arXiv:1602.02282v1 [stat.ML] 6 Feb 2016

How to Train Deep Variational Autoencoders and Probabilistic Ladder Networks

Casper Kaae Sønderby[1] CASPERKAAE@GMAIL.COM
Tapani Raiko[2] TAPANI.RAIKO@AALTO.FI
Lars Maaløe[3] LARSMA@DTU.DK
Søren Kaae Sønderby[1] SKAAESONDERBY@GMAIL.COM
Ole Winther[1,3] OLWI@DTU.DK

[1]Bioinformatics Centre, Department of Biology, University of Copenhagen, Denmark
[2]Department of Computer Science, Aalto University, Finland
[3]Department of Applied Mathematics and Computer Science, Technical University of Denmark

Abstract

Variational autoencoders are a powerful framework for unsupervised learning. However, previous work has been restricted to shallow models with one or two layers of fully factorized stochastic latent variables, limiting the flexibility of the latent representation. We propose three advances in training algorithms of variational autoencoders, for the first time allowing to train deep models of up to five stochastic layers, (1) using a structure similar to the Ladder network as the inference model, (2) warm-up period to support stochastic units staying active in early training, and (3) use of batch normalization. Using these improvements we show state-of-the-art log-likelihood results for generative modeling on several benchmark datasets.

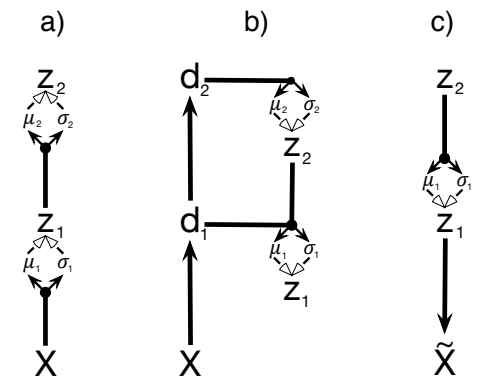

Figure 1. Inference (or encoder/recognition) and generative (or decoder) models. a) VAE inference model, b) Probabilistic ladder inference model and c) generative model. The $\mathbf{z}$'s are latent variables sampled from the approximate posterior distribution q with mean and variances parameterized using neural networks.

1. Introduction

The recently introduced variational autoencoder (VAE) (Kingma & Welling, 2013; Rezende et al., 2014) provides a framework for deep generative models (DGM). DGMs have later been shown to be a powerful framework for semi-supervised learning (Kingma et al., 2014; Maaloee et al., 2016). Another line of research starting from denoising autoencoders introduced the Ladder network for unsupervised learning (Valpola, 2014) which have also been shown to perform very well in the semi-supervised setting (Rasmus et al., 2015).

Here we study DGMs with several layers of latent variables, each conditioned on the layer above, which allows highly flexible latent distributions. We study two different model parameterizations: the first is a simple extension of the VAE to multiple layers of latent variables and the second is parameterized in such a way that it can be regarded as a probabilistic variational variant of the Ladder network which, contrary to the VAE, allows interactions between a *bottom up* and *top-down* inference signal.

Previous work on DGMs have been restricted to shallow models with one or two layers of stochastic latent variables constraining the performance by the restrictive mean field approximation of the intractable posterior distribution. Using only gradient descent optimization we show that DGMs are only able to utilize the stochastic latent variables in the second layer to a limited degree and not at all in the third

Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

layer or above. To alleviate these problems we propose (1) the probabilistic ladder network, (2) a warm-up period to support stochastic units staying active in early training, and (3) use of batch normalization, for optimizing DGMs and show that the likelihood can increase for up to five layers of stochastic latent variables. These models, consisting of deep hierarchies of latent variables, are both highly expressive while maintaining the computational efficiency of fully factorized models. We first show that these models have competitive generative performance, measured in terms of test log likelihood, when compared to equally or more complicated methods for creating flexible variational distributions such as the Variational Gaussian Processes (Tran et al., 2015) Normalizing Flows (Rezende & Mohamed, 2015) or Importance Weighted Autoencoders (Burda et al., 2015). We find that batch normalization and warm-up always increase the generative performance, suggesting that these methods are broadly useful. We also show that the probabilistic ladder network performs as good or better than strong VAEs making it an interesting model for further studies. Secondly, we study the learned latent representations. We find that the methods proposed here are necessary for learning rich latent representations utilizing several layers of latent variables. A qualitative assessment of the latent representations further indicates that the multi-layered DGMs capture high level structure in the datasets which is likely to be useful for semi-supervised learning.

In summary our contributions are:

- A new parametrization of the VAE inspired by the Ladder network performing as well or better than the current best models.
- A novel warm-up period in training increasing both the generative performance across several different datasets and the number of active stochastic latent variables.
- We show that batch normalization is essential for training VAEs with several layers of stochastic latent variables.

2. Methods

Variational autoencoders simultaneously train a generative model $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})$ for data $\mathbf{x}$ using auxiliary latent variables $\mathbf{z}$, and an inference model $q_\phi(\mathbf{z}|\mathbf{x})$[1] by optimizing a variational lower bound to the likelihood $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z})d\mathbf{z}$.

[1]The inference models is also known as the recognition model or the encoder, and the generative model as the decoder.

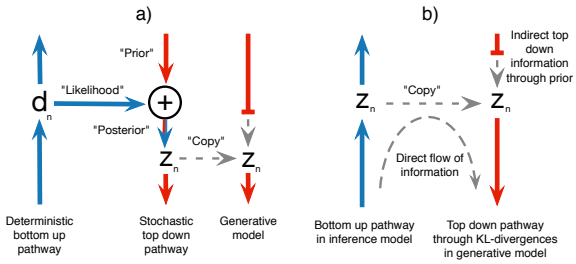

Figure 2. Flow of information in the inference and generative models of a) probabilistic ladder network and b) VAE. The probabilistic ladder network allows direct integration (+ in figure, see Eq. (21)) of bottom-up and top-down information in the inference model. In the VAE the top-down information is incorporated indirectly through the conditional priors in the generative model.

The generative model p_θ is specified as follows:

$$p_\theta(\mathbf{x}|\mathbf{z}_1) = \mathcal{N}\left(\mathbf{x}|\mu_\theta(\mathbf{z}_1), \sigma^2_\theta(\mathbf{z}_1)\right) \text{ or} \quad (1)$$

$$P_\theta(\mathbf{x}|\mathbf{z}_1) = \mathcal{B}\left(\mathbf{x}|\mu_\theta(\mathbf{z}_1)\right) \quad (2)$$

for continuous-valued (Gaussian $\mathcal{N}$) or binary-valued (Bernoulli $\mathcal{B}$) data, respectively. The latent variables $\mathbf{z}$ are split into L layers $\mathbf{z}_i$, $i = 1 \ldots L$:

$$p_\theta(\mathbf{z}_i|\mathbf{z}_{i+1}) = \mathcal{N}\left(\mathbf{z}_i|\mu_{\theta,i}(\mathbf{z}_{i+1}), \sigma^2_{\theta,i}(\mathbf{z}_{i+1})\right) \quad (3)$$

$$p_\theta(\mathbf{z}_L) = \mathcal{N}\left(\mathbf{z}_L|0, \mathbf{I}\right) \ . \quad (4)$$

The hierarchical specification allows the lower layers of the latent variables to be highly correlated but still maintain the computational efficiency of fully factorized models.

Each layer in the inference model $q_\phi(\mathbf{z}|\mathbf{x})$ is specified using a fully factorized Gaussian distribution:

$$q_\phi(\mathbf{z}_1|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}_1|\mu_{\phi,1}(\mathbf{x}), \phi^2_{\phi,1}(\mathbf{x})\right) \quad (5)$$

$$q_\phi(\mathbf{z}_i|\mathbf{z}_{i-1}) = \mathcal{N}\left(\mathbf{z}_i|\mu_{\phi,i}(\mathbf{z}_{i-1}), \sigma^2_{\phi,i}(\mathbf{z}_{i-1})\right) \quad (6)$$

for $i = 2 \ldots L$.

Functions $\mu(\cdot)$ and $\sigma^2(\cdot)$ in both the generative and the inference models are implemented as:

$$\mathbf{d}(\mathbf{y}) = \texttt{MLP}(\mathbf{y}) \quad (7)$$

$$\mu(\mathbf{y}) = \texttt{Linear}(\mathbf{d}(\mathbf{y})) \quad (8)$$

$$\sigma^2(\mathbf{y}) = \texttt{Softplus}(\texttt{Linear}(\mathbf{d}(\mathbf{y}))) \ , \quad (9)$$

where `MLP` is a two layered multilayer perceptron network, `Linear` is a single linear layer, and `Softplus` applies $\log(1 + \exp(\cdot))$ non linearity to each component of its argument vector. In our notation, each $\texttt{MLP}(\cdot)$ or $\texttt{Linear}(\cdot)$ gives a new mapping with its own parameters, so the deterministic variable $\mathbf{d}$ is used to mark that the `MLP`-part is shared between μ and σ^2 whereas the last `Linear` layer is not shared.

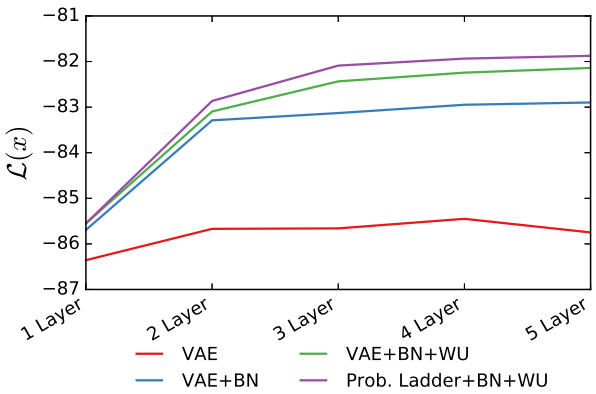

Figure 3. MNIST test-set log-likelihood values for VAEs and the probabilistic ladder networks with different number of latent layers, Batch normalization*BN* and Warm-up *WU*

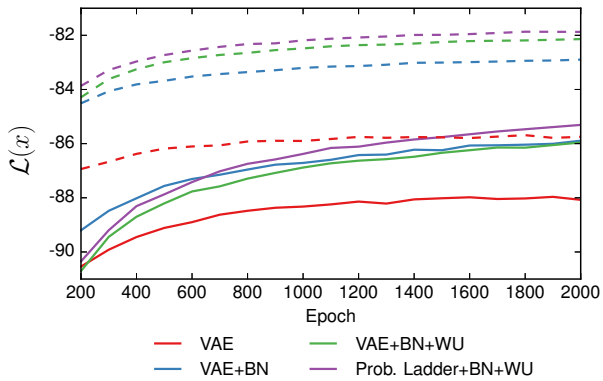

Figure 4. MNIST train (*full lines*) and test (*dashed lines*) performance during training. The test set performance was estimated using 5000 importance weighted samples providing a tighter bound than the training bound explaining the better performance here.

The variational principle provides a tractable lower bound on the log likelihood which can be used as a training criterion $\mathcal{L}$.

$$\log p(\mathbf{x}) \geq E_{q_\phi(z|x)}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = -\mathcal{L}(\theta,\phi;\mathbf{x}) \quad (10)$$

$$= -KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) + E_{q_\phi(\mathbf{z}|\mathbf{x})}\left[p_\theta(\mathbf{x}|\mathbf{z})\right], \quad (11)$$

where *KL* is the Kullback-Leibler divergence.

A strictly tighter bound on the likelihood may be obtained at the expense of a K-fold increase of samples by using the importance weighted bound (Burda et al., 2015):

$$\log p(\mathbf{x}) \geq E_{q_\phi(z^{(1)}|x)} \dots E_{q_\phi(z^{(K)}|x)}\left[\log \sum_{k=1}^{K} \frac{p_\theta(\mathbf{x},\mathbf{z}^{(\mathbf{k})})}{q_\phi(\mathbf{z}^{(\mathbf{k})}|\mathbf{x})}\right]$$

$$\geq -\mathcal{L}(\theta,\phi;\mathbf{x}) \,. \quad (12)$$

The inference and generative parameters, θ and ϕ, are jointly trained by optimizing Eq. (11) using stochastic gradient descent where we use the reparametrization trick for stochastic backpropagation through the Gaussian latent variables (Kingma & Welling, 2013; Rezende et al., 2014). All expectations are approximated using Monte Carlo sampling by drawing from the corresponding q distribution.

Previous work has been restricted to shallow models ($L \leq 2$). We propose two improvements to VAE training and a new variational model structure allowing us to train deep models with up to at least $L = 5$ stochastic layers.

2.1. Probabilistic Ladder Network

We propose a new inference model where we use the structure of the Ladder network (Valpola, 2014; Rasmus et al., 2015) as the inference model of a VAE, as shown in Figure 1. The generative model is the same as before.

The inference is constructed to first make a deterministic upward pass:

$$\mathbf{d}_1 = \texttt{MLP}(\mathbf{x}) \quad (13)$$

$$\mu_{d,i} = \texttt{Linear}(\mathbf{d}_i), i = 1 \dots L \quad (14)$$

$$\sigma^2_{d,i} = \texttt{Softplus}(\texttt{Linear}(\mathbf{d}_i)), i = 1 \dots L \quad (15)$$

$$\mathbf{d}_i = \texttt{MLP}(\mu_{d,i-1}), i = 2 \dots L \quad (16)$$

followed by a stochastic downward pass:

$$q_\phi(\mathbf{z}_L|\mathbf{x}) = \mathcal{N}\left(\mu_{d,L}, \sigma^2_{d,L}\right) \quad (17)$$

$$\mathbf{t}_i = \texttt{MLP}(\mathbf{z}_{i+1}), i = 1 \dots L-1 \quad (18)$$

$$\mu_{t,i} = \texttt{Linear}(\mathbf{t}_i) \quad (19)$$

$$\sigma^2_{t,i} = \texttt{Softplus}(\texttt{Linear}(\mathbf{t}_i)) \quad (20)$$

$$q_\theta(\mathbf{z}_i|\mathbf{z}_{i+1},\mathbf{x}) = \mathcal{N}\left(\frac{\mu_{t,i}\sigma^{-2}_{t,i} + \mu_{d,i}\sigma^{-2}_{d,i}}{\sigma^{-2}_{t,i} + \sigma^{-2}_{d,i}}, \frac{1}{\sigma^{-2}_{t,i} + \sigma^{-2}_{d,i}}\right). \quad (21)$$

Here μ_d and σ^2_d carry the *bottom-up* information and σ^2_t and μ_t carry *top-down* information. This parametrization has a probabilistic motivation by viewing μ_d and σ^2_d as the Gaussian likelihood that is combined with a Gaussian prior μ_t and σ^2_t from the top-down connections, together forming the approximate posterior distribution $q_\theta(\mathbf{z}_i|\mathbf{z}_{i+1},\mathbf{x})$, see Figure 2 a).

A line of motivation, already noted by Dayan et al. (1995), is that a purely bottom-up inference process as in i.e. VAEs

does not correspond well with real perception, where iterative interaction between bottom-up and top-down signals produces the final activity of a unit[2], see Figure 2. Notably it is difficult for the purely bottom-up inference networks to model the *explaining away* phenomenon, see van den Broeke (2016, Chapter 5) for a recent discussion on this phenomenon. The probabilistic ladder network provides a framework with the wanted interaction, while keeping complications manageable. A further extension could be to make the inference in k steps over an iterative inference procedure (Raiko et al., 2014).

2.2. Warm-up from deterministic to variational autoencoder

The variational training criterion in Eq. (11) contains the reconstruction term $p_\theta(\mathbf{x}|\mathbf{z})$ and the variational regularization term. The variational regularization term causes some of the latent units to become inactive during training (MacKay, 2001) because the approximate posterior for unit k, $q(z_{i,k}|\dots)$ is regularized towards its own prior $p(z_{i,k}|\dots)$, a phenomenon also recognized in the VAE setting (Burda et al., 2015). This can be seen as a virtue of automatic relevance determination, but also as a problem when many units are pruned away early in training before they learned a useful representation. We observed that such units remain inactive for the rest of the training, presumably trapped in a local minima or saddle point at $KL(q_{i,k}|p_{i,k}) \approx 0$, with the optimization algorithm unable to re-activate them.

We propose to alleviate the problem by initializing training using the reconstruction error only (corresponding to training a standard deterministic auto-encoder), and then gradually introducing the variational regularization term:

$$-\mathcal{L}(\theta, \phi; \mathbf{x})_T = \\ -\beta KL(q_\phi(z|x)||p_\theta(\mathbf{z})) + E_{q_\phi(z|x)}\left[p_\theta(\mathbf{x}|\mathbf{z})\right], \tag{22}$$

where β is increased linearly from 0 to 1 during the first N_t epochs of training. We denote this scheme *warm-up* (abbreviated *WU* in tables and graphs) because the objective goes from having a delta-function solution (corresponding to zero temperature) and then move towards the fully stochastic variational objective. A similar idea has previously been considered in Raiko et al. (2007, Section 6.2), however here used for Bayesian models trained with a coordinate descent algorithm.

2.3. Batch Normalization

Batch normalization (Ioffe & Szegedy, 2015) is a recent innovation that improves convergence speed and stabilizes training in deep neural networks by normalizing the outputs from each layer. We show that batch normalization (abbreviated *BN* in tables and graphs), applied to all layers except the output layers, is essential for learning deep hierarchies of latent variables for $L > 2$.

[2]The idea was dismissed at the time, since it could introduce substantial theoretical complications.

Table 1. Fine-tuned test log-likelihood values for 5 layered VAE and probabilistic ladder networks trained on MNIST. *ANN. LR:* Annealed Learning rate, *MC:* Monte Carlo samples to approximate $E_{q(\cdot)}[\cdot]$, *IW:* Importance weighted samples

FINETUNING	NONE	ANN. LR. MC=1 IW=1	ANN. LR. MC=10 IW=1	ANN. LR. MC=1 IW=10	ANN. LR. MC=10 IW=10
VAE	−82.14	−81.97	−81.84	−81.41	−81.30
PROB. LADDER	−81.87	−81.54	−81.46	−81.35	**-81.20**

3. Experiments

To test our models we use the standard benchmark datasets MNIST, OMNIGLOT (Lake et al., 2013) and NORB (LeCun et al., 2004). The largest models trained used a hierarchy of five layers of stochastic latent variables of sizes 64, 32, 16, 8 and 4, going from bottom to top. We implemented all mappings using two-layered MLP's. In all models the MLP's between x and z_1 or d_1 were of size 512. Subsequent layers were connected by MLP's of sizes 256, 128, 64 and 32 for all connections in both the VAE and probabilistic ladder network. Shallower models were created by removing latent variables from the top of the hierarchy. We sometimes refer to the five layer models as 64-32-16-8-4, the four layer models as 64-32-16-8 and so fourth. The models were trained end-to-end using the Adam (Kingma & Ba, 2014) optimizer with a mini-batch size of 256. The reported test log-likelihoods were approximated using Eq. (12) with 5000 importance weighted samples as in Burda et al. (2015). The models were implemented in the Theano (Bastien et al., 2012), Lasagne (Dieleman et al., 2015) and Parmesan[3] frameworks.

For MNIST we used a sigmoid output layer to predict the mean of a Bernoulli observation model and leaky rectifiers ($\max(x, 0.1x)$) as nonlinearities in the MLP's. The models were trained for 2000 epochs with a learning rate of 0.001 on the complete training set. Models using warm-up used $N_t = 200$. Similarly to Burda et al. (2015) we resample the binarized training values from the real-valued images using a Bernoulli distribution after each epoch which prevents the models from over-fitting. Some of the models were fine-tuned by continuing training for 2000 epochs while multiplying the learning rate with 0.75 after every 200 epochs and increase the number of Monte Carlo and

[3]github.com/casperkaae/parmesan

Table 2. Number of active latent units in five layer VAE and probabilistic ladder networks trained on MNIST. A unit was defined as active if $KL(q_{i,k}||p_{i,k}) > 0.01$

	VAE	VAE +BN	VAE +BN +WU	PROB. LADDER +BN +WU
LAYER 1	20	20	34	46
LAYER 2	1	9	18	22
LAYER 3	0	3	6	8
LAYER 4	0	3	2	3
LAYER 5	0	2	1	2
TOTAL	21	37	61	81

importance weighted samples to 10 to reduce the variance in the approximation of the expectations in Eq. (10) and improve the inference model, respectively.

Models trained on the OMNIGLOT dataset[4], consisting of 28x28 binary images images were trained similar to above except that the number of training epochs was 1500.

Models trained on the NORB dataset[5], consisting of 32x32 grays-scale images with color-coding rescaled to $[0, 1]$, used a Gaussian observation model with mean and variance predicted using a linear and a softplus output layer respectively. The settings were similar to the models above except that: *hyperbolic tangent* was used as nonlinearities in the MLP's, the learning rate was 0.002, $N_t = 1000$ and the number of training epochs were 4000.

4. Results & Discussion

To asses the performance of the VAE and the probabilistic ladder networks with different numbers of stochastic latent variable layers and with or without batch normalization and warm-up we use the MNIST, OMNIGLOT and NORB datasets. Firstly we study the generative model log-likelihood for the different datasets and show that models using our contributions perform better than the vanilla VAE models[6]. Secondly we study the latent space representation and how batch normalization, warm-up and the probabilistic ladder network affect these compared to the vanilla VAE.

Generative log-likelihood performance

In Figure 3 we show the test set log-likelihood for a series of different models with varying number of stochastic layers. The performance of the vanilla VAE model did not improve with more than two layers of stochastic latent variables. Contrary to this, models trained with batch normalization and warm-up consistently increase the model performance for additional layers of stochastic latent variables. As expected the improvement in performance is decreasing for each additional layer, but we emphasize that the improvements are consistent even for the addition of the top-most layers. The best performance is achieved using the probabilistic ladder network slightly outperforming the best VAE models trained with batch normalization and temperature. We emphasize that these VAE models already have a very strong performance showing that the probabilistic ladder network have competitive generative performance.

Table 3. Test set Log-likelihood values for models trained on the OMNIGLOT and NORB datasets. The left most column show dataset and the number of latent variables i each model.

	VAE	VAE +BN	VAE +BN +WU	PROB. LADDER +BN +WU
OMNIGLOT				
64	-114.45	-108.79	-104.63	–
64-32	-112.60	-106.86	-102.03	-102.12
64-32-16	-112.13	-107.09	-101.60	**-101.26**
64-32-16-8	-112.49	-107.66	-101.68	-101.27
64-32-16-8-4	-112.10	-107.94	-101.86	-101.59
NORB				
64	2630.8	3263.7	3481.5	–
64-32	2830.8	3140.1	**3532.9**	3522.7
64-32-16	2757.5	3247.3	3346.7	3458.7
64-32-16-8	2832.0	3302.3	3393.6	3499.4
64-32-16-8-4	3064.1	3258.7	3393.6	3430.3

The models in Figure 3 were trained using fixed learning rate and one Monte Carlo (MC) and one importance weighted (IW) sample. To improve performance we fine-tuned the best performing five layer models by training these for a further 2000 epochs with annealed learning rate and increasing the number of MC and IW samples. In Table 1 we see that that by annealing the learning rate, drawing 10 MC and IW samples we can improve the log-likelihood of the VAE and probabilistic ladder network to -81.30 and -81.20 respectively.

Comparing the results obtained here with current state-of-the art results on permutation invariant MNIST, Burda et al. (2015) report a test-set performance of -82.90 using 50 IW samples and a two layer model with 100-50 latent units. Using a similar model Tran et al. (2015) achieves -81.90 by adding a variational Gaussian Process prior for each layer of latent variables, however here we note that our results are not directly comparable to these due to differences

[4]The OMNIGLOT data was partitioned and preprocessed as in Burda et al. (2015), https://github.com/yburda/iwae/tree/master/datasets/OMNIGLOT

[5]The NORB dataset was downloaded in resized format from github.com/gwtaylor/convnet_matlab

[6]we use vanilla to refer to VAE models with out batch normalization and warm-up

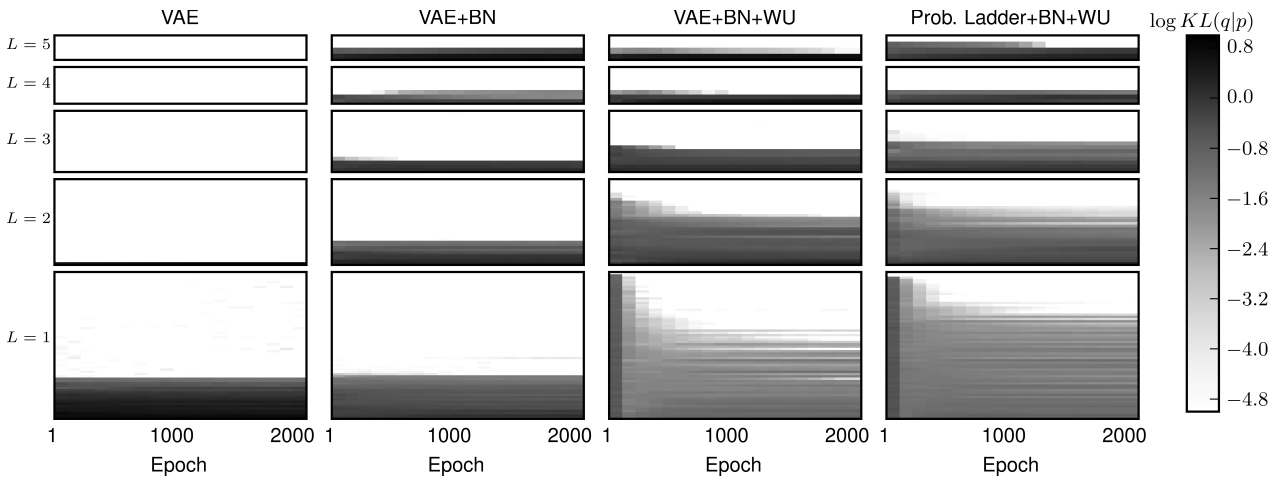

Figure 5. $\log KL(q|p)$ for each latent unit is shown at different training epochs. Low KL (white) corresponds to an inactive unit. The units are sorted for visualization. It is clear that vanilla VAE cannot train the higher latent layers, while introducing batch normalization helps. Warm-up creates more active units early in training, some of which are then gradually pruned away during training, resulting in a more distributed final representation. Lastly, the probabilistic ladder network activates the highest number of units in each layer. The division into active and inactive units at the end of the training is rather clear in most cases.

in the training procedure.

As shown above we found batch normalization (Ioffe & Szegedy, 2015) to be essential for achieving good performance for DGMs with $L > 2$. Figure 4 shows the MNIST training and test convergence for models with five layers of stochastic units. Comparing the VAE model with and without batch normalization we see a large increase in both trained and test performance. Adding warm-up does not increase the training performance however the models generalize better indicating a more robust latent representation. Lastly the probabilistic ladder network converges to slightly better values on both the test and training sets. We saw no signs of over-fitting for any of our models even though the hierarchical latent representations are highly expressive.

To test the models on more challenging data we used the OMNIGLOT dataset, consisting of characters from 50 different alphabets with 20 samples of each character. The Log-likelihood values, Table 3, shows similar trends as for MNIST with substantial increases in performance by adding batch-normalization and warm-up and again with the probabilistic ladder network performing slightly better than the other models. For both VAEs and probabilistic ladder networks the performance increases up to three layers of stochastic latent variables after which no gain is seen by adding more layers. The best Log-likelihood results obtained here, -101.26, is higher than the best results from Burda et al. (2015) at -103.38, which were obtained using more latent variables (100-50 vs 64-32-16) and 50 importance weighted samples for training.

We tested the models using a continuous Gaussian observation model on the NORB dataset consisting of gray-scale images of 5 different toy objects under different illuminations and observation angles. As seen in Table 3 we again find that bath normalization and warm-up increases performance. However here we do not see a gain in performance for $L > 2$ layers of latent stochastic variables. We found the Gaussian observation models to be harder to train requiring lower learning rates and *hyperbolic tangent* instead of leaky rectifiers as nonlinearities for stable training. The probabilistic ladder network were sometimes unstable during training explaining the high variance in the results. The harder optimization of the Gaussian observation model, also recognized in Oord et al. (2016), might explain the lower utilization of the topmost latent layers in these models.

The parametrization of the probabilistic ladder network allows for natural parameter sharing between top-down part $\mu_{t,i}(\mathbf{z}_{i+1})$ and $\sigma^2_{t,i}(\mathbf{z}_{i+1})$ of the inference model q_ϕ, and the top-down mapping $\mu_{\theta,i}(\mathbf{z}_{i+1})$ and $\sigma^2_{\theta,i}(\mathbf{z}_{i+1})$ of the generative model p_θ in Equation (3). We did initial experiments with this setup but did not see improvements in performance. However we believe that parameter sharing between the inference and generative model is an interesting future research direction.

Altogether for permutation invariant MNIST and OMNIGLOT we show state-of-the-art generative performance achieved using a deep hierarchy of latent variables consisting of fewer units (124 compared to 150) than current best methods. We show similar findings on the NORB dataset

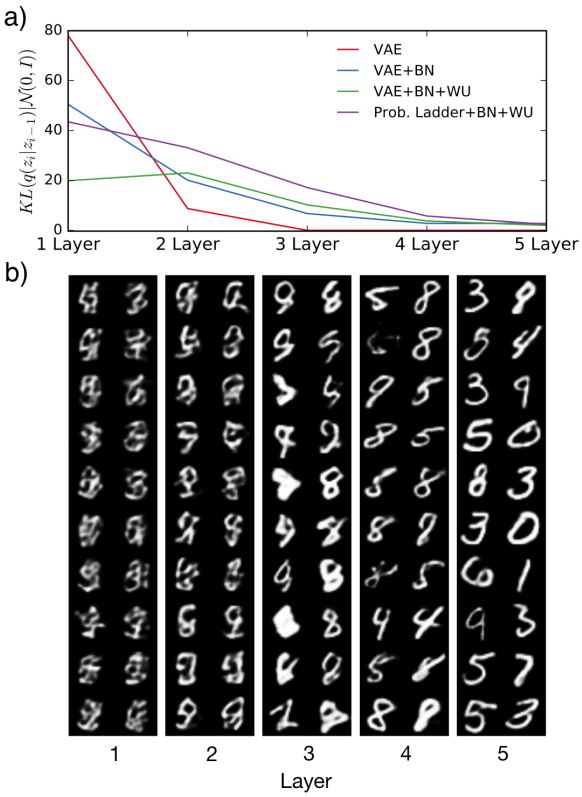

Figure 6. a) MNIST test set KL-divergence between $q(z_i|z_{i-1})$ and $\mathcal{N}(0, \mathbf{I})$ in each layer of four different five layer models. The KL-divergence is large for the lower layers indicating highly non (standard) Gaussian distributions. b) Generative model outputs created by injecting samples $z^* \sim \mathcal{N}(0, \mathbf{I})$ at different layers in the probabilistic ladder network. As expected the samples get progressively better as the sampling is moved up in the hierarchy.

using the more challenging continuous Gaussian observation model.

Latent representations

The probabilistic generative models studied here automatically tune the model complexity to the data by reducing the effective dimension of the latent representation due to the regularization effect of the priors in Eq. (10). However, as previously identified (Raiko et al., 2007; Burda et al., 2015), the latent representation is often overly sparse with few stochastic latent variables propagating useful information.

To study this effect we calculated the KL-divergence between $q(z_{i,k}|z_{i-1,k})$ and $p(z_i|z_{i+1})$ for each stochastic latent variable k during training as seen in Figure 5. This term is zero if the inference model is independent of the data, i.e. $q(z_{i,k}|z_{i-1,k}) = q(z_{i,k})$, and hence collapsed

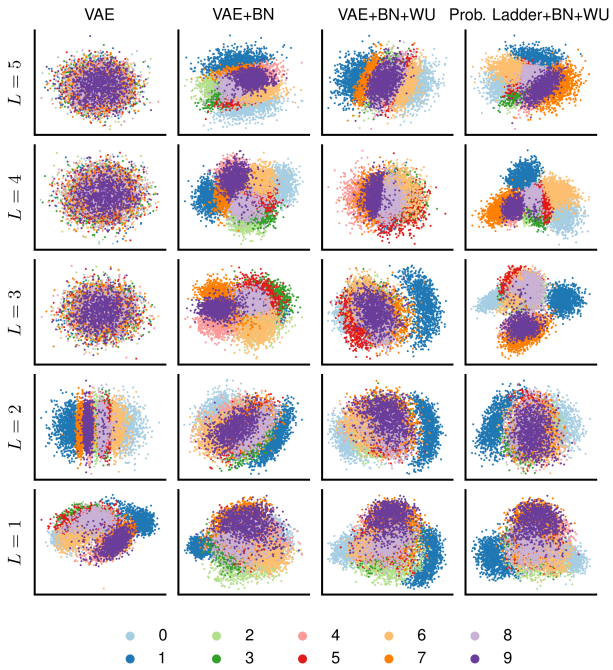

Figure 7. PCA-plots of samples from $q(z_i|z_{i-1})$ for five layer VAE and probabilistic ladder networks trained on MNIST. In the vanilla VAE the stochastic units above the second layer is clearly seen to be inactive with no dependency on the the input data. For the other models a clear structure according to class is seen for the higher layers.

onto the prior carrying no information about the data. For the models without warm-up we find that the KL-divergence for each unit is stable during all training epochs with only very few new units activated during training. For the models trained with warm-up we initially see many active units which are then gradually pruned away as the variational regularization term is introduced. At the end of training warm-up results in more active units indicating a more distributed representation. This effect is quantified in Table 2 where we defined a stochastic latent unit $z_{i,k}$ to be active if $KL(q_{i,k}||p_{i,k}) > 0.01$. We find that batch normalization is essential for activating latent units above the second layer but find that it does not affect the number of active units in the first layer. Warm-up causes a large increase in the number of active stochastic latent units, especially in the lower layers. We can explain this observation from the structure of the VAE. For $\beta = 0$, i.e. fully deterministic training, the stochastic latent layers above the first layer are effectively disconnected from the model since the variational regularization term is ignored. This causes all information to flow through the lowest stochastic layer z_1 as illustrated in Figure 2 b). We further find that the probabilistic ladder network activates the most stochastic latent units. We speculate that this is due to the deterministic path

from the input data to all latent distributions in the inference model.

To qualitatively study the latent representations, PCA plots of $z_i \sim q(z_i|z_{i-1})$ are seen in Figure 7. It is clear that the VAE model without batch normalization is completely collapsed on a standard normal prior above the second layer. The latent representations shows progressively more clustering according to class which is best seen in the topmost layer of the probabilistic ladder network. These findings indicate that hierarchical latent variable models produce structured high level latent representations that are likely useful for semi-supervised learning.

The hierarchical latent variable models used here allows highly flexible distributions of the lower layers conditioned on the layers above. We measure the divergence between these conditional distributions and the restrictive mean field approximation by calculating the KL-divergence between $q(z_i|z_{i-1})$ and a standard normal distribution for several models trained on MNIST, see Figure 6 a). As expected the lower layers have highly non (standard) Gaussian distributions when conditioned on the layers above. Interestingly the probabilistic ladder network seems to have more active intermediate layers than t,he VAE with batch normalization and warm-up. Again this might be explained by the deterministic upward pass easing flow of information to the intermediate and upper layers. We further note that the KL-divergence is approximately zero in the vanilla VAE model above the second layer confirming the inactivity of these layers. Figure 6 b) shows generative samples from the probabilistic ladder network created by injecting $z^* \sim \mathcal{N}(0, \mathbf{I})$ at different layers in the model. Sampling from the top-most layer, having a standard normal prior, produces nice looking samples whereas the samples get progressively worse when we inject z^* in the lower layers supporting the finding above.

5. Conclusion

We presented three methods for optimization of VAEs using deep hierarchies of latent stochastic variables. Using these models we demonstrated state-of-the-art generative performance for the MNIST and OMNIGLOT datasets and showed that a new parametrization, the probabilistic ladder network, performed as good or better than current models. Further we demonstrated that VAEs with up to five layer hierarchies of active stochastic units can be trained and that especially the new probabilistic ladder network were able to utilize many stochastic units. Future work includes extending these models to semi-supervised learning and studying more elaborate iterative inference schemes in the probabilistic ladder network.

Acknowledgments

This research was supported by the Novo Nordisk Foundation, Danish Innovation Foundation and the NVIDIA Corporation with the donation of TITAN X and Tesla K40 GPUs.

References

Bastien, Frédéric, Lamblin, Pascal, Pascanu, Razvan, Bergstra, James, Goodfellow, Ian, Bergeron, Arnaud, Bouchard, Nicolas, Warde-Farley, David, and Bengio, Yoshua. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.

Burda, Yuri, Grosse, Roger, and Salakhutdinov, Ruslan. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Dayan, Peter, Hinton, Geoffrey E, Neal, Radford M, and Zemel, Richard S. The Helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

Dieleman, Sander, Schlter, Jan, Raffel, Colin, Olson, Eben, ren Kaae Sø nderby, Sø, Nouri, Daniel, van den Oord, Aaron, and and, Eric Battenberg. Lasagne: First release., August 2015. URL http://dx.doi.org/10.5281/zenodo.27878.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Kingma, Diederik P, Mohamed, Shakir, Rezende, Danilo Jimenez, and Welling, Max. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pp. 3581–3589, 2014.

Lake, Brenden M, Salakhutdinov, Ruslan R, and Tenenbaum, Josh. One-shot learning by inverting a compositional causal process. In *Advances in neural information processing systems*, pp. 2526–2534, 2013.

LeCun, Yann, Huang, Fu Jie, and Bottou, Leon. Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pp. II–97. IEEE, 2004.

Maaloee, Lars P, Sønderby, Casper Kaae, Sønderby, Søren Kaae, and Winther, Ole. Improving semi-supervised learning with auxiliary deep generative models. *arXiv preprint arXiv:1312.6114*, 2016.

MacKay, David JC. Local minima, symmetry-breaking, and model pruning in variational free energy minimization. *Inference Group, Cavendish Laboratory, Cambridge, UK*, 2001.

Oord, Aaron van den, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

Raiko, Tapani, Valpola, Harri, Harva, Markus, and Karhunen, Juha. Building blocks for variational bayesian learning of latent variable models. *The Journal of Machine Learning Research*, 8:155–201, 2007.

Raiko, Tapani, Li, Yao, Cho, Kyunghyun, and Bengio, Yoshua. Iterative neural autoregressive distribution estimator nade-k. In *Advances in Neural Information Processing Systems*, pp. 325–333, 2014.

Rasmus, Antti, Berglund, Mathias, Honkala, Mikko, Valpola, Harri, and Raiko, Tapani. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pp. 3532–3540, 2015.

Rezende, Danilo Jimenez and Mohamed, Shakir. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Tran, Dustin, Ranganath, Rajesh, and Blei, David M. Variational gaussian process. *arXiv preprint arXiv:1511.06499*, 2015.

Valpola, Harri. From neural pca to deep unsupervised learning. *arXiv preprint arXiv:1411.7783*, 2014.

van den Broeke, Gerben. What auto-encoders could learn from brains - generation as feedback in unsupervised deep learning and inference, 2016.