Variational autoencoder

In machine learning, a **variational autoencoder (VAE)**, is an artificial neural network architecture introduced by Diederik P. Kingma and Max Welling, belonging to the families of probabilistic graphical models and variational Bayesian methods.[1]

Variational autoencoders are often associated with the autoencoder model because of its architectural affinity, but with significant differences in the goal and mathematical formulation. Variational autoencoders are probabilistic generative models that require neural networks as only a part of their overall structure, as e.g. in VQ-VAE. The neural network components are typically referred to as the encoder and decoder for the first and second component respectively. The first neural network maps the input variable to a latent space that corresponds to the parameters of a variational distribution. In this way, the encoder can produce multiple different samples that all come from the same distribution. The decoder has the opposite function, which is to map from the latent space to the input space, in order to produce or generate data points. Both networks are typically trained together with the usage of the reparameterization trick, although the variance of the noise model can be learned separately.

Although this type of model was initially designed for unsupervised learning,[2][3] its effectiveness has been proven for semi-supervised learning[4][5] and supervised learning.[6]

Contents

Overview of architecture and operation

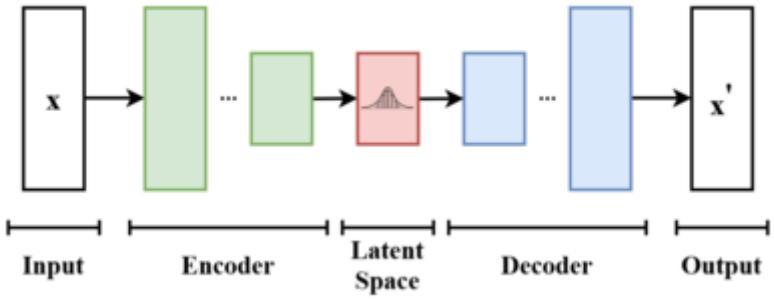
A variational autoencoder is a generative model with a prior and noise distribution respectively. Usually such models are trained using the Expectation-Maximization meta-algorithm (e.g. probabilistic PCA, (spike & slab) sparse coding). Such a scheme optimizes a lower bound of the data likelihood, which is usually intractable, and in doing so requires the discovery of q-distributions, or variational posteriors. These q distributions are normally parameterized for each individual data point in a separate optimization process. However, variational autoencoders use a neural network as an amortized approach to jointly optimize across data points. This neural network takes as input the data points themselves, and outputs parameters for the variational distribution. As it maps from a known input space to the low-dimensional latent space, it is called the encoder.

The decoder is the second neural network of this model. It is a function that maps from the latent space to the input space, e.g. as the means of the noise distribution. It is possible to use another neural network that maps to the variance, however this can be omitted for simplicity. In such a case, the variance can be optimized with gradient descent.

To optimize this model, one needs to know two terms: the "reconstruction error", and the Kullback–Leibler divergence. Both terms are derived from the free energy expression of the probabilistic model, and therefore differ depending on the noise distribution and the assumed prior of the data. The KL-D from the free energy expression maximizes the probability mass of the q distribution that overlaps with the p distribution, which unfortunately can result in mode-seeking behaviour. The "reconstruction" term is the remainder of the free energy expression, and requires a sampling approximation to compute its expectation value.[7]

Formulation

From the point of view of probabilistic modelling, one wants to maximize the likelihood of the data $\boldsymbol{x}$ by their chosen parameterized probability distribution $p(\boldsymbol{x}|\boldsymbol{\theta})$. This distribution is usually chosen to be a Gaussian $N(\boldsymbol{x}|\mu,\sigma)$ which is parameterized by μ and σ respectively, and as a member of the exponential family it is easy to work with as a noise distribution. Simple distributions are easy enough to maximize, however distributions where a prior is assumed over the latents $\boldsymbol{z}$ results in intractable integrals. Let us find $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ via marginalizing over $\boldsymbol{z}$.



The basic scheme of a variational autoencoder. The model receives $\boldsymbol{x}$ as input. The encoder compresses it into the latent space. The decoder receives as input the information sampled from the latent space and produces $\boldsymbol{x}'$ as similar as possible to $\boldsymbol{x}$.

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int_{\boldsymbol{z}} p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})\, d\boldsymbol{z},$$

where $p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})$ represents the joint distribution under $p_{\boldsymbol{\theta}}$ of the observable data $\boldsymbol{x}$ and its latent representation or encoding $\boldsymbol{z}$. According to the chain rule, the equation can be rewritten as

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int_{\boldsymbol{z}} p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z})\, d\boldsymbol{z}$$

In the vanilla variational autoencoder, $\boldsymbol{z}$ is usually taken to be a finite-dimensional vector of real numbers, and $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ to be a Gaussian distribution. Then $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is a mixture of Gaussian distributions.

It is now possible to define the set of the relationships between the input data and its latent representation as

- Prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$
- Likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$
- Posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$

Unfortunately, the computation of $p_\theta(x)$ is expensive and in most cases intractable. To speed up the calculus to make it feasible, it is necessary to introduce a further function to approximate the posterior distribution as

$$q_\phi(z|x) \approx p_\theta(z|x)$$

with ϕ defined as the set of real values that parametrize q. This is sometimes called *amortized inference*, since by "investing" in finding a good q_ϕ, one can later infer z from x quickly without doing any integrals.

In this way, the problem is of finding a good probabilistic autoencoder, in which the conditional likelihood distribution $p_\theta(x|z)$ is computed by the *probabilistic decoder*, and the approximated posterior distribution $q_\phi(z|x)$ is computed by the *probabilistic encoder*.

Evidence lower bound (ELBO)

As in every deep learning problem, it is necessary to define a differentiable loss function in order to update the network weights through backpropagation.

For variational autoencoders, the idea is to jointly optimize the generative model parameters θ to reduce the reconstruction error between the input and the output, and ϕ to make $q_\phi(z|x)$ as close as possible to $p_\theta(z|x)$. As reconstruction loss, mean squared error and cross entropy are often used.

As distance loss between the two distributions the reverse Kullback–Leibler divergence $D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x))$ is a good choice to squeeze $q_\phi(z|x)$ under $p_\theta(z|x)$.[8][9]

The distance loss just defined is expanded as

$$\begin{aligned} D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x)) &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln \frac{q_\phi(z|x)}{p_\theta(z|x)}\right] \\ &= \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln \frac{q_\phi(z|x)p_\theta(x)}{p_\theta(x,z)}\right] \\ &= \ln p_\theta(x) + \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln \frac{q_\phi(z|x)}{p_\theta(x,z)}\right] \end{aligned}$$

Now define the evidence lower bound (ELBO):

$$L_{\theta,\phi}(x) := \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] = \ln p_\theta(x) - D_{KL}(q_\phi(\cdot|x) \parallel p_\theta(\cdot|x))$$

Maximizing the ELBO

$$\theta^*, \phi^* = \operatorname*{argmax}_{\theta,\phi} L_{\theta,\phi}(x)$$

is equivalent to simultaneously maximizing $\ln p_\theta(x)$ and minimizing $D_{KL}(q_\phi(z|x) \parallel p_\theta(z|x))$. That is, maximizing the log-likelihood of the observed data, and minimizing the divergence of the approximate posterior $q_\phi(\cdot|x)$ from the exact posterior $p_\theta(\cdot|x)$.

For a more detailed derivation and more interpretations of ELBO and its maximization, see its main page.

Reparameterization

To efficient search for

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} L_{\theta,\phi}(x)$$

the typical method is gradient descent.

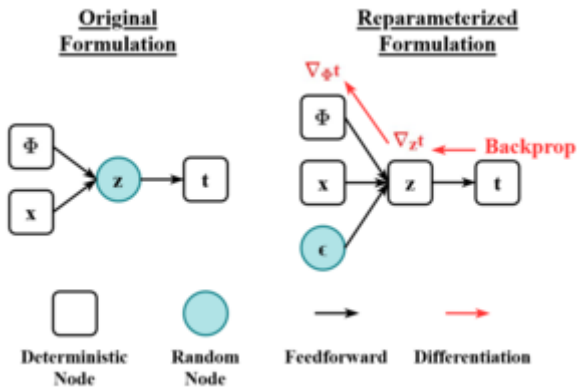
It is straightforward to find

$$\nabla_\theta \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] = \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\nabla_\theta \ln \right.$$

However,

$$\nabla_\phi \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln \frac{p_\theta(x,z)}{q_\phi(z|x)}\right]$$

does not allow one to put the ∇_ϕ inside the expectation, since ϕ appears in the probability distribution itself. The **reparameterization trick** (also known as stochastic backpropagation[10]) bypasses this difficulty.[8][11][12]



The scheme of the reparameterization trick. The randomness variable $\boldsymbol{\varepsilon}$ is injected into the latent space $\boldsymbol{z}$ as external input. In this way, it is possible to backpropagate the gradient without involving stochastic variable during the update.

The most important example is when $z \sim q_\phi(\cdot|x)$ is normally distributed, as $\mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$.

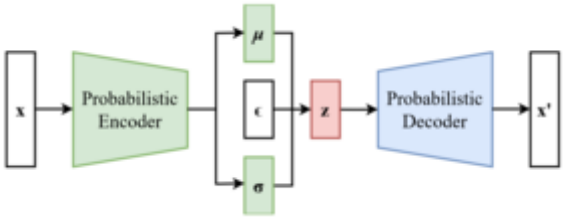
This can be reparametrized by letting $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ be a "standard random number generator", and construct z as $z = \mu_\phi(x) + L_\phi(x)\epsilon$. Here, $L_\phi(x)$ is obtained by the Cholesky decomposition:

$$\Sigma_\phi(x) = L_\phi(x) L_\phi(x)^T$$

Then we have

$$\nabla_\phi \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\ln \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] = \mathbb{E}_\epsilon\left[\nabla_\phi \ln \frac{p_\theta(x,}{q_\phi(\mu_}\right.$$

and so we obtained an unbiased estimator of the gradient, allowing stochastic gradient descent.



The scheme of a variational autoencoder after the reparameterization trick.

Since we reparametrized z, we need to find $q_\phi(z|x)$. Let q_0 by the probability density function for ϵ, then

$$\ln q_\phi(z|x) = \ln q_0(\epsilon) - \ln\left|\det(\partial_\epsilon z)\right|$$

where $\partial_\epsilon z$ is the Jacobian matrix of ϵ with respect to z. Since $z = \mu_\phi(x) + L_\phi(x)\epsilon$, this is

$$\ln q_\phi(z|x) = -\frac{1}{2}\|\epsilon\|^2 - \ln\left|\det L_\phi(x)\right| - \frac{n}{2}\ln(2\pi)$$

Variations

Many variational autoencoders applications and extensions have been used to adapt the architecture to other domains and improve its performance.

β-VAE is an implementation with a weighted Kullback–Leibler divergence term to automatically discover and interpret factorised latent representations. With this implementation, it is possible to force manifold disentanglement for β values greater than one. This architecture can discover disentangled latent factors without supervision.[13][14]

The conditional VAE (CVAE), inserts label information in the latent space to force a deterministic constrained representation of the learned data.[15]

Some structures directly deal with the quality of the generated samples[16][17] or implement more than one latent space to further improve the representation learning.[18][19]

Some architectures mix VAE and generative adversarial networks to obtain hybrid models.[20][21][22]

See also

- Autoencoder
- Artificial neural network
- Deep learning
- Generative adversarial network
- Representation learning
- Sparse dictionary learning
- Data augmentation
- Backpropagation

References

1. Pinheiro Cinelli, Lucas; et al. (2021). "Variational Autoencoder" (https://www.google.com/books/edition/Variational_Methods_for_Machine_Learning/N5EtEAAAQBAJ?hl=en&gbpv=1&pg=PA111). *Variational Methods for Machine Learning with Applications to Deep Networks*. Springer. pp. 111–149. doi:10.1007/978-3-030-70679-1_5 (https://doi.org/10.1007%2F978-3-030-70679-1_5). ISBN 978-3-030-70681-4. S2CID 240802776 (https://api.semanticscholar.org/CorpusID:240802776).
2. Dilokthanakul, Nat; Mediano, Pedro A. M.; Garnelo, Marta; Lee, Matthew C. H.; Salimbeni, Hugh; Arulkumaran, Kai; Shanahan, Murray (2017-01-13). "Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders". arXiv:1611.02648 (https://arxiv.org/abs/1611.02648) [cs.LG (https://arxiv.org/archive/cs.LG)].
3. Hsu, Wei-Ning; Zhang, Yu; Glass, James (December 2017). "Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation" (https://ieeexplore.ieee.org/abstract/document/8268911?casa_token=i8S9DzueB5gAAAAA:SnZUh5mfUYtRpusQLMJxN7eC_-6-qOQs9vpkEcA0Ai_ju-nJH7o1H1DN6nDFdeCY-LgGg3OVKQ). *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. pp. 16–23. arXiv:1707.06265 (https://arxiv.org/abs/1707.06265). doi:10.1109/ASRU.2017.8268911 (https://doi.org/10.1109%2FASRU.2017.8268911). ISBN 978-1-5090-4788-8. S2CID 22681625 (https://api.semanticscholar.org/CorpusID:22681625).

4. Ehsan Abbasnejad, M.; Dick, Anthony; van den Hengel, Anton (2017). *Infinite Variational Autoencoder for Semi-Supervised Learning* (https://openaccess.thecvf.com/content_cvpr_2017/html/Abbasnejad_Infinite_Variational_Autoencoder_CVPR_2017_paper.html). pp. 5888–5897.
5. Xu, Weidi; Sun, Haoze; Deng, Chao; Tan, Ying (2017-02-12). "Variational Autoencoder for Semi-Supervised Text Classification" (https://ojs.aaai.org/index.php/AAAI/article/view/10966). *Proceedings of the AAAI Conference on Artificial Intelligence*. **31** (1). doi:10.1609/aaai.v31i1.10966 (https://doi.org/10.1609%2Faaai.v31i1.10966). S2CID 2060721 (https://api.semanticscholar.org/CorpusID:2060721).
6. Kameoka, Hirokazu; Li, Li; Inoue, Shota; Makino, Shoji (2019-09-01). "Supervised Determined Source Separation with Multichannel Variational Autoencoder" (https://direct.mit.edu/neco/article/31/9/1891/8494/Supervised-Determined-Source-Separation-with). *Neural Computation*. **31** (9): 1891–1914. doi:10.1162/neco_a_01217 (https://doi.org/10.1162%2Fneco_a_01217). PMID 31335290 (https://pubmed.ncbi.nlm.nih.gov/31335290). S2CID 198168155 (https://api.semanticscholar.org/CorpusID:198168155).
7. Kingma, Diederik (2013). "Autoencoding Variational Bayes". *Arxiv*.
8. Kingma, Diederik P.; Welling, Max (2014-05-01). "Auto-Encoding Variational Bayes". arXiv:1312.6114 (https://arxiv.org/abs/1312.6114) [stat.ML (https://arxiv.org/archive/stat.ML)].
9. "From Autoencoder to Beta-VAE" (https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html). *Lil'Log*. 2018-08-12.
10. Rezende, Danilo Jimenez; Mohamed, Shakir; Wierstra, Daan (2014-06-18). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models" (https://proceedings.mlr.press/v32/rezende14.html). *International Conference on Machine Learning*. PMLR: 1278–1286. arXiv:1401.4082 (https://arxiv.org/abs/1401.4082).
11. Bengio, Yoshua; Courville, Aaron; Vincent, Pascal (2013). "Representation Learning: A Review and New Perspectives" (https://ieeexplore.ieee.org/abstract/document/6472238?casa_token=wQPK9gUGfCsAAAAA:FS5uNYCQVJGH-bq-kVvZeTdnQ8a33C6qQ4VUyDyGLMO13QewH3wcry9_Jh-5FATvspBj8YOXfw). *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **35** (8): 1798–1828. arXiv:1206.5538 (https://arxiv.org/abs/1206.5538). doi:10.1109/TPAMI.2013.50 (https://doi.org/10.1109%2FTPAMI.2013.50). ISSN 1939-3539 (https://www.worldcat.org/issn/1939-3539). PMID 23787338 (https://pubmed.ncbi.nlm.nih.gov/23787338). S2CID 393948 (https://api.semanticscholar.org/CorpusID:393948).
12. Kingma, Diederik P.; Rezende, Danilo J.; Mohamed, Shakir; Welling, Max (2014-10-31). "Semi-Supervised Learning with Deep Generative Models". arXiv:1406.5298 (https://arxiv.org/abs/1406.5298) [cs.LG (https://arxiv.org/archive/cs.LG)].
13. Higgins, Irina; Matthey, Loic; Pal, Arka; Burgess, Christopher; Glorot, Xavier; Botvinick, Matthew; Mohamed, Shakir; Lerchner, Alexander (2016-11-04). "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework" (https://openreview.net/forum?id=Sy2fzU9gl).
14. Burgess, Christopher P.; Higgins, Irina; Pal, Arka; Matthey, Loic; Watters, Nick; Desjardins, Guillaume; Lerchner, Alexander (2018-04-10). "Understanding disentangling in β-VAE". arXiv:1804.03599 (https://arxiv.org/abs/1804.03599) [stat.ML (https://arxiv.org/archive/stat.ML)].
15. Sohn, Kihyuk; Lee, Honglak; Yan, Xinchen (2015-01-01). "Learning Structured Output Representation using Deep Conditional Generative Models" (https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf) (PDF).
16. Dai, Bin; Wipf, David (2019-10-30). "Diagnosing and Enhancing VAE Models". arXiv:1903.05789 (https://arxiv.org/abs/1903.05789) [cs.LG (https://arxiv.org/archive/cs.LG)].

17. Dorta, Garoe; Vicente, Sara; Agapito, Lourdes; Campbell, Neill D. F.; Simpson, Ivor (2018-07-31). "Training VAEs Under Structured Residuals". arXiv:1804.01050 (https://arxiv.org/abs/1804.01050) [stat.ML (https://arxiv.org/archive/stat.ML)].
18. Tomczak, Jakub; Welling, Max (2018-03-31). "VAE with a VampPrior" (http://proceedings.mlr.press/v84/tomczak18a.html). *International Conference on Artificial Intelligence and Statistics*. PMLR: 1214–1223. arXiv:1705.07120 (https://arxiv.org/abs/1705.07120).
19. Razavi, Ali; Oord, Aaron van den; Vinyals, Oriol (2019-06-02). "Generating Diverse High-Fidelity Images with VQ-VAE-2". arXiv:1906.00446 (https://arxiv.org/abs/1906.00446) [cs.LG (https://arxiv.org/archive/cs.LG)].
20. Larsen, Anders Boesen Lindbo; Sønderby, Søren Kaae; Larochelle, Hugo; Winther, Ole (2016-06-11). "Autoencoding beyond pixels using a learned similarity metric" (http://proceedings.mlr.press/v48/larsen16.html). *International Conference on Machine Learning*. PMLR: 1558–1566. arXiv:1512.09300 (https://arxiv.org/abs/1512.09300).
21. Bao, Jianmin; Chen, Dong; Wen, Fang; Li, Houqiang; Hua, Gang (2017). "CVAE-GAN: Fine-Grained Image Generation Through Asymmetric Training". pp. 2745–2754. arXiv:1703.10155 (https://arxiv.org/abs/1703.10155) [cs.CV (https://arxiv.org/archive/cs.CV)].
22. Gao, Rui; Hou, Xingsong; Qin, Jie; Chen, Jiaxin; Liu, Li; Zhu, Fan; Zhang, Zhao; Shao, Ling (2020). "Zero-VAE-GAN: Generating Unseen Features for Generalized and Transductive Zero-Shot Learning" (https://ieeexplore.ieee.org/abstract/document/8957359?casa_token=d6k1X5ClbTsAAAAA:AiOSfZQ7S3EsflaecikiuLX8Y9-Lf5FHqTFRjL-FMQQ8bNjdW2rD0UZxA0BC4gVMO0QjF_YXkw). *IEEE Transactions on Image Processing*. **29**: 3665–3680. Bibcode:2020ITIP...29.3665G (https://ui.adsabs.harvard.edu/abs/2020ITIP...29.3665G). doi:10.1109/TIP.2020.2964429 (https://doi.org/10.1109%2FTIP.2020.2964429). ISSN 1941-0042 (https://www.worldcat.org/issn/1941-0042). PMID 31940538 (https://pubmed.ncbi.nlm.nih.gov/31940538). S2CID 210334032 (https://api.semanticscholar.org/CorpusID:210334032).
