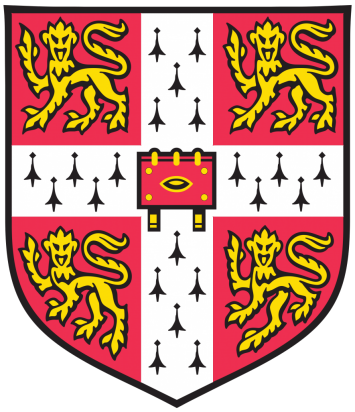
Importance Weighted Autoencoders

MLSALT 4

C. Tegho J.Rampersad S.Pascual Diaz

Machine Learning,
Speech and Language Technology
MPhil



CUED
Cambridge University
UK

Contents

1 Introduction **2**

2 Variational Autoencoders (VAEs) **2**
2.1 Architecture and Objective function 2
2.2 Re-parametrisation Trick 3

3 Importance Weighted Autoencoder (IWAE) **4**
3.1 Objective function 4
3.2 Training 5
3.3 Relation to α-divergence minimisation 6

4 Experiments **6**

5 Results and Discussion **7**
5.1 Generative log-likelihood performance 7
5.2 Increasing the number of stochastic layers 7
5.3 Latent Space Representation 8
5.4 Warm-Up 10
5.5 Effective Sample Size 11

6 Future Work **13**

7 Conclusion **13**

1 Introduction

Variational based generative approaches have been shown to efficiently perform both inference and learning in deep directed probabilistic models even in the presence of continuous latent variables with intractable posterior distributions [1]. This report reviews the theory of variational methods, beginning with an introduction to the VAE by Kingma and Welling. We then discuss modifications made by Burda et al in their recently published work on Importance Weighted Autoencoders (IWAEs) before recreating their experiments. With similar architectures and training times, IWAEs have been shown to capture richer latent space representations than VAEs through optimisation of a tighter lower bound on the log-likelihood. Combined with a less constraining objective function has resulted in improvements on several density modelling benchmarks. [2]

In the extensions, we empirically measure the activity of latent units and compute a heuristic measure of the *effective sample size* for both VAEs and IWAEs. Our results show that both systems fail to exploit their full modelling capacity - an effect that is compounded with increasing numbers of stochastic layers. In these higher layers, a number of units 'switch off' with their firing activity falling close to zero. The consequence is that the learned latent space representations have an effective dimensionality much lower than the model's potential. We discuss possible methods to avoid inactivity within the layers including gradual inclusion of the KL-divergence term through use of the *Warm-up* technique in [14].

2 Variational Autoencoders (VAEs)

VAEs are a family of directed graphical-based, generative models that consists of two mayor parts: a bottom-up *recognition* network over a set of latent variables h, $p_\theta(h|x)$, for inference; and a top-down *generative* network over the observed data $p_\theta(x|h)$. Typically the true posterior distribution $p_\theta(h|x)$ is intractable, but we would like to learn and infer θ and h respectively so that we can perform data generation, or marginal likelihood estimation. VAEs introduce an approximate posterior $q_\phi(h|x)$ parametrised by parameters ϕ predictable by Deep Neural Networks. Both architectures (recognition and generative networks) are jointly trained to drive up a lower bound on the log-likelihood.

2.1 Architecture and Objective function

Neural networks are exploited in the parametrisation of both generative and recognition networks, given their ability to universally approximate arbitrary probability distributions [15]. In the case of the recognition model, hidden layers of the neural network can be factorised allowing back-propagated parameter updates in an analogous fashion to standard deterministic neural networks.

$$q_\phi(h|x) = q_\phi(\mathbf{h}^1|x)q_\phi(\mathbf{h}^1|\mathbf{h}^2)...q_\phi(\mathbf{h}^L|\mathbf{h}^{L-1}) \tag{1}$$

where $h = (\mathbf{h}^1, \ldots, \mathbf{h}^L)$ denotes the units in the stochastic hidden layers. The derivation of the variational lower-bound on the log-likelihood for the VAE relies upon an assumption that the posterior distribution is approximately factorial – leading to oversimplifications. As it will discussed in later sections, using multiple posterior samples can give the model extra flexibility to model latent distributions that do not fully match up with the constrained variational assumptions of VAEs.

Similarly, in the generative distribution:

$$p(x|\theta) = \int_{\mathbf{h}^1,\ldots,\mathbf{h}^L} p(\mathbf{h}^L|\theta)p(\mathbf{h}^{L-1}|\mathbf{h}^L,\theta)...p(x|\mathbf{h}^1,\theta)dh \tag{2}$$

VAEs are trained to maximize a variational lower bound on $log(p(x))$ derived from Jensen's inequality. As we will see, contrary to IWAEs, this objective can be written as the log-likelihood with a KL divergence term acting as a form of variational regularisation:

$$\mathcal{L}(x) = \mathbb{E}_{h\sim q_\phi(h|x)}[\log \frac{p_\theta(x,h)}{q_\phi(h|x)}] \tag{3}$$

$$= \log p(x) - \mathcal{D}_{KL}(q_\phi(h|x)||p_\theta(h|x)) \tag{4}$$

This objective function severely discourages samples from the posterior that do not explain the data (note argument of logarithm) even if the majority of the approximate posterior explains the data well, limiting the expressive power of the model.

2.2 Re-parametrisation Trick

One of the main contributions of the VAE article, which is also adopted for IWAEs, is the implementation of the *re-parametrisation trick* during training. Gradient-based stochastic optimisation techniques require an unbiased gradient estimator of the variational objective in order to perform approximate inference. In previous work [6], the gradient with respect to the recognition parameters ϕ, $\nabla_\phi \mathcal{L}(x)$, is estimated using $\frac{1}{K}\sum_{k=1}^{K}[\log \frac{p_\theta(x,h_k)}{q_\phi(h_k|x)}]\nabla_\phi q_\phi(h_k|x)$, with posterior samples $h_k \sim q_\phi(h|x)$. Although training is feasible using a REINFORCE update with reward signal $\frac{p_\theta(x,h)}{q_\phi(h|x)}$, it is impractically slow for inference as the scaling of the gradient term causes a high variance in the gradient estimator.

In order to obtain an estimator with a lower variance, the re-parametrisation trick computes recognition network samples as a *deterministic* function $h_k = h(x,\phi,\epsilon_k)$ of a set of independent *auxiliary random variables* $\epsilon = (\boldsymbol{\epsilon}^1, \boldsymbol{\epsilon}^2, \ldots, \boldsymbol{\epsilon}^L)$ drawn from fixed distributions. In particular, for the Gaussian factor distributions in the recognition network used in our implementation of VAEs and IWAEs, $q_\phi(\mathbf{h}^l|\mathbf{h}^{l-1}) \sim \mathcal{N}(\mathbf{h}^l|\boldsymbol{\mu}(\mathbf{h}^{l-1},\phi), \Sigma(\mathbf{h}^{l-1},\phi))$, the l^{th} layer units can be expressed as:

$$\mathbf{h}^l(\mathbf{h}^{l-1},\boldsymbol{\epsilon}^l,\phi) = \boldsymbol{\mu}(\mathbf{h}^{l-1},\phi) + \Sigma^{1/2}(\mathbf{h}^{l-1},\phi)\boldsymbol{\epsilon}^l \text{ for } \boldsymbol{\epsilon}^l \sim \mathcal{N}(0,\mathbf{I}) \tag{5}$$

Resulting in an unbiased differentiable estimator of $\mathcal{L}(x)$ suitable for optimisation through stochastic back-propagation:

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I})}[\log \frac{p_\theta(x,h(x,\phi,\epsilon))}{q_\phi(h(x,\phi,\epsilon)|x)}] \tag{6}$$

Removing the dependence on sampled parameters ϕ of posterior samples h_k, allows ∇_ϕ to pass though the expectation symbol and compute derivatives with respect to the latent variables explicitly. As a consequence, the resulting gradient estimator of the variational objective is unbiased and has a lower variance.

3 Importance Weighted Autoencoder (IWAE)

The VAE objective as described in (3) highly penalises those posterior samples $h_k \sim q_\phi(h|x)$ fail to explain observations, since q $p_\theta(x, h_k)$ gives a even smaller $\mathcal{L}(x)$. Hence in order to ensure $\mathcal{L}(x)$ sets a good lower bound to the log-likelihood when training, variational assumptions must be approximately satisfied. This forces the generative network posterior $p_\theta(x|h_k)$ to be approximately factorial:

$$p_\theta(x|h) = p_\theta(x|\mathbf{h}^1)p_\theta(\mathbf{h}^L)\prod_{l=1}^{L-1} p_\theta(\mathbf{h}^l|\mathbf{h}^{l+1})$$

Where the parameters θ are predictable through feed forward Neural Networks. To solve this issue, in Importance Weighted Autoencoders (IWAEs), a similar architecture to VAEs is trained to optimise a tighter lower bound to $\mathcal{L}(x)$, so that Recognition network samples spread out within the posterior can also have a contribution. Consequently, variational assumptions can be loosen up and the model gains flexibility in the generative process.

3.1 Objective function

The idea behind the derivation of the IWAE objective is to approximate the marginal log-likelihood $\log p(x_{1:N})$ with its Importance Sampling approximation using the recognition network posterior $q_\phi(h|x)$ as proposal:

$$\log p(x_{1:N}) = \sum_{n=1}^{N} \log \int_{h_n} p_\theta(x_n, h_n) dh_n \approx \sum_{n=1}^{N} \log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(x_n, h_{n,k})}{q_\phi(h_{n,k}|x_n)} \tag{7}$$

The IWAE lower bound for $\log p(x)$ for a data point x is then defined as:

$$\mathcal{L}_K(x) = \mathbb{E}_{h_{1:K}\sim q_\phi(h|x)}[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(x, h_k)}{q_\phi(h_k|x)}] \tag{8}$$

obtained at the expense of a K-fold increase of recognition posterior samples $h_{1:K} = (h_1, \ldots, h_K)$. For convenience, we often write importance weights $\frac{p_\theta(x,h_k)}{q_\phi(h_k|x)} = w(x, h_k, (\theta, \phi)) = w_k$. In our experiments, these will be used as an heuristic metric of the number of effective samples yielded by the recognition network. There are a few properties worth noticing about the IWAE objective:

- $\log p(x) \geq \mathcal{L}_K(x)$ for all $K \in \mathbb{N}$ – IWAE objective is lower bound of the log-likelihood.
- $\mathcal{L}_m(x) \geq \mathcal{L}_k(x)$ for all $m \geq k$ – The lower bound gets tighter as the number of posterior samples increases, although this may have computational drawbacks.
- $\lim_{k\to\infty} \mathcal{L}_k(x) = \log p(x)$ as long as importance $w_{1:k}$ are all bounded.
- $\mathcal{L}_1(x) = \mathcal{L}(x)$ – IWAE objective with one sample is equivalent to the VAE objective.

It is a well-known fact that Importance Sampling estimates suffer from high-variance as the number of dimensions increases, in cases where the proposal does not match the target distribution. To show this is not the case for the Monte Carlo estimator of $\mathcal{L}_K(x)$, first we show the probability that

the estimator $\hat{\mathcal{L}}_K(x)$ overestimates $\log p(x)$ is relatively small since $Pr(\hat{\mathcal{L}}_K(x) > \log p(x)+y) \leq e^{-y}$. Then, we find an upper bound to the mean absolute deviation (MAD) of the estimator in terms of the gap between $\mathcal{L}_K(x) = \mathbb{E}[\hat{\mathcal{L}}_K(x)]$ and $\log p(x)$:

$$\mathbb{E}[|\hat{\mathcal{L}}_K(x) - \mathcal{L}_K(x)|] \leq 2 + 2(\log p(x) - \mathcal{L}_K(x)) \tag{9}$$

Using MAD as deviation metric does not set a direct bound on the variance, but it shows how unlikely it is for the Monte Carlo estimator to have a high variance.

3.2 Training

Similarly to VAEs, the *re-parametrisation trick* can also be adopted for IWAEs. In this case each of the K-recognition posterior samples h_k can be computed as a deterministic function $h(x, \epsilon_k, \phi)$ of a set of auxiliary random variables $\epsilon_k = (\boldsymbol{\epsilon}_k^1, , \ldots, \boldsymbol{\epsilon}_k^L)$ from a fixed distribution:

$$\mathcal{L}_K(x) = \mathbb{E}_{\epsilon_{1:K} \sim \mathcal{N}(0,\mathbf{I})}[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\theta(x, h(x, \epsilon_k, \phi))}{q_\phi(h(x, \epsilon_k, \phi)|x)}] \tag{10}$$

Importance weights are now a function of ϵ_k, $w_k = w(x, h(x, \epsilon_k, \phi), (\theta, \phi))$. Then, the gradient is simply:

$$\nabla_\phi \mathcal{L}_K(x) = \mathbb{E}_{\epsilon_{1:K}}[\nabla_\phi \log \frac{1}{K} \sum_{k=1}^{K} w_k] = \mathbb{E}_{\epsilon_{1:K}}[\frac{1}{\sum_{j=1}^{K} w_j} \sum_{k=1}^{K} \nabla_\phi w_k] \tag{11}$$

Using $\nabla_\phi w_k = w_k \nabla_\phi \log w_k$,

$$= \mathbb{E}_{\epsilon_{1:K}}[\sum_{k=1}^{K} (\frac{w_k}{\sum_{j=1}^{K} w_j}) \nabla_\phi \log w_k] = \mathbb{E}_{\epsilon_{1:K}}[\sum_{k=1}^{K} \tilde{w}_k \nabla_\phi \log w_k] \tag{12}$$

where we set $\tilde{w}_k = \frac{w_k}{\sum_{j=1}^{K} w_j}$ as the normalised importance weights. The highest computational cost of this training procedure comes from computing gradients $\nabla_\phi \log w(x, h(x, \epsilon_k, \phi), (\theta, \phi))$ which requires separate forward and backwards passes in back-propagation for each sample k. These can be reduce by making making the smart choice $\epsilon_k \propto \tilde{w}_k$, but it comes to an expense of increasing the variance of the Monte Carlo estimator of (12).

Expanding the gradients $\nabla_\phi \log w_k$ in terms of $\nabla_\phi \log p_\theta(x, h(x, \epsilon_k, \phi)) - \nabla_\phi \log q_\phi(h(x, \epsilon_k, \phi)|x)$, we see how the first term adjusts the recognition network to produce posterior samples that make good predictions, whilst the second term encourages the recognition network to have a spread-out distribution. This split is exploited by other training procedures such as the *wake-sleep algorithm* [7], which does not make use of the re-parametrisation trick. Instead, p_θ and q_ϕ are updated in turns using (12) as objective as biased estimate of the log-likelihood gradient.

Contrary to VAEs, in IWAEs theses gradients are scaled by the normalised importance weights $\tilde{w}_k$, which penalises less those posterior samples that do not explain the data well, given the model more flexibility.

3.3 Relation to α-divergence minimisation

The maximisation of the IWAE objective is closely related to the local minimization of α-divergence in the *Power Expectation–Propagation* (PEP) setting [8]. For generative models, an intractable posterior $p_\theta(h|x) \propto p(h)p_\theta(x|h)$ is approximated with a proposal distribution $q_\phi(h|x) \propto p(h)\tilde{q}_\phi(h|x)$. The factor $\tilde{q}_\phi(h|x)$ is easy to sample from (typically Gaussian) and tuned so that for each data sample n, the local α-divergence[1] with respect to latent variables h:

$$D_\alpha(p_\theta(x_n|h)\prod_{j\neq n}\tilde{q}_\phi(h|x_j)||\tilde{q}_\phi(h|x_n)\prod_{j\neq n}\tilde{q}_\phi(h|x_j)) \tag{13}$$

is minimised. In practice, this is done by maximising the PEP evidence:

$$\log Z_{PEP} = \log Z_q + \frac{1}{\alpha}\sum_{n=1}^{N}\log \mathbb{E}_{h\sim q}[\frac{p_\theta(x_n|h)}{\tilde{q}_\phi(h|x_n)}]^\alpha \tag{14}$$

where the expectation can estimated using Monte Carlo with K-samples $h_{k,n} \sim q_\phi(h|x_n)$. If we now set $\alpha = 1$ and replace $\frac{p_\theta(x_n|h_{k,n})p(h_{k,n})}{\tilde{q}_\phi(h_{k,n}|x_n)p(h_{k,n})} = \frac{p_\theta(x_n,h_{k,n})}{q_\phi(h_{k,n}|x_n)}Z_q^{-1/N}$, we obtain an estimate of the evidence $\log Z_{PEP}$:

$$\sum_{n=1}^{N}\log\frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(x_n,h_{k,n})}{q_\phi(h_{k,n}|x_n)} = \sum_{n=1}^{N}\mathcal{L}(x_n) \tag{15}$$

Hence jointly training the recognition network and the generative model to optimise the IWAE objective is equivalent to minimizing local α-divergences with $\alpha = 1$. Anecdotally, for any two distributions $D_1(p||q) = \mathcal{D}_{KL}(p||q)$ which is the reverse KL-divergence.

4 Experiments

To test the models, we used MNIST, a dataset of handwritten, binarized 28x28 images [4]. The dataset was split into 60k training and 10k test examples. We followed the network architectures used in [2]. The largest model trained used 3 layers of stochastic latent variables, h^1, h^2 and h^3, of sizes 50, 100 and 100 nodes going from bottom to top. All mappings were implemented with MLPs with two layers of deterministic hidden units, with 100 nodes per layer.

The models were trained end to end using the Adam [5] optimizer with mini batches of 20. We tested both the VAE and IWAE models with different numbers of samples, $k \in \{1, 5, 50\}$. The models were implemented using the Theano framework, with code from Burda et al [3] adapted to include tracking of the number of effective samples and to allow configurations with more than 2 stochastic layers.

The deterministic layers used the `tanh` nonlinearity. The stochastic layers used Gaussian distributions with diagonal covariance, and the visible layer used Bernoulli distributions. An `exp` nonlinearity was applied to the predicted variances of the Gaussian distributions. The same learning schedule as in [2] was used where training proceeded for 3^i passes over the data with learning

[1] $D_\alpha(p||q) = \frac{1}{\alpha(1-\alpha)}\int_h[\alpha p(h) + (1-\alpha)q(h)] - p^\alpha(h)q^{(1-\alpha)}(h)dh$

rate of $0.001 * 10^{\frac{0i}{7}}$ for $i = 0 \ldots 7$.

All log-likelihood values are estimated as the mean of $\mathcal{L}_{5000}$ on the test set.

5 Results and Discussion

5.1 Generative log-likelihood performance

Figure 3 illustrates our results from the recreation of Burda et al's MNIST experiment. Their results showed the IWAE improved significantly on VAE log-likelihood scores with higher values of k increasing the advantage even further. This was expected, as each weighted sample contributed to a tighter estimate of the lower-bound. Indeed, it appears that further gains in performance could yet be made by increasing k further as the log-likelihood shows no sign of plateauing for the two layer IWAE. This would be at the cost of a linear increase in computational cost. Use of a second stochastic layer was found to be of much greater benefit to the IWAE than the VAE. This may be due to the IWAE using a less constraining form of objective function, allowing a greater degree of the network's modelling capacity to be exploited, thus producing richer latent space representations. Our results are compatible to those presented in the paper to within a small margin of error caused by random variability.

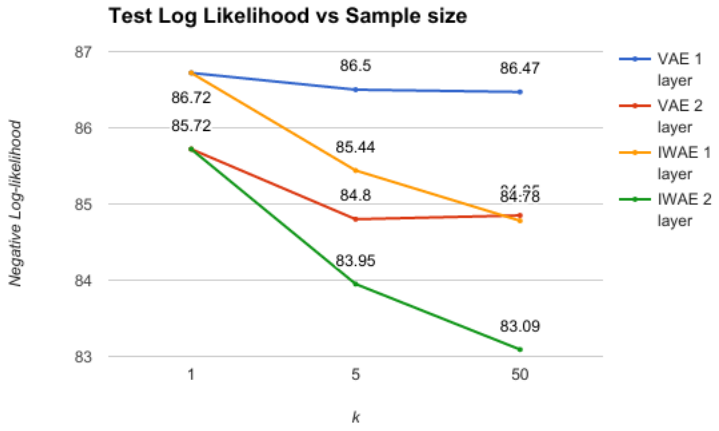
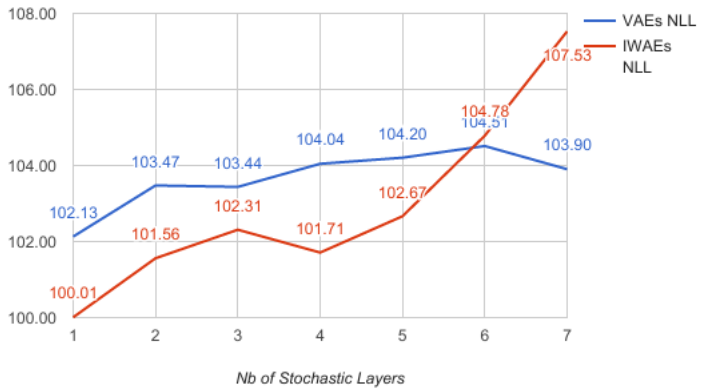


Figure 1: Our recreation of Burda et al's experiments. The graph shows test log-likelihoods for single and double layer VAEs and IWAEs under different sampling rates.

5.2 Increasing the number of stochastic layers

To examine the impact of increasing number of stochastic layers, we trained the VAE and IWAE models with k = 50, for up to 7 stochastic layers. Due to time constraints, we only trained the models for up to 3 epochs. Results are shown in Figure 2.



(a)

Figure 2: Results on density estimation for VAE and IWAE models with 1 to 7 stochastic layers, trained for 1 then 3 epochs, with 50 samples. The log-likelihood values were estimated as the mean of $\mathcal{L}_{5000}$ on the test set.

Figure 2 shows that the NLL increases for increasing number of stochastic layers for IWAE models with more than 4 stochastic layers. No trend can be observed from the results for the VAE model. The results obtained with 1 and 2 layers are not consistent with the ones observed earlier, in that the models with 2 layers achieved a higher NLL than the models with 1 layer, when trained for 1 then 3 epochs. Given these discrepancies, and due to the limiting nature of the experiments, we cannot make generalisations from these results. Nonetheless, many issues can arise with increasing depth in neural networks. This includes a vanishing gradient descent that prevents learning, and the increase of the possibilities of the gradient descent getting stuck at saddle points.

5.3 Latent Space Representation

Both VAEs and IWAEs tend to learn representations with effective dimensions far below their capacity, with few stochastic latent variables propagating useful information. The activity of a latent dimensions u is measured using the statistic:

$$A_u = Cov_x(E_{u \sim q(u|x)}[u]) \tag{16}$$

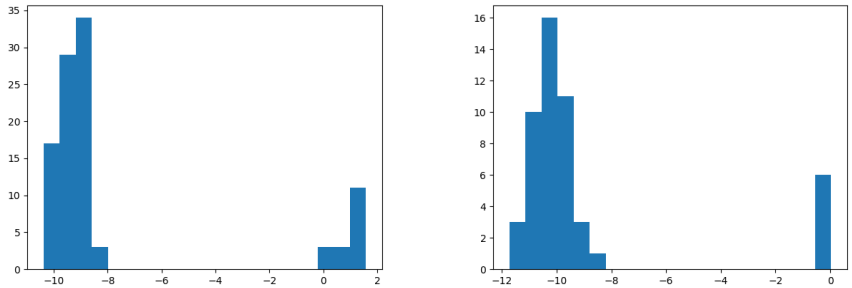
A dimension u is active when $A_u > 10^{-2}$. The figures below (Figures 3 and 4) show the histogram of the activity of units for a VAE and a IWAE with 2 layers and 50 samples. The number of active units for each stochastic layer for the different configurations we attempted in our experiments are incorporated in Tables 1 and 2.

k	VAE	IWAE
1	19/100	19/100
5	19/100	23/100
50	19/100	25/100

Table 1: Single layer active units under different sampling schemes.

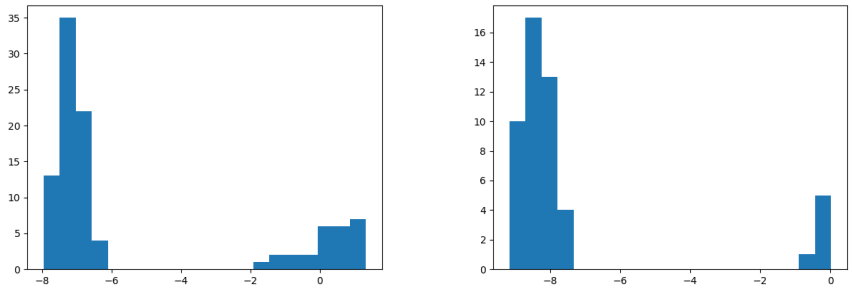
k	VAE L1	VAE L2	IWAE L1	IWAE L2
1	16/100	3/50	16/100	3/50
5	17/100	4/50	21/100	6/50
50	17/100	6/50	26/100	6/50

Table 2: Secondary layer active units under different sampling schemes. VAE vs IWAE



(a) (b)

Figure 3: Distribution of activity statistic for the VAE model with 2 stochastic layers. Histogram of $\log A_u$ for (a) the first layer and (b) the second layer.



(a) (b)

Figure 4: Distribution of activity statistic for the IWAE model with 2 stochastic layers. Histogram of $\log A_u$ for (a) the first layer and (b) the second layer.

For both the VAEs and IWAEs and for all configurations, the number of active dimensions was less than the total number of dimensions (50). We also notice that both VAEs and IWAEs use the lower layers the most, as the number of active dimensions decreases with higher layers. The IWAE model learned more latent dimensions than the VAE, and larger values of k with IWAE resulted in a more distributed latent representation. The higher log-likelihood obtained with IWAEs suggest that a larger number of active dimensions reflects richer latent representation.

Inactive dimensions can be related to units not propagating a gradient signal. The problem of inactive units can result from an optimization issue or a modelling issue. The issue of optimization can be resolved with a better initialisation, or by perturbing the gradient descent with Gaussian noise [11].

To determine whether the inactive units resulted from an optimization issue or a modelling issue, Burda et al took the VAE and IWAE models with 2 layers and k = 50, and continued training the VAE models using the IWAE objective and vice versa. They found that training with the VAE objective actively reduced both the number of active dimensions and the log-likelihood, suggesting that inactivation of the latent dimensions is driven by the objective functions rather than by optimizations [2].

5.4 Warm-Up

Weighting the KL divergence term by a variable parameter β can dictate the extent to which gradients are driven by pure deterministic reconstruction error and the variational regularisation term given by the KL divergence. We intended to follow the results of [3] by implementing 'warm up', a technique that gradually increases β from 0 to 1 over the course of training. Previous results with VAEs and Ladder VAEs show that as the variational regularization term is introduced, many active units are gradually pruned away. However, at the end of training, warm-up resulted in more active units indicating a more distributed representation and improving performance over the regular VAEs [14]. This method does not work however with IWAEs because the KL divergence term cannot be separated in the objective function for $k > 1$, as done by Kingma and Welling [1].

5.5 Effective Sample Size

Importance sampling uses unequally weighted observations. One of the w_i may be (vastly) larger than all the others reducing the sample size to effectively one sample. To measure how many of the k samples are effective samples i.e. samples that get into high density regions of the posterior, we consider [12]:

$$k_e = \frac{(\sum_{i=1}^{n} w_i)^2}{\sum_{i=1}^{n} w_i^2} \tag{17}$$

where k_e is the number of effective samples. If the weights are too imbalanced then the result is similar to averaging only $k_e << k$ observations. We use the same method to compute the effective sample size for the VAE, where we compute the weights but do not use them for the computation of the gradient.

The variance of importance sampling estimate of the gradient can also be estimated and used to assess when the weights are problematic. A large variance indicates the importance sampling had not worked well. Unfortunately the variance estimate is itself based on the same weights that the estimate has used. Badly skewed weights could give a badly estimated mean along with a bad variance estimate that masks the problem.

Figure 5 shows the number of effective samples during training for both the IWAE and VAE models with 3 stochastic layers and $k = 50$. We show the number of effective samples according to equation 17. We also looked at the number of samples whose weights are above a threshold of 10^{-2}. Figure 6 shows the number of effective samples for the VAE and IWAE models trained with up to 7 stochastic layers, with $k = 50$, and 3 epochs. The results in these two figures show the average number of effective samples per datapoint, per epoch. Figure 7 shows the distribution of weights after training a VAE and an IWAE for 243 epochs with k = 50 and 3 stochastic layers. The figure includes the distribution of weights of samples for each minibatch (50 samples per datapoint for 20 datapoints for 243 epochs).

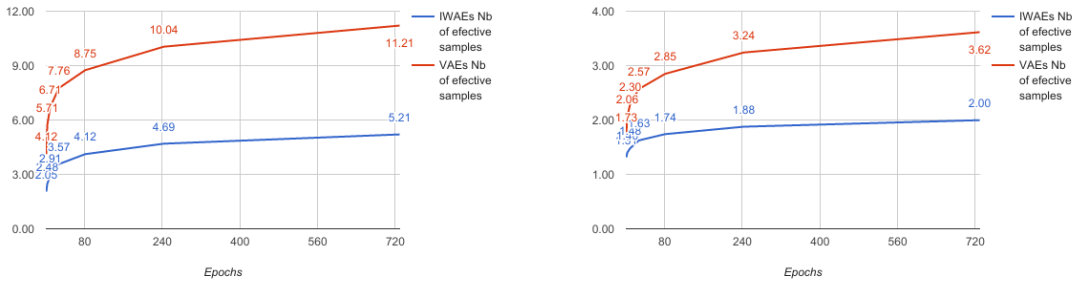
Tracking the number of effective samples show the following:

1. The number of effective samples n_e is far below 50 for all settings of both the VAE and the IWAE. For n_e to be higher, q(x) must be approximately proportional to p(x) for most x. This suggests that the small n_e is an issue of modelling where the choice of a Gaussian for q(x) might not match the structural properties of the target density p(x) [13]. This observation is consistent with the phenomenon of inactive units as discussed in section 5.3.

2. As the number of epochs is increased, the average number of samples per epoch increases.

3. However, the number of effective samples is higher for VAEs than IWAEs, and the difference in numbers between VAEs and IWAEs increases with the number of epochs.

4. The number of effective samples decreases with increasing number of stochastic layers. This might be the reason why all units for all stochastic layers above layer 2 are “dead units”



(a)

Figure 5: Number of effective samples during training, as a function of the number of epochs. The configuration with $k = 50$, $l = 3$ was used. (a) Number of samples with weights above a threshold of 10^{-2}. (b) Number of effective samples according to equation 17.

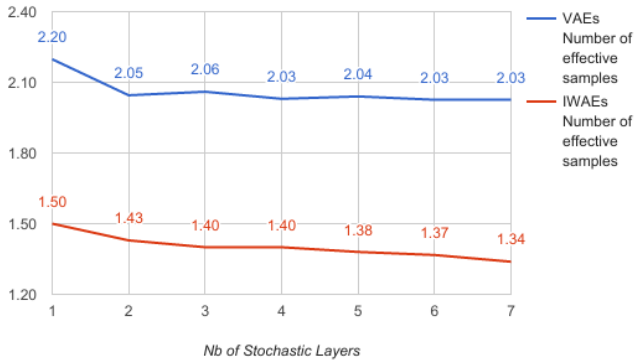
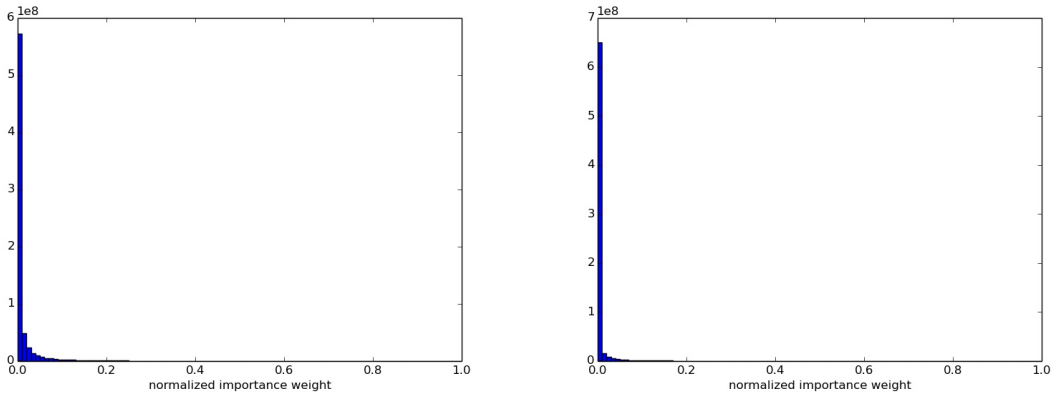


Figure 6: Number of effective samples for different number of stochastic layers. The configuration with $k = 50$ and the equation 17 for estimating the number of effective samples were used.



(a)

Figure 7: Distribution of weights after training (a) a VAE and (b) an IWAE for 243 epochs with $k = 50$ and 3 stochastic layers.

6 Future Work

Inactive dimensions and a low number of effective samples can be related to an issue with modelling where the choice of q(x) may not provide a good approximation to p(x). It could be interesting to combine several importance distributions through adaptive parametric importance sampling. Importance sampling from a mixture of m sampling densities with m control variates, one for each mixture component can be used [16]. The method of control variates uses knowledge of one integral to reduce the variance in the estimate of another. With this approach, at least one of the chosen $q_i(x)$ is expected to lead to an efficient importance sample for the estimate of the gradient. One can simply use mixture importance sampling as well where a mixture distribution $q_\alpha(x) = \sum_{j=1}^{J} \alpha_j q_j(x)$ with $\alpha_j \geq 0$, $\sum_{j=1}^{J} \alpha_j = 1$ can be used. Mixtures of unimodal densities provide a flexible approximation to the target density p(x) which could be a multimodal density [12].

7 Conclusion

We have reviewed the contributions made by Burda et al in their work on IWAEs and recreated key experiments. We provided further analysis of their work by measuring the activity of stochastic units with increasing k and measuring the effective sample size. It was found that despite the modifications of the IWAE to improve the lower-bound, neither method fully exploited the modelling capacity of the network and the effective sample size was significantly lower than k (true sample size). The inactivity of units was found to worsen in deeper layers - perhaps as a result of vanishing gradients - but was alleviated with increasing k, improving the tightness of the lower-bound. In general, the number of effective samples was higher in the VAE than the IWAE, yet despite this the IWAE produced better log-likelihood scores and richer latent-space representations.

References

[1] Kingma, Welling "Auto-Encoding Variational Bayes" arXiv:1312.6114v10 (2014)

[2] Burda, Yuri, Roger Grosse, and Ruslan Salakhutdinov. "Importance weighted autoencoders." arXiv preprint arXiv:1509.00519 (2015).

[3] Burda, Yuri, Roger Grosse, and Ruslan Salakhutdinov. Code to train Importance Weighted Autoencoders on MNIST and OMNIGLOT. https://github.com/yburda/iwae

[4] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

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

[6] Mnih, A. and Gregor, K. Neural variational inference and learning in belief networks. In International Conference on Machine Learning, pp. 1791–1799, 2014.

[7] Bornschein, J. and Bengio, Y. Reweighted wake-sleep. International Conference on Learning Representations, 2015.

[8] J. M. Hernandez–Lobato Slides on Alpha-divergence minimization for Bayesian deep learning. 2013.

[9] Gal, Yarin. "Uncertainty in Deep Learning." PhD diss., PhD thesis, University of Cambridge, 2016.

[10] John Paisley, David Blei, and Michael Jordan. Variational Bayesian inference with stochastic search. ICML, 2012.

[11] Jin, Chi, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. "How to Escape Saddle Points Efficiently." arXiv preprint arXiv:1703.00887 (2017).

[12] Owen, Art B. "Monte Carlo theory, methods and examples." Monte Carlo Theory, Methods and Examples. Art Owen (2013).

[13] Tokdar, Surya T., and Robert E. Kass. "Importance sampling: a review." Wiley Interdisciplinary Reviews: Computational Statistics 2, no. 1 (2010): 54-60.

[14] Sønderby, Raiko et al "Ladder Variational Autoencoders" arXiv:1602.02282v3 May 2016

[15] Balázs Csanád Csáji (2001) Approximation with Artificial Neural Networks; Faculty of Sciences; Eötvös Loránd University, Hungary

[16] Owen, Art, and Yi Zhou. "Safe and effective importance sampling." Journal of the American Statistical Association 95, no. 449 (2000): 135-143.