Gaussian dropout as an information bottleneck layer

Mélanie Rey
DeepMind
melanierey@deepmind.com

Andriy Mnih
DeepMind
andriy@deepmind.com

1 Introduction

As models become more powerful, they can acquire the ability to fit the data well in multiple qualitatively different ways. At the same time, we might have requirements other than high predictive performance that we would like the model to satisfy. For example, in causality, which variables are used and how they are combined to make predictions is paramount, and in generative modelling, we might want to specify which aspects of the data should be modelled by which model component. One way to express such preferences is by controlling the information flow in the model with carefully placed information bottleneck layers, which limit the amount of information that passes through them by applying noise to their inputs. The most notable example of such a layer is the stochastic representation layer of the Deep Variational Information Bottleneck [Alemi et al., 2016, DVIB] which requires including a variational upper bound on the mutual information [Poole et al., 2019] between its inputs and outputs as a penalty in the loss function. We show that using Gaussian dropout [Wang and Manning, 2013, Kingma et al., 2015], which involves multiplicative Gaussian noise, achieves the same goal in a simpler way, as it induces an upper bound on the mutual information between its inputs and outputs without requiring any additional terms in the objective.

We evaluate the two approaches in the generative modelling setting, by applying them to the long-standing problem of increasing the use of latent variables in a VAE with an autoregressive decoder for modelling images. By placing an information bottleneck layer in the decoder we encourage the latent variables to encode the more global and high-level properties of the image while letting the autoregressive component capture the fine detail. We observe that the Gaussian dropout layer yields a better trade-off between the use of latent variables and the fit to the data than the widely used variational information bottleneck layer.

2 Two types of information bottlenecks layers

Perhaps the most common strategy for limiting the information passing through a particular pathway is inserting a Gaussian layer (with a diagonal covariance) with its mean and variance being functions of the preceding layer. Such a layer destroys some of the information contained in the means by adding zero-mean Gaussian noise to them. To prevent learning from counteracting this effect by driving the variance of the noise to zero and thus making information capacity infinite, a variational upper bound on the mutual information between the inputs and outputs is added to the overall loss, after being scaled by a strength hyperparameter. The encoder in the DVIB is the most prominent application of this approach, as is described in Appendix A. Despite the success of DVIB, altering the loss in this way potentially makes the optimization problem harder and requires estimating the mutual information, even when we are not interested in measuring it, in order to penalize it.

We will show that Gaussian dropout, which has been introduced as an alternative to the original (binary) dropout [Srivastava et al., 2014, Wang and Manning, 2013], provides a simple and highly competitive alternative information bottleneck layer. Unlike DVIB, which is based on additive Gaussian noise, Gaussian dropout uses multiplicative noise: given an input X, it outputs $Y = X\varepsilon$ using $\varepsilon \sim \mathcal{N}(1, \sigma^2)$. Here σ is a hyperparameter that controls the strength of the bottleneck, so unlike for the variational bottleneck layer, this noise distribution has no learnable parameters and does not

Bayesian Deep Learning workshop, NeurIPS 2021.

require estimating mutual information or including any additional terms in the loss. While Gaussian dropout is typically used for regularizing models, we provide a new perspective on it by showing that it induces a bound on the mutual information between its input and output. To do this, we analyze Gaussian dropout in information-theoretic terms in the following theorem, following the strategy used for the additive white Gaussian noise channel [Ch. 10 of Cover and Thomas, 2006].

Theorem 1. *The mutual information between the input and output of a Gaussian dropout layer has an upper bound which is invariant to input rescaling.*

Proof. Let X be the input and Y output of a Gaussian dropout layer with noise variance σ^2. Then $Y|X \sim \mathcal{N}(X, X^2\sigma^2)$, with the conditional (differential) entropy $h(Y|X) = \mathbb{E}_X\left[\frac{1}{2}\log(2\pi e X^2\sigma^2)\right]$. We can compute the marginal variance of Y using the law of total variance: $V_y = \mathbb{E}_X[\mathrm{Var}(Y|X)] + \mathrm{Var}(\mathbb{E}[Y|X]) = \left(V_X + \bar{X}^2\right)\sigma^2 + V_X$, where $\bar{X}$ and V_X are the expectation and variance of X, respectively. Then

$$
\begin{aligned}
I(X,Y) &= h(Y) - h(Y|X) && (1)\\
&\leq \tfrac{1}{2}\log\left(2\pi e V_y\right) - \mathbb{E}_X\left[\tfrac{1}{2}\log\left(2\pi e X^2\sigma^2\right)\right] && (2)\\
&= \tfrac{1}{2}\mathbb{E}_X\left[\log\tfrac{V_y}{X^2\sigma^2}\right] && (3)\\
&= \tfrac{1}{2}\mathbb{E}_X\left[\log\tfrac{\left(V_X+\bar{X}^2\right)\sigma^2+V_X}{X^2\sigma^2}\right] && (4)\\
&= \tfrac{1}{2}\mathbb{E}_X\left[\log\tfrac{V_X+\bar{X}^2+V_X/\sigma^2}{X^2}\right], && (5)
\end{aligned}
$$

where (2) holds since the Gaussian distribution is the maximum entropy distribution with the given variance. Since for $a \in \mathbb{R}$, $\mathrm{Var}(aX) = a^2V_X$ and $\mathbb{E}_X[aX] = a\bar{X}$, the bound in (5) is invariant to rescaling of the input X. □

This invariance of the bound under input rescaling is interesting because it does not hold for the Gaussian channel with additive noise, since its capacity can be made arbitrarily high by increasing the magnitude its inputs.

3 Application: Hybrid VAE-PixelCNN model

We apply our approach of controlling information flows to the problem of effectively combining two generative models with complementary strengths for modelling images: a VAE and a PixelCNN. Latent variable models such as VAEs tend to capture the global structure and dependencies between different parts of an image well but have difficulty modelling details such as edges. Autoregressive models such as PixelCNN can model complex local structures and produce rich, detailed images, but have more trouble with longer-range dependencies. The existing approaches for combining VAEs with autoregressive decoders [Gulrajani et al., 2017, Chen et al., 2017] work by restricting the expressive power of the autoregressive decoders in various ways in order to ensure that the latent variables are used. Gulrajani et al. [2017], for example, use a reduced number of PixelCNN layers. Instead of imposing architectural constraints, we propose to limit the decoder by inserting an information bottleneck layer.

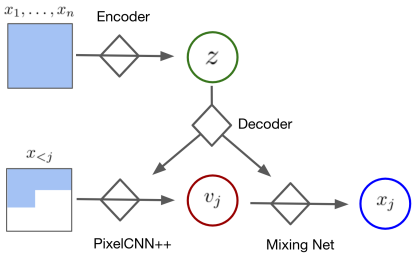

Figure 1: Hybrid VAE-PixelCNN++ model

Let $x = (x_1, \ldots, x_n)$ denote the input image and z the VAE latents. We use a hybrid decoder that combines the outputs of a convolutional net decoding the VAE latents with those of a PixelCNN++ on the per-pixel basis using a simple MLP mixing network. More precisely, the VAE decoder outputs a tensor $u = (u_1, \ldots, u_n)$ with the same spatial dimensions as the input image; then the PixelCNN++ predicts a low-dimensional output vector v_j for each pixel j using the preceding pixels in the input image $x_{<j}$ and the decoded VAE latents u. Finally, a one-hidden-layer MLP combines v_j and u_j

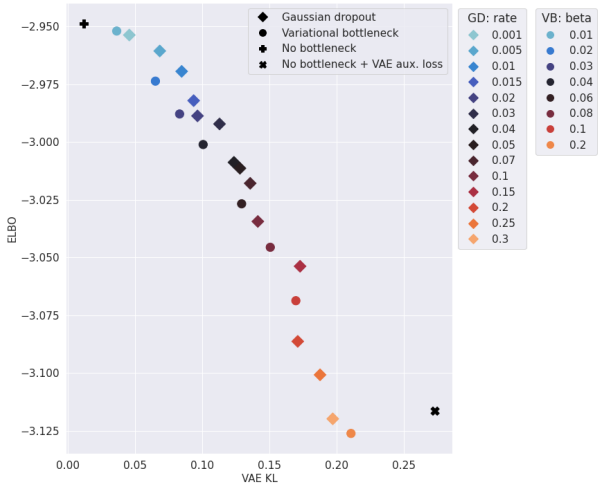

Figure 2: Test set ELBO vs. VAE KL for models trained on CIFAR10 using either a Gaussian dropout layer (GD) or a variational bottleneck layer (VB). Models with GD and VB are trained for a number of dropout rates and values of β respectively. For reference, we also include results for a model with no bottleneck layer, trained with and without an auxiliary decoder.

to obtain the output distribution at each pixel location. This clean separation of the two modelling contributions allows us to limit the information flow through the PixelCNN++ component by passing the low-dimensional output vector v_j it produces for each pixel j, through an information bottleneck layer before feeding the result into the mixing network.

We train models that do not use the variational information bottleneck by maximizing the ELBO:

$$\mathcal{L}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \mathrm{KL}(q_\phi(z|x) \parallel p(z)). \tag{6}$$

When using the the variational information bottleneck layer, which induces a distribution $p(v_j|x_{<j}, z)$ over the noisy PixelCNN++ output for each pixel j, we add the DVIB-like penalty term to the ELBO resulting in the following objective:

$$\mathcal{L}_{VBL}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \mathrm{KL}(q_\phi(z|x) \parallel p(z)) - \beta \sum_j \mathrm{KL}(p(v_j|x_{<j}, z) \parallel \mathcal{N}(0, I)).$$

No matter which of the bottleneck layers we use, the information passed through autoregressive pathways will be bottlenecked, whereas the global information captured by the VAE latents will be available without any restrictions for predicting each pixel.

4 Experimental results

We trained two versions of the model on CIFAR10: one using the Gaussian dropout layer and one using the variational bottleneck layer. In each case, we performed a sweep over the hyperparameter controlling the strength of the bottleneck, which for the Gaussian dropout layer is the dropout rate p which maps to the noise standard deviation σ as $\sigma(p) = \sqrt{\frac{p}{1-p}}$. We evaluate the resulting models on the set test in terms of their ELBO, which quantifies the fit to the data, and the VAE KL, which measured the use of the latent variables. From Figure 2 we can see that there is a negative dependence between these two quantities for the models, which means that we have to trade one off against the other. However, we can see that the models trained using the Gaussian dropout layer tend to achieve a better trade-off compared to the models trained using the variational bottleneck layer. For reference, the figure also includes results for a model trained without the information bottleneck layer, as well as a model trained jointly with an auxiliary non-autoregressive decoder to encourage the use of latent variables [Lucas and Verbeek, 2018]. See Appendix B for the details of our experimental setup.

Figure 3: Samples from models trained with different Gaussian dropout rates (0.015 first panel and 0.15 second panel) compared to samples obtained using different KL penalty weights (0.02 third and 0.08 last panel)

References

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2016.

Alexander A. Alemi, Ben Poole, Ian Fischer, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. An information-theoretic analysis of deep latent-variable models. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Xi Chen, Diederik P. Kingma, Tim Salimans, Duan Yan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. In *International Conference on Learning Representations*, 2017.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville. PixelVAE: A latent variable model for natural images. In *International Conference on Learning Representations*, 2017.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2015.

Thomas Lucas and Jakob Verbeek. Auxiliary guided autoregressive variational autoencoders. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, 2019.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations*, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37th Annual Allerton Conference on Communication, Control and Computing*, 1999.

Sida Wang and Christopher Manning. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

A Deep Variational Information Bottleneck

The information bottleneck framework (IB) [Tishby et al., 1999] provides a way of learning a representation Z of an input X, so that it contains little information from X other than that useful for predicting the target Y. Representations are learned by maximizing the objective $I(Z,Y) - \beta I(X,Z)$, where the β hyperparameter specifies the desired trade-off between informativeness and compression. The resulting optimization problem is intractable in general, and Deep Variational Information Bottleneck provides a practical alternative based on variational bounds on mutual information. DVIB models Z as a Gaussian vector with the mean and variance parameterized as functions of the the input X (typically neural networks): $p(z|x) = \mathcal{N}(\mu(x), \mathrm{diag}(\sigma(x)^2))$. To deal with the difficulty of estimating mutual information, DBIV replaces each of the two MI terms with a variational bound [Alemi et al., 2018] on it. Here we will consider only the bound for $I(X,Z)$, which implements the kind of bottleneck layer we are interested in in this paper. The bound has the form

$$I(X,Z) \leq \mathbb{E}_X[\mathrm{KL}(p(z|x) \parallel q(z))], \tag{7}$$

and is obtained by replacing the intractable marginal $p(z)$ in $I(X,Z) = \mathbb{E}_X[\mathrm{KL}(p(z|x) \parallel p(z))]$ with any distribution $q(z)$, which in practice is usually $\mathcal{N}(0, I)$.

In computational terms we can interpret DVIB as limiting the information content of a representation Z by adding zero-mean Gaussian noise $\mathcal{N}(0, \mathrm{diag}(\sigma(x)^2))$ to a deterministic representation $\mu(x)$. The inclusion of the $\beta\mathrm{KL}(p(z|x) \parallel p(z))$ in the DVIB objective ensures that the amount of information passing through Z does not become too large, by adjusting the noise variances accordingly.

B Experimental setup

All models use the same architecture, with the VAE encoder and decoder being ResNets with 7 blocks, each with 512 channels, 128 bottleneck channels, and ELU activations; the decoder has 256 output channels. The VAE latents are arranged spatially as 4×4 with 16 latents at each location; we use the standard normal prior. The PixelCNN++ network is implemented following Salimans et al. [2017], including the use of binary dropout with the rate of 0.5. The pixel output distribution is a discretized mixture of logistics with 10 mixture components and conditional color channels. The PixelCNN++ output for each pixel v_j is a 10-dimensional vector, which we feed to the information bottleneck layer when we use one. The mixing network is a one-hidden-layer MLP with 266 ELU units. Training is done with the Adam optimizer (β_1 =0.9, β_2 =0.999, $\epsilon = 10^{-8}$) for 350 epochs, annealing the learning rate from 10^{-4} to 4×10^{-6}. The numbers in Figure 2 were computed on the CIFAR10 test set, with the ELBO and VAE KL reported in bits per dimension. At test time, we do not inject noise for either type of the information bottleneck layer, using their unchanged input as the output instead.