β-VAE

Benjamin Levy

December 10, 2020
STAT185

1 Introduction

A β-variational autoencoder (β-VAE) is a family of generative models whose goal is to infer a "disentangled," low-dimensional representation of some high-dimensional data, often images [4]. In general, auto-encoders seek to reconstruct some input $\vec{x} \in \mathbb{R}^d$ while simultaneously learning a latent representation, $\vec{z} \in \mathbb{R}^k$, which is typically chosen to be lower-dimensional than $\vec{x}$. This is accomplished by training two models: an encoder parameterized by ϕ, denoted $g_\phi(\vec{x})$, and a decoder parameterized by θ, denoted $f_\theta(\vec{z})$. The algorithm is, roughly:

1. Given example $\vec{x}$, compute $\vec{z} = g_\phi(\vec{x})$
2. Compute reconstruction $\vec{x}' = f_\theta(\vec{z})$

A limitation of this deterministic autoencoder is that the latent space in which $\vec{z}$ resides is typically quite sparse and/or irregular. This can make it difficult to use the latent representations for inference or other purposes. Variational auto-encoders (VAEs) improve upon this by replacing the decoder f_θ with a stochastic sampling function. Consequently, g_ϕ now outputs a mean $\vec{\mu}$ and a covariance Σ, which parameterize a Gaussian distribution [2]. The algorithm then becomes:

1. Given example $\vec{x}$, compute $(\vec{\mu}, \Sigma) = g_\phi(\vec{x})$
2. Sample latent variable $\vec{z} \sim \mathcal{N}(\vec{\mu}, \Sigma)$
3. Compute reconstruction $\vec{x}' = f_\theta(\vec{z})$

To make this model tractable, we typically use isotropic Gaussians, meaning the covariance matrix Σ is diagonal. By replacing the deterministic algorithm with a stochastic one, the model is forced to learn latent representations that are more robust to Gaussian noise, since if small perturbations led to dramatically different outputs, the model would perform poorly at reconstructing the output. The loss function for the VAE can be expressed as a sum of two terms: (1) reconstruction error and (2) regularization (defined more rigorously later on in the methods section). The reconstruction term pushes the model to faithfully reconstruct the original data, while the regularization term encourages the model to learn a generalizable and efficient latent representation.

While the VAE model solves the problem of irregular latent spaces, there are no guarantees that the latent representations are *meaningful*. This gives rise to the idea of *disentanglement*, which is roughly the notion that the different components of the latent variable $\vec{z}$ should correspond to distinct *generative factors* (i.e. the parameters of the generative process) [1]. For instance, if we are modelling an image dataset consisting of human faces, we might want one dimension to correspond to skin tone, another to correspond to eye shape, a further one to correspond to amount of hair, and so on. Introduced by Burgess et al. in 2018, the β-VAE is a model designed with this goal in mind [3]. An additional hyperparameter, $\beta \in \mathbb{R}$ is added as a coefficient for the regularization term from the VAE loss. When $\beta = 1$, the loss function is identical to the original VAE loss function. When $\beta > 1$, the model is forced to learn a less expressive, and therefore more information-dense, latent encoding for the data. Furthermore, the representations learned by β-VAE tend to be more disentangled, meaning that it is more possible to recover the original factors that may have generated the data.

2 Methods

2.1 Variational auto-encoders

The goal of a VAE is to learn a distribution p_θ to model our data $\vec{x}$ and its hypothesized latent variables $\vec{z}$. We can arrange related $\vec{x}$ and $\vec{z}$ using Bayes' rule:

$$p_\theta(\vec{x}|\vec{z}) = \frac{p_\theta(\vec{z}|\vec{x})p_\theta(\vec{x})}{p_\theta(\vec{z})}$$

Under this framework, we want to maximize the log-probability of data generated according to the process: (1) sample a point $\vec{z} \sim p_\theta(\vec{z})$, then sample $\vec{x} \sim p_\theta(\vec{x}|\vec{z})$:

$$\ell = \max_\theta \sum_{i=1}^{N} \log p_\theta(\vec{x}_i)$$

Marginalizing over all values of $\vec{x}$,

$$= \max_\theta \sum_{i=1}^{N} \log \int p_\theta(\vec{x}_i|\vec{z})p_\theta(\vec{z})d\vec{z}$$

We now introduce an auxiliary distribution, q_ϕ. This will serve two purposes: (1) allow us to massage this expression into one involving a KL-divergence, and (2) enable *amortised* inference (explained later).

$$\ell = \max_\theta \sum_{i=1}^{N} \log \int p_\theta(\vec{x}_i|\vec{z})p_\theta(\vec{z})d\vec{z}$$

3 Discussion

References

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation Learning: A Review and New Perspectives. ArXiv e-prints". In: *arXiv preprint arXiv:1206.5538* (2012).

[2] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[3] Irina Higgins et al. "beta-vae: Learning basic visual concepts with a constrained variational framework". In: (2016).

[4] Lilian Weng. "From Autoencoder to Beta-VAE". In: *lilianweng.github.io/lil-log* (2018). URL: `http://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html`.