling regularizers, like the β-VAE (Higgins et al., 2017), can force information to be split across dimensions, resulting in capturing incomplete features. On the other hand, having too few dimensions inevitably leads to an entangled representation, such that each dimension could capture enough information for the subsequent reconstruction.

2. Methods

In this paper, we introduce an approximated L_0 regularization (Louizos et al., 2018) to prune the dimension of the latent representation vector. Consequently, our Pruning Variational Autoencoders (PVAE) framework is applicable to most state-of-the-art VAE-based models due to its orthogonality with current approaches. But in this challenge, we choose to put

* Work done while at Imperial College London.

the pruning mechanism onto the DIP-VAE (for Disentangled Inferred Prior VAE) (Kumar et al., 2018) due to its decent performance on MPI3D (Gondal et al., 2019). In the context of pruning, the aim of L_0 is to compress the network, while here the goal is seeking for the intrinsic dimension for the latent representation, which is achieved by the balance between several terms.

2.1. The Masked Base Model: Masked DIP-VAE

Basically, we desire to achieve binary masks $\boldsymbol{m}$, depending on some learnable parameters $\boldsymbol{\alpha}$, to control each dimension. Thus, the DIP-VAE loss term with masks can be formulated as follow:

$$\begin{aligned}\mathcal{L}_{\text{DIP}}(\theta, \phi, \boldsymbol{\alpha}) = \mathbb{E}_{p(\boldsymbol{x})}\left[-\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{m} \circ \boldsymbol{z})\right] + \sum_i m_i D_{\text{KL}}(q_\phi(z_i|\boldsymbol{x})\|p(z_i))\right] \\ + \lambda_{\text{od}} \sum_{i \neq j} C_{ij}^2 + \lambda_{\text{d}} \sum_i (C_{ii} - 1)^2,\end{aligned} \tag{1}$$

where $\boldsymbol{x}$, $p(\boldsymbol{x})$, $\boldsymbol{z}$, and $p(\boldsymbol{z})$ are the input images, the data distribution, the latent variables (the output of the encoder), and their prior, respectively, and $\boldsymbol{\mu}_\phi(\cdot)$, and $p_\theta(\cdot)$, $q_\phi(\boldsymbol{z}|\cdot)$ denote the function of the encoder's mean path, the decoder, and the encoder. Meanwhile, $\boldsymbol{C} = \text{Cov}_{p(\boldsymbol{x})}[\boldsymbol{m} \circ \boldsymbol{\mu}_\phi(\boldsymbol{x})]$ denotes the covariance matrix of the pruned mean representations.

There are two points to note about the Kullback–Leibler (KL) divergence terms. Firstly, they decompose across dimensions (z_i) because we assumed factorized prior and variational posterior distributions. Secondly, the KL term for each dimension is multiplied by the mask for consistency when that dimension is forced to zero, which can be understood in terms of inference with spike-and-slab distributions (see Louizos et al., 2018, Appendix A).

2.2. Approximate L_0 Regularization

With a second term denoting L_0 regularization over $\boldsymbol{e}$, the samples drawn from the $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, the total loss can be formulated as

$$\mathcal{L}_{\text{total}}(\theta, \phi, \boldsymbol{\alpha}) = \mathcal{L}_{\text{DIP}}(\theta, \phi, \boldsymbol{\alpha}) + \tau \mathbb{E}_{p(\boldsymbol{m}|\boldsymbol{\alpha})}\Big[\mathbb{E}_{p(\boldsymbol{x})}\big[\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\|\boldsymbol{m} \circ \boldsymbol{z}\|_0\right]\big]\Big], \tag{2}$$

where $\|\boldsymbol{m} \circ \boldsymbol{z}\|_0 = \sum_{j=1}^{|\boldsymbol{z}|} \mathbb{I}[m_j z_j \neq 0] = \sum_{j=1}^{|\boldsymbol{z}|} m_j$.

To solve the difficulty of L_0 computation, the L_0 loss is reformulated as $\sum_{j=1}^{|\boldsymbol{\alpha}|}(1 - p(m(\alpha_j) \leq 0|\alpha_j))$, which is the sum of the probability of m_j being positive.

$$\begin{aligned}\mathcal{L}_{\text{total}}(\theta, \phi, \boldsymbol{\alpha}) &= \mathcal{L}_{\text{DIP}}(\theta, \phi, \boldsymbol{\alpha}) + \tau \sum_{j=1}^{|\boldsymbol{\alpha}|} p(m_j > 0|\alpha_j), \\ p(m_j > 0|\alpha_j) &= \text{sigmoid}\Big(\log \alpha_j - \beta \log \frac{-\gamma}{\zeta}\Big),\end{aligned} \tag{3}$$

where $\gamma < 0$ and $\zeta > 1$ are the lower and upper bounds for the stretched range, and β is the temperature coefficient of the masks generation process introduced in Section 2.3. The given formulation is slightly different from Louizos et al. (2018) for clarity. The mask vector $\boldsymbol{m}$ is clamped such that $m_j \in [0, 1]$.

2.3. Realization of Pruning for VAE: The L0Pair Layer

The binary masks $\boldsymbol{m}$ are modelled as following Bernoulli distributions with parameters $\boldsymbol{\alpha}$: $m_i \sim \text{Bern}(\alpha_i)$. Louizos et al. (2018) proposed to obtain these masks in a differentiable fashion, feeding uniform random variables through a sigmoid-like function whose location depends on $\boldsymbol{\alpha}$. Furthermore, to ensure that masks are likely to be exactly 0 or 1, they stretch the value range of the sigmoid-like function to be $[\zeta, \gamma]$ and then clamp it to be $[0, 1]$. This process can be formulated as below, and is illustrated in Appendix A:

$$\begin{aligned} u_i \sim \mathcal{U}(0,1), \quad s_i = \text{sigmoid}((\log u_i - \log(1-u_i) + \log \alpha_i)/\beta), \\ \bar{s}_i = s_i(\zeta - \gamma) + \gamma, \quad m_i = \min(1, \max(0, \bar{s}_i)). \end{aligned} \tag{4}$$

To align it with VAE, we need the encoder to output means $\boldsymbol{\mu}_\phi(\boldsymbol{x})$ and variances $\boldsymbol{\sigma}^2_\phi(\boldsymbol{x})$ of $q(\boldsymbol{z}|\boldsymbol{x})$ instead of means $\boldsymbol{\mu}_\phi(\boldsymbol{x})$ and $\log \boldsymbol{\sigma}^2_\phi(\boldsymbol{x})$ such that after pruning we have a $\mathcal{N}(0,0)$ rather than $\mathcal{N}(0,1)$ for a specific dimension. In detail, a mask is multiplied with each pair of mean and variance and the KL divergence for the corresponding dimension, such that dimensions can effectively be 'switched off' and not affect training. To avoid numerical instability in the KL divergence, we add a small positive constant to $\boldsymbol{\sigma}^2_\phi(\boldsymbol{x})$. Given the outputs of the last layer of the original encoder, the L0Pair layer can be expressed as

$$f^{\text{L0Pair}}_\phi(\boldsymbol{x}; \boldsymbol{m}) = [\boldsymbol{m} \circ \boldsymbol{\mu}_\phi(\boldsymbol{x}); \boldsymbol{m} \circ \boldsymbol{\sigma}^2_\phi(\boldsymbol{x})]. \tag{5}$$

3. Experiments

In terms of the structure of the encoder and the decoder, we adopt the default settings given in the starter kit[1], which is based on row 3 of Table 1 on page 13 of Higgins et al. (2017). We list our choices of hyperparameters in Appendix B.

The L_0 regularization in the pruning mechanism facilitates the performance and the robustness of vanilla DIP-VAE on MPI3D (Gondal et al., 2019) by approaching the intrinsic dimension during training. In Appendix B, we additionally present results on MNIST (LeCun and Cortes, 2010) with a JointVAE (Dupont, 2018) extension of the proposed PVAE (PJVAE), which further validates the disentanglement benefits of pruning.

4. Conclusion

A pruning mechanism that is complementary to most current state-of-the-art VAE-based disentangling models is introduced and validated on MPI3D and MNIST. The approximated L_0 regularization facilitates the model to capture better-disentangled representations with optimal size and increases the robustness to initialization. Moreover, with the same hyperparameters, the model approaches the intrinsic dimension for several datasets including MNIST and MPI3D, even with an extra-large number of dimensions at initialization. Even given the intrinsic dimension, the PVAE still outperforms other SOTA methods in terms of disentanglement and reconstruction.

1. https://github.com/google-research/disentanglement_lib/blob/master/disentanglement_lib

References

Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2018)*, pages 708–718, 2018.

Muhammad Waleed Gondal, Manuel Wüthrich, Đorđe Miladinović, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. *arXiv e-prints*, art. arXiv:1906.03292, Jun 2019.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6114.

Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1kG7GZAW.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through L0 regularization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1Y8hhg0b.

Appendix A. Mask Generation Details

The main process of calculating masks is applying the formula

$$s = \text{sigmoid}((\log u - \log(1-u) + \log \alpha)/\beta), \tag{6}$$

where α, β are the parameters that control the position and the extent of the approximation to pulse function. During training, the model will adjust α only, which can be interpreted as the π of the Bernoulli distribution. As we can see Figure 1

When α increases, the function moves towards left, enabling more x area to produce non-zero output. Thus, α is learned to decide how many pairs are activated. According to the experiments, even initially we set 64 or 32 pairs for MNIST, our pruning VAE can prune it to around 16 with the same hyperparameters.

Another parameter, the temperature β, of the function is set to be a constant here. The effect is shown in Figure 2

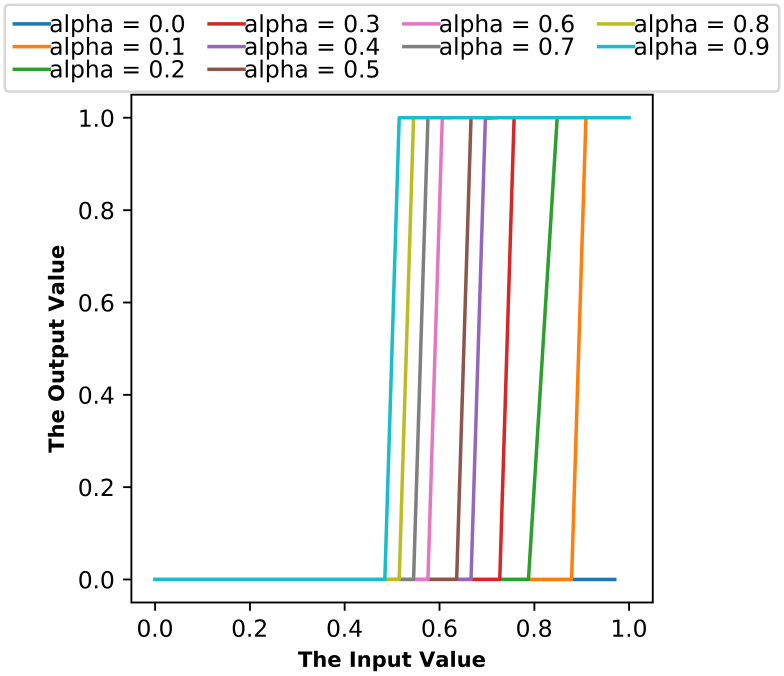

Figure 1: The Role of α

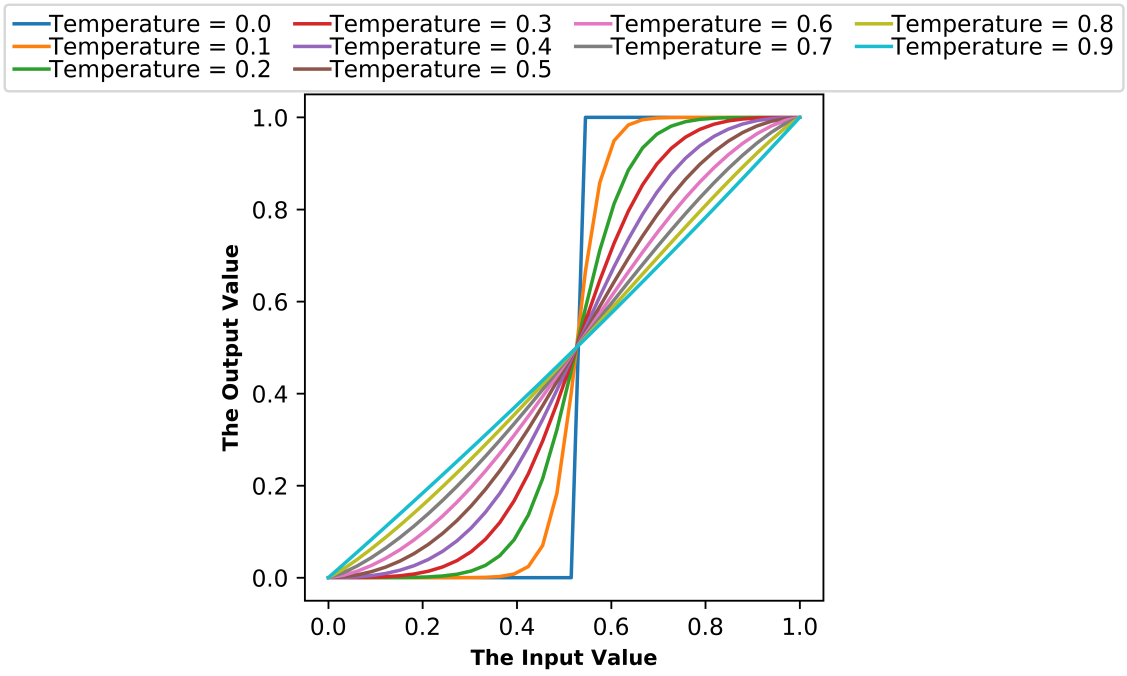

Figure 2: The Role of β

Table 1: Default values of parameters

Parameters	Default Values
β	0.05
τ	0.1
λ_{od}	20
λ_{d}	2
ζ	1.1
γ	-0.1

Appendix B. Further Experiments on MNIST

The default parameters are given in Table 1.

As for the optimizer and its Learning rate, we select Adam optimizer with 10^{-4}. Moreover, the $\tau = 0.1$ generalizes well on both MNIST and CelebA[2]. To capture the discrete features like Digits (Dupont, 2018), we adopt one additional discrete variable and the model becomes Pruning Joint VAE (PJVAE). Since there is only one discrete variable, it is unnecessary to impose further disentanglement on it (the disentanglement on discrete variables is beyond the scope of this report).

In Figure 3, we can see the advantage of pruning on MNIST, especially when the initialization far deviates from the intrinsic dimension (which is still unknown for MNIST, but is estimated to be around 10 by several methods). However, the PJVAE is robust to the initialization as long as it is given enough latent space at initialization.

Surprisingly, with appropriate initialization, its reconstruction occasionally becomes better than the VAE, with consistent higher disentanglement performance. Furthermore, on this dataset PJVAE outperforms DIP-VAE in both metrics. Inspecting the variation between different initialization, we can validate the robustness of PJVAE versus the other two methods.

In general, in terms of TC, PJVAE possesses obvious advantages. And reconstruction performance is the same, PJVAE also showing a consistent lower error. Note that both VAE and DIP-VAE are initialized with one additional 10-value categorical (discrete) variable for a fair comparison. The only difference between this DIP-VAE (actually, DIP-JointVAE) and PJVAE, is the approximated L_0.

2. http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

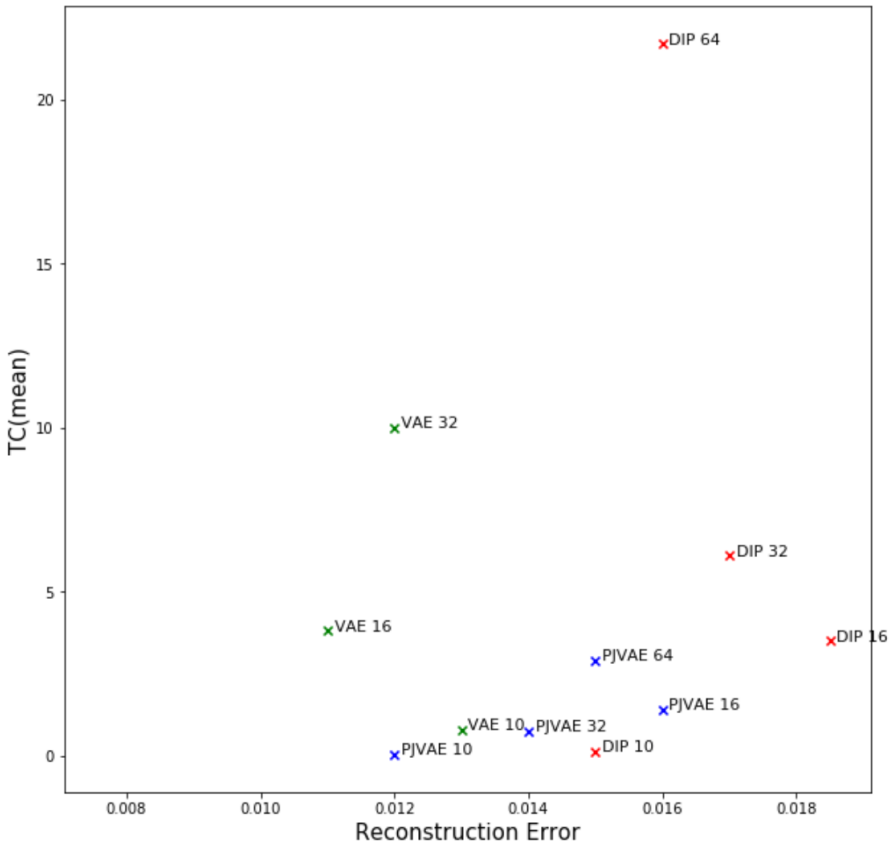

Figure 3: Comparison between VAE, DIP-VAE, and PJVAE with different initialization on MNIST. The number denotes the total dimensionality of the latent variables at initialization. TC stands for Total Correlation.