Importance Weighted Autoencoders with Random Neural Network Parameters

Daniel Hernández-Lobato
Universidad Autónoma de Madrid
daniel.hernandez@uam.es

Thang D. Bui
University of Cambridge
tdb40@cam.ac.uk

Yinzhen Li
University of Cambridge
yl494@cam.ac.uk

José Miguel Hernández-Lobato
University of Cambridge
jmh233@cam.ac.uk

Richard E. Turner
University of Cambridge
ret26@cam.ac.uk

1 Introduction to Variational and Importance Weighted Autoencoders

Deep generative models for unsupervised learning have recently received considerable attention [1, 2, 3, 4, 5, 6]. These models can find a set of low-dimensional representative latent features that can accurately describe observed data. Furthermore, they can also infer the underlying mechanism that generates, from these features, new data instances similar to the observed ones. In general, these models need to perform posterior inference during learning, a task that is carried out by training, in addition to the top-down generative network, a bottom-up recognition network. This recognition network is used to predict the posterior distribution of the latent variables given the observed ones.

Variational autoencoders (VAEs) are a family of generative models in which the parameters of the generative network and the recognition network are optimized during training to maximize a lower bound on the log-likelihood [3, 5, 6]. The VAE in [3] defines a generative process $p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta})$ for the observed variables of the i-th data instance, $\mathbf{x}_i$, given the corresponding latent variables $\mathbf{z}_i$. $p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta})$ is set to be a factorized Gaussian distribution (or a product of Bernoulli distributions in the case of binary data) whose mean and variance is computed by a deterministic feed-forward neural network with parameters $\boldsymbol{\theta}$. The recognition network computes $q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi})$, an approximation to $p(\mathbf{z}_i|\mathbf{x}_i)$. $q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi})$ is also a factorized Gaussian whose mean and variance is also computed by a feed-forward network with parameters $\boldsymbol{\phi}$. The prior for each $\mathbf{z}_i$, $p(\mathbf{z}_i)$, is set to be a product of standard Gaussians. During training, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are found by maximizing the lower bound

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_{i=1}^{N} \mathbb{E}_{\mathbf{z}_i} \left[\log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta})\right] - \sum_{i=1}^{N} \mathrm{KL}(q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi})||p(\mathbf{z}_i))\,, \tag{1}$$

where $\mathbf{z}_i$ is sampled from $q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi})$ and $\mathrm{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence. The expectations can be approximated via Monte Carlo, and the required gradients can be obtained using automatic tools. The objective in (1) is optimized using stochastic optimization tools combined with the reparametrization trick [3, 7].

The previous VAE is improved by the importance weighted autoencoder (IWAE) [6]. The IWAE considers a tighter lower bound of the log-likelihood of the data obtained by importance sampling:

$$\mathcal{L}_k(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_{i=1}^{N} \mathbb{E}_{\mathbf{z}_i^{(1)}, \ldots, \mathbf{z}_i^{(k)}} \left[\frac{1}{k} \sum_{j=1}^{k} \frac{p(\mathbf{x}_i|\mathbf{z}_i^{(j)}, \boldsymbol{\theta}) p(\mathbf{z}_i^{(j)})}{q(\mathbf{z}_i^{(j)}|\mathbf{x}_i, \boldsymbol{\phi})}\right] \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})\,, \tag{2}$$

where each $\mathbf{z}_i^{(1)}, \ldots, \mathbf{z}_i^{(k)}$ is sampled from the corresponding posterior approximation. When $k = 1$, $\mathcal{L}_k(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$. When $k > 1$, a tighter bound is obtained. The objective in (2) can be approximated and optimized as the objective of the previous VAE. Several experiments show that when $k > 1$ the IWAE outperforms the VAE in terms of the test log-likelihood. Furthermore, in [6] it is considered multiple stochastic hidden layers in the recognition model and the generative network.

Workshop on Bayesian Deep Learning, NIPS 2016, Barcelona, Spain.

2 Randomness in the Neural Network Parameters

We consider the possibility of enhancing the VAEs described in the previous section by introducing random network parameters. Instead of considering just a single point-estimate for the parameters of the generative model $\boldsymbol{\theta}$ and the recognition network $\boldsymbol{\phi}$, we introduce probability distributions over them. We expect that this will lead to more flexible models with an improved generalization performance, as in the networks considered in [8]. We set the generative model to be $p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\theta})q(\boldsymbol{\theta})$ and the recognition network to be $q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi})q(\boldsymbol{\phi})$, where $q(\boldsymbol{\theta})$ and $q(\boldsymbol{\phi})$ are variational distributions which have the form of a factorizing Gaussian. The means and variances of these distributions $\Omega = \{\boldsymbol{\mu}_\theta, \boldsymbol{\mu}_\phi, \boldsymbol{\sigma}^2_\theta, \boldsymbol{\sigma}^2_\phi\}$, are shared across data instances and they are found by maximizing the corresponding IWAE objective:

$$\mathcal{L}_k(\Omega) = \sum_{i=1}^{N} \mathbb{E}_{\substack{\mathbf{z}_i^{(1)},\ldots,\mathbf{z}_i^{(k)} \\ \boldsymbol{\theta}_i^{(1)},\ldots,\boldsymbol{\theta}_i^{(k)} \\ \boldsymbol{\phi}_i^{(1)},\ldots,\boldsymbol{\phi}_i^{(k)}}} \left[\frac{1}{k} \sum_{j=1}^{k} \frac{p(\mathbf{x}_i|\mathbf{z}_i^{(j)}, \boldsymbol{\theta}_i^{(j)}) p(\mathbf{z}_i^{(j)})}{q(\mathbf{z}_i^{(j)}|\mathbf{x}_i, \boldsymbol{\phi}_i^{(j)})} \right], \tag{3}$$

where each $\boldsymbol{\theta}_i^{(j)}$, $\boldsymbol{\phi}_i^{(j)}$ and $\mathbf{z}_i^{(j)}$, for $j = 1, \ldots, k$, is sampled from $q(\boldsymbol{\theta})$, $q(\boldsymbol{\phi})$ and $q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi}_i^{(j)})$, respectively. The objective in (3) can also be approximated by Monte Carlo and optimized using stochastic optimization tools, as in the previous section. Note the lack of a prior distribution for the parameters of the generative model, $\boldsymbol{\theta}$, in our formulation. This is because $q(\boldsymbol{\theta})$ is simply a variational distribution. The lack of $p(\boldsymbol{\theta})$ can also be motivated by the lack of any regularization for the weights of the generative network in the original VAE [3]. Furthermore, according to our experiments (not shown), introducing a prior distribution $p(\boldsymbol{\theta})$ for the generative model deteriorates the results reported in the next section.

3 Experimental Evaluation

The proposed IWAE method with random network weights (IWAER) is evaluated on two datasets: MNIST [9] and Omniglot [10]. The generative and recognition models are neural networks with one deterministic hidden layer of 400 units. We consider 20 latent variables. We train each method during a total of 500 epochs with ADAM and its default parameters [11]. A minibatch size of 100 is used and k, the number of samples, is set to 25. The quality of the inference and generative networks learnt is compared using the log-likelihood of test images, which is computed using importance sampling with 2000 samples drawn from the recognition model and the variational distributions $q(\boldsymbol{\theta})$ and $q(\boldsymbol{\phi})$ [6]. We compare with the results obtained by the IWAE and by a version of IWAER that only considers randomness in the recognition weights, $\text{IWAER}_{\text{rec}}$. The results averaged over 5 trials are displayed in Table 1. These results show a significant gain obtained by considering random network weights, demonstrated by a better log-likelihood on test data of IWAER.

Table 1: Average test log-likelihood for each method.

Dataset	**IWAE**	**IWAER**	**$\text{IWAER}_{\text{rec}}$**
MNIST	-95.182±0.022	**-94.346±0.025**	-94.709±0.025
Omniglot	-118.771±0.035	**-118.540±0.049**	-118.647±0.031

4 Conclusions and Future Work

We have addressed the task of introducing randomness in the parameters of the neural networks employed in the IWAE for generative and recognition tasks. For this, we have introduced additional variational distributions in the model, *i.e.*, $q(\boldsymbol{\theta})$ and $q(\boldsymbol{\phi})$, whose parameters are found by maximizing a lower bound on the log-likelihood of the training data. The resulting model, IWAER, has been evaluated on two datasets: MNIST and Omniglot. The results obtained show a significant improvement in terms of the log-likelihood of test images. These results confirm the benefits of a more flexible recognition and generative model obtained by considering random network weights. Future work includes addressing more complicated networks with several stochastic hidden layers, similar to the ones described in [6]. In that case we expect to obtain results that are close and even better than the state-of-the-art. Additional future work includes considering other approaches for training the model, such as black-box-alpha [12] and considering other models such as the ladder-VAE [13].

Acknowledgments

TDB thanks Google for funding his European Doctoral Fellowship. JMHL acknowledges support from the Rafael del Pino Foundation. DHL and JMHL acknowledge support from Plan Nacional I+D+i, Grants TIN2013-42351-P and TIN2015-70308-REDT, and from Comunidad de Madrid, Grant S2013/ICE-2845 CASI-CAM-CM. YL thanks the Schlumberger Foundation for her Faculty for the Future PhD fellowship. RET thanks EPSRC grants EP/G050821/1 and EP/L000776/1.

References

[1] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

[2] R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, 2009.

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

[4] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and Wierstra. DRAW: A recurrent neural network for image generation. In *International Conference on Machine Learning*, 2015.

[5] D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2015.

[6] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

[7] D. P. Kingma, T. Salimans, and M. Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*. 2015.

[8] S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft. Learning and policy search in stochastic dynamical systems with Bayesian neural networks. 2016. arXiv:1605.07127.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998.

[10] B. M. Lake, R. R. Salakhutdinov, and J. Tenenbaum. One-shot learning by inverting a compositional causal process. In *Advances in Neural Information Processing Systems*. 2013.

[11] D. Kingma and J. L Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[12] J. M. Hernández-Lobato, Y. Li, M. Rowland, T. D. Bui, D. Hernández-Lobato, and R. E. Turner. Black-box alpha divergence minimization. In *International Conference on Machine Learning*, 2016.

[13] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. 2016. arXiv:1602.02282.