Assessing learned representations in variational autoencoders using uncertainty

Daniel Flam-Shepherd and Tony Wu
University of Toronto

Abstract

We describe a useful application of uncertainty in the fully Bayesian variational autoencoder (BVAE), that uses Bayesian neural networks (BNNs) in its encoder and decoder. In this framework, we describe how uncertainty can be used to assess the model's learned representation by interpreting the approximate posterior and probabilistic decoder as predictive distributions This allows us to quantify uncertainty in the learned representation and generative model. We visualize these posterior predictive distributions on a dimensionality reduction task on the swiss roll dataset [1] and in generative modeling of the MNIST dataset [2].

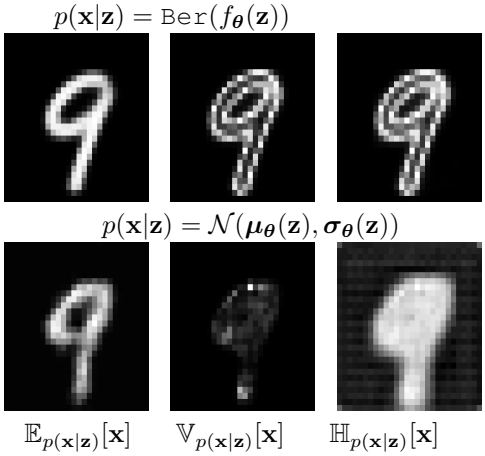


Figure 1: The mean, variance and entropy of a multivariate Bernoulli decoder versus a multivariate Gaussian decoder (in a standard VAE). We condition on a single sample of $\mathbf{z} \sim p(\mathbf{z})$. In both cases the variance and entropy are not very informative. With uncertainty in the decoder we can get useful variance and entropy.

1 Introduction and Motivation

The variational autoencoder (VAE) [3] is a popular generative model pairing a top-down generative network with a bottom-up recognition network which approximates posterior inference. It assumes the posterior distribution factorizes and that its parameters can be functionally approximated by mapping from the observations. Typically, neural networks have been used as these density networks [4], all though Deep Gaussian Processes have also been experimented with [5]. Bayesian neural networks [6, 7] combine the scalability and flexibility of neural networks with uncertainty modeling. Bayesian neural networks have also been used in VAE architectures, but are difficult to train: good results can be obtained by using importance weighting and no priors on the encoder and decoder were found useful or necessary [8], (the original VAEs also do not need any regularization of any kind on the encoder and decoder [3]).

In general, BNNs are more expensive to train and hence are not commonly used in VAEs or other generative models. However, recently [9] propose using the Bayesian variational autoencoder to detect out-of-distribution inputs. In this work, we reveal how using BNN encoders and decoders in VAEs that can help us assess the model's learned representation. Specifically, we highlight the use of the encoder and decoder as *posterior predictive distributions*, and approximate them by sampling the approximate posterior on their respective parameters. Thus, we can get uncertainty estimates for the generative model and encoding distribution. By using a BNN in the decoder we can get model based uncertainty estimates that will tell us what parts of each digit in what region of the latent space, the decoder was unsure about. We can obtain these uncertainty estimates in the decoder for a variety of data types, including high dimensional images.

For low dimensional representations, we can also use uncertainty in the learned latent representation.

2 Background and Related Work

2.1 Bayesian Neural Networks

For a given dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$, to be bayesian in a neural network [10, 11] we place priors on the parameters $p(\mathbf{w})$ of the network $\mathbf{y} = f(\mathbf{x}, \mathbf{w})$. The likelihood $p(\mathcal{D}|\mathbf{w})$ factorizes

$$p(\mathcal{D}|\mathbf{w}) = \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{x}_i, \mathbf{w}) \tag{1}$$

and is categorical for classification or Gaussian for regression. The intractable true posterior is

$$p(\mathbf{w}|\mathcal{D}) = p(\mathcal{D}|\mathbf{w})p(\mathbf{w}) / \int p(\mathcal{D}, \mathbf{w}) d\mathbf{w} \tag{2}$$

Since the marginal likelihood is also intractable we can use variational inference [12] to approximate $p(\mathbf{w}|\mathcal{D})$ with $q_{\boldsymbol{\phi}}(\mathbf{w})$ by maximizing a lower bound $\mathcal{L}(\boldsymbol{\phi})$ on the marginal log-likelihoo $\log p(\mathcal{D}) \geq \mathcal{L}(q)$.

$$\mathcal{L}(q) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{w})} \left[\log \frac{p(D, \mathbf{w})}{q_{\boldsymbol{\phi}}(\mathbf{w})} \right] \tag{3}$$

In mean-field variational inference [12], we specify an approximate posterior that factorizes, for example, the diagonal multivariate Gaussian :

$$q_{\boldsymbol{\phi}}(\mathbf{w}) = \prod_i q_{\boldsymbol{\phi}}(\mathbf{w}_i) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\sigma}^2) \tag{4}$$

then using the reparametrization trick [13] we can differentiate through sampling during training [14]

$$\mathbf{w} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbb{I}) \tag{5}$$

and use stochastic variational inference [15].

2.2 The posterior predictive distribution:

For new data $(\mathbf{x}, \mathbf{y})$, the predictive posterior is $p(\mathbf{y}|\mathbf{x}, \mathcal{D})$ and is also intractable: but can be approximated using our approximate posterior.

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}, \mathcal{D}) &= \int p(\mathbf{y}|\mathbf{x}, \mathbf{w}) p(\mathbf{w}|\mathcal{D}) d\mathbf{w} \\ &\approx \frac{1}{S} \sum_{s=1} p(\mathbf{y}|\mathbf{x}, \mathbf{w}^s) \end{aligned}$$

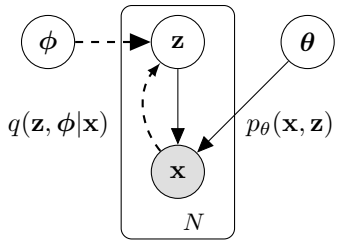


Figure 2: The graphical model of the BVAE. Parameters are global latent variables while $\mathbf{z}$ is local latent variable

where we sample $\mathbf{w} \sim q_{\boldsymbol{\phi}}(\mathbf{w})$ since we are approximating $p(\mathbf{w}|\mathcal{D}) \approx q_{\boldsymbol{\phi}}(\mathbf{w})$ Each possible set of weights from the posterior distribution, makes a prediction about the unknown label given the corresponding $\mathbf{x}$ so we can interpret the predictive posterior as

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}) = \mathbb{E}_{p(\mathbf{w}|\mathcal{D})}[p(\mathbf{y}|\mathbf{x}, \mathbf{w})] \approx \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{w})}[p(\mathbf{y}|\mathbf{x}, \mathbf{w})]$$

being an expectation under the posterior distribution on weights is equivalent to using an infinite ensemble of neural networks. Unfortunately, this is intractable for all neural networks.

2.3 Variational autoencoder

The Variational autoencoder is a latent variable model with a bottom up recognition network paired with a top down generator network. Instead of optimizing the intractable marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z}$ to learn the true posterior $p(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})/p_\theta(\mathbf{x})$, we optimize a lower bound $\mathcal{L}$ on the marginal log-likelihood to approximate as close as we can $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \approx p(\mathbf{z}|\mathbf{x})$:

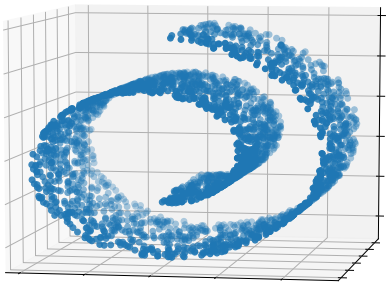
$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \right] \tag{6}$$

The first term encourages $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ to focus where the model puts high probability, $p(\mathbf{x}, \mathbf{z})$. While the KL term encourages $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ to avoid concentrating density.

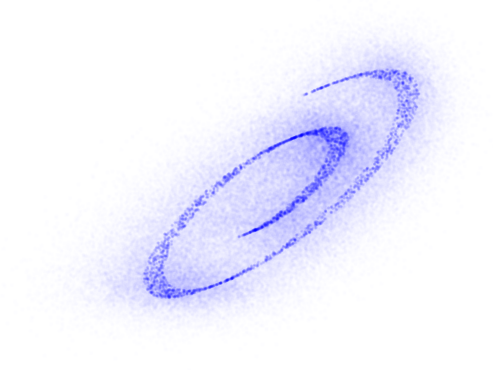
Typically, the encoder is a diagonal Gaussian whose parameters are a mapping from the data manifold to the latent space $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x}))$. For the observation model, for continuous data the decoder can be

$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z} \mid \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\sigma}^2_{\boldsymbol{\theta}}(\mathbf{z})) \tag{7}$$

and for binary data $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \text{Bernoulli}(f_{\boldsymbol{\theta}}(\mathbf{z}))$



a) Swiss Roll dataset in 3D



b) $q(\boldsymbol{\mu}_{\phi})$

Figure 3: toy example on the swiss roll dataset b) The learned lower dimensional representation of the Swiss roll in 2D with uncertainty, found by repeatedly sampling the approximate posterior on the decoder's parameters and computing the mean of $q_{\phi}(\mathbf{z}|\mathbf{x})$ with the samples. The samples are plotting in light blue and the mean (sample) is plotted in dark blue.

3 The BVAE : Uncertainty in uncertainty

The BVAE is an extended version of the standard VAE, which incorporates randomness in the encoder and decoder neural networks. For a full variational bayes treatment, we place priors on the $p(\boldsymbol{\phi})$ encoder and decoder $p(\boldsymbol{\theta})$. Assuming the following joint distribution

$$p(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\theta}) = p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z})p(\boldsymbol{\phi})p(\boldsymbol{\theta})$$

and corresponding approximate posterior

$$q_{\boldsymbol{\lambda}}(\mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\theta}|\mathbf{x}) = q_{\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})q_{\boldsymbol{\lambda}}(\boldsymbol{\phi})q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}),$$

since once again we have an intractable marginal likli-hood

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\theta}) d\mathbf{z} d\boldsymbol{\phi} d\boldsymbol{\theta}$$

we can obtain the following lower bound:

$$\begin{aligned} \log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \log \mathbb{E}_{q_{\boldsymbol{\lambda}}(\mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\theta}|\mathbf{x})} \left[\frac{p(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\theta})}{q_{\boldsymbol{\lambda}}(\mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\theta}|\mathbf{x})} \right] \\ &\geq \mathbb{E}_{q_{\boldsymbol{\lambda}}(\mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\theta}|\mathbf{x})} \left[\log \frac{p(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\theta})}{q_{\boldsymbol{\lambda}}(\mathbf{z}, \boldsymbol{\phi}, \boldsymbol{\theta}|\mathbf{x})} \right] \end{aligned}$$

By using a BNN in a VAE, we can sample parameters from its approximate posterior and approximate the predictive distribution to quantify uncertainty. We can do this in both the encoder and decoder which we describe in the next section.

4 The Inference Model

'Predictive' approximate posterior We can also treat the approximate posterior as a predictive distribution, where for some data point $\mathbf{x}'$ and corresponding $\mathbf{z}'$ we must approximate:

$$q(\mathbf{z}'|\mathbf{x}', \mathbf{X}) = \int q(\mathbf{z}'|\mathbf{x}', \boldsymbol{\phi}) q(\boldsymbol{\phi}|\mathbf{X}) d\boldsymbol{\phi} \tag{8}$$

$$= \mathbb{E}_{q(\boldsymbol{\phi}|\mathbf{X})}[q(\mathbf{z}'|\mathbf{x}', \boldsymbol{\phi})] \tag{9}$$

This is can be approximated using simple Monte Carlo sampling. We can also approximate the first moments of this distribution, which are also compound distributions

$$q(\boldsymbol{\mu}_{\phi}) = \int \boldsymbol{\mu}_{\phi} q(\boldsymbol{\phi}|\mathbf{X}) d\boldsymbol{\phi} \tag{10}$$

$$q(\boldsymbol{\sigma}_{\phi}) = \int \boldsymbol{\sigma}_{\phi} q(\boldsymbol{\phi}|\mathbf{X}) d\boldsymbol{\phi} \tag{11}$$

These distributions allows us to quantify uncertainty in our learned latent representation.

4.1 Swiss roll experiment

A VAE with a linear BNN encoder will perform dimensionality reduction, and will be a very similar model to Bayesian probabilistic PCA. We test such a model on the 3D Swiss roll dataset and visualize the distribution of the mean $q(\boldsymbol{\mu}_{\phi})$ using samples from the $q(\boldsymbol{\phi}|\mathbf{X})$. See Figure 3b. Notice the model learns to project the spiral into 2D space, shifting the orientation, with diffuse uncertainty from the center of its shifted axis.

5 The Generative Model

Posterior predictive *generative* distribution (PPGD) Our generative model also implicitly defines a posterior predictive distribution. Conditioning on training data $\mathbf{X}$ whenever we seek to sample new data or random samples from our model $\mathbf{x}'$ we are effectively sampling from this implicitly defined predictive distribution $p(\mathbf{x}'|\mathbf{X})$, by first sampling from some part of our latent space $\mathbf{z} \sim p(\mathbf{z})$. Then we must sample from the approximate posterior on the decoder's parameters conditioned on the training data $\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\mathbf{X})$:

$$p(\mathbf{x}'|\mathbf{X}) = \int p(\mathbf{x}'|\mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z}) q(\boldsymbol{\theta}|\mathbf{X}) d\mathbf{z} d\boldsymbol{\theta} \tag{12}$$

$$= \int p(\mathbf{x}'|\mathbf{z}', \boldsymbol{\theta}) q(\boldsymbol{\theta}|\mathbf{X}) d\boldsymbol{\theta}, \;\; \mathbf{z}' \sim p(\mathbf{z}) \tag{13}$$

$$= \frac{1}{S} \sum_{s=1}^{S} p(\mathbf{x}'|\boldsymbol{\theta}_s; \mathbf{z}) \tag{14}$$

where $\theta_s \sim q(\boldsymbol{\theta}|\mathbf{X}), \mathbf{z} \sim p(\mathbf{z})$ we can also perform this approximation using a traversal of the learned latent space and condition on a certain area $\mathbf{z}_\ell$ in the learned latent space $p(z) = \delta(\mathbf{z} - \mathbf{z}_\ell)$. Then we have :

$$p(\mathbf{x}'|\mathbf{X}) = \int p(\mathbf{x}'|\mathbf{z}, \boldsymbol{\theta}) p(\mathbf{z}) q(\boldsymbol{\theta}|\mathbf{X}) d\mathbf{z} d\boldsymbol{\theta} \tag{15}$$

$$= \int p(\mathbf{x}'|\mathbf{z}', \boldsymbol{\theta}) \delta(\mathbf{z} - \mathbf{z}_\ell) q(\boldsymbol{\theta}|\mathbf{X}) d\boldsymbol{\theta} \tag{16}$$

$$= \frac{1}{S} \sum_{s=1}^{S} p(\mathbf{x}'|\boldsymbol{\theta}_s; \mathbf{z}), \quad \theta_s \sim q(\boldsymbol{\theta}|\mathbf{X}) \tag{17}$$

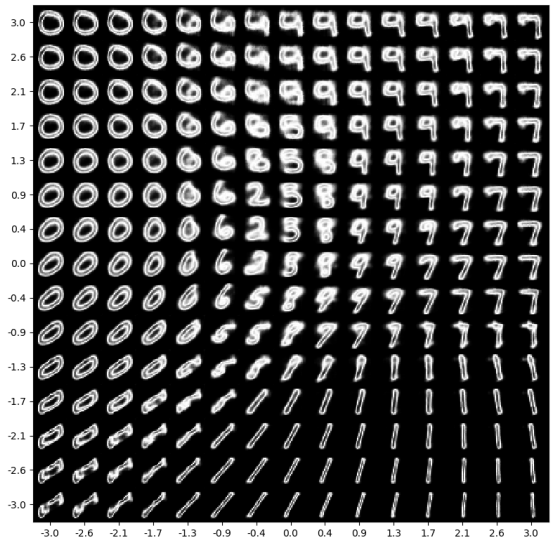
Estimate of PPGD's moments We can approximate the uncertainty in the generative model by sampling s parameter sets from $q(\boldsymbol{\theta}|\mathbf{X})$ to approximate the variance :

$$\text{Var}_{p(\mathbf{x}'|\mathbf{X})}[\mathbf{x}'] \approx \frac{1}{S} \sum_{s=1}^{S} \boldsymbol{m}(\boldsymbol{\theta}_s) \boldsymbol{m}(\boldsymbol{\theta}_s)^\top \tag{18}$$

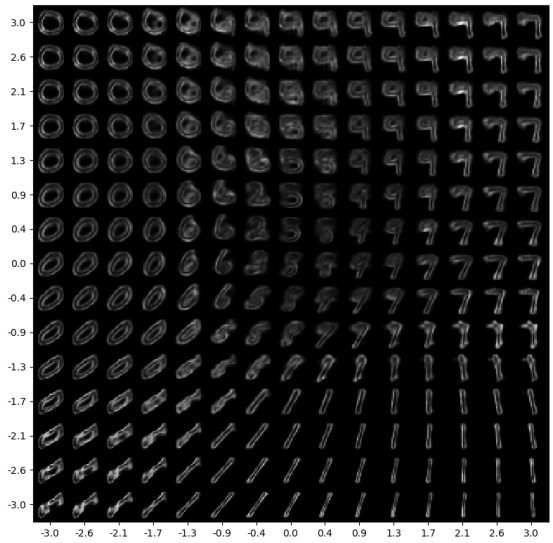
$$\mathbb{E}_{p(\mathbf{x}'|\mathbf{X})}[\mathbf{x}'] \equiv \boldsymbol{\mu}(\mathbf{z}_\ell) \approx \frac{1}{S} \sum_{s=1}^{S} f_{\boldsymbol{\theta}_s}(\mathbf{z}_\ell) \tag{19}$$

where $\boldsymbol{m}(\boldsymbol{\theta}_s) \equiv f_{\boldsymbol{\theta}_s}(\mathbf{z}_\ell) - \boldsymbol{\mu}(\mathbf{z}_\ell)$ and $f_{\boldsymbol{\theta}}(\mathbf{z}_\ell)$ is the decoder's generated sample x'

MNIST example We can visualize this uncertainty very easily on high dimensional image data. This visualization tells us what part of each digit the generative model is uncertain about creating. See Figure 4b for the uncertainty visualization of a BVAE trained on MNIST. Naturally, the generative model is more uncertain at the edges of each digit, as well as the region where the class changes.



(a) Entropy manifold



(b) Standard error manifold

Figure 4: The uncertainty visualization of the generative model of a VAE with 2D latent space sampled over a uniform grid.

6 Conclusion and Discussion

In conclusion, we have interpreted the BVAE's approximate posterior and probabilistic decoder as predictive distributions and used this to quantify uncertainty in the learned representation and the generative model. This shows BNNs can be useful in assessing learned representations in generative modeling.

References

[1] S. Marsland. *Machine Learning: An Algorithmic Perspective*. CRC Press, 2011.

[2] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The mnist database of handwritten digits, 1998. *URL http://yann. lecun. com/exdb/mnist*, 10:34, 1998.

[3] D. P Kingma and M. Welling. Auto-Encoding Variational Bayes. *International Conference on Learning Representations*, 2014.

[4] Christopher M Bishop. Mixture density networks. 1994.

[5] Zhenwen Dai, Andreas Damianou, Javier González, and Neil Lawrence. Variational auto-encoded deep gaussian processes. *arXiv preprint arXiv:1511.06455*, 2015.

[6] Geoffrey Hinton and Drew Van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *in Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*. Citeseer, 1993.

[7] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[8] Daniel Hernández-Lobato, Thang D Bui, Yinzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. Importance weighted autoencoders with random neural network parameters.

[9] Erik Daxberger and José Miguel Hernández-Lobato. Bayesian variational autoencoders for unsupervised out-of-distribution detection. *arXiv preprint arXiv:1912.05651*, 2019.

[10] Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011.

[11] Radford Neal. Priors for infinite networks. *In Bayesian Learning for Neural Networks*, 1996.

[12] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[14] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[15] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.