Evaluating NVAE on MNIST

Amy Jinxin Chen **David Stevens** **Malcolm Bagger Toräng**

Abstract

Variational autoencoders provide an architecture for generative modeling that has a number of benefits, such as an explicit latent space that can be explored. However, this has historically come at the cost of the quality of the generated samples. Recently, [1] introduced a deep hierarchical VAE called NVAE that bridges this gap. However, it was only quantitatively evaluated using the negative log likelihood (NLL) of the test data. This paper extends the work of [1] by evaluating NVAE with the metrics FID, precision and recall on the MNIST data set. NVAE is shown to perform competitively also on these sample based metrics. This paper also performs an ablation study evaluating the use of spectral normalization (SN) [2] instead of spectral regularization (SR) [3], as well as performing the KL-divergence warmup factor as a function increasing once per epoch rather than in each step. While epoch-based warmup is shown to perform worse than step-based, the choice between SN and SR with fixed hyperparameters from [1] is shown to be a trade-off where SN performs better in FID but worse in NLL. The source code for the TensorFlow implementation is available at `https://github.com/stevensdavid/nvae-tf`

1 Introduction

While the field of generative models for image generation has recently been dominated by generative adverserial networks (GANs) and autoregressive models, variational autoencoders (VAEs) provide some additional benefits by explicitly modeling the latent space. However, this has historically come at the cost of the generated images not meeting the same quality standard. NVAE took an important step for VAE-based models as it helped close this gap to GANs and autoregressive models in the image generation domain [1]. Furthermore, NVAE advanced the state-of-the-art for VAEs on images of size 256x256, using CelebA HQ and FFHQ containing images of human faces. The performance of NVAE was measured quanitatively through negative log-likelihood (NLL), which acts as a density estimation metric.

However, NLL only measures the density estimation performance of the model and says nothing of the quality of the generated samples. Several attempts at evaluating generative models with other metrics have been made, for example attempting to estimate humanly perceived image similarity [4, 5], thus enabling testing a model with real images. For instance, many metrics utilize the latent embeddings of models such as VGG-16 as they've been shown to enforce humanly perceived similarity in the input images [6]. Besides the isolated quality in images, there are more aspects of a generative model to evaluate such as latent space sensitivity or coverage of the training space [7, 8]. In related work popular metrics of evaluation include FID, PR and PPL, defined in section 2.3. PPL, for instance, was introduced in [8] for evaluating latent space sensitivity and disentanglement. Used together, these metrics measure properties of a generative model not captured by NLL and as a result, facilitate more nuanced comparisons to other models as well as offering a deeper insight into the behavior of the generative model.

Project report for DD2412 or FDD3412

2 Method

2.1 Data

The dataset we use throughout this research is MNIST [9] that contains 70 000 samples of handwritten digits from 0 to 9. The samples are grayscale images of size 28x28x1. The training set consists of 60 000 samples and the remaining samples were used for testing. This dataset was downloaded with TensorFlow with existing train/test split. Training was done with a binarized version of the dataset, where binary samples were drawn from the corresponding Bernoulli distribution where the grayscale values are interpreted as probabilities.

The MNIST dataset has been widely used as a benchmarking dataset within machine learning. The current best performing network for NLL is the Locally Masked PixelCNN [10] that achieved 77.58 nats. Some other networks that achieve state-of-the-art results are the normalizing flow based networks Sliced Iterative Generator [11] and Generative Latent Flow [12] that attained 5.5 and 5.8 in FID score, respectively. Generative Latent Flow also reached a precision of 0.981 and recall of 0.963.

2.2 Implementation

The authors of NVAE make use of three techniques for mitigating the unbounded and unstable hierarchical KL loss. Normalizing flows are proposed for increasing expressiveness in the hierarchical groups. We have omitted these to simplify implementation and as it was found to not provide significant benefits by [1]. Furthermore, the approximate posterior $q(z|x)$ is modeled as a residual, or relative, Normal distribution. The residual Normal distribution alters the KL loss term in that the scaling and location of the posterior, relative to prior, is the main contributing factor. Granted relatively modeling the posterior is an easier task than doing so absolutely, the KL loss might prove less unstable. Another important step for overall training stability, is the balancing of the KL terms during a warm-up training-phase. The effect of balancing is increased for each training step during warm-up.

Spectral Regularization (SR) is also used for taming the unbounded KL term. As originally shown in [3], SR minimizes the model sensitivity to input perturbation and is calculated using power iteration as in [3, 2]. SR is not to be confused with Spectral Normalization (SN) [2] as although both techniques are based on minimizing the spectral norm, SN considers the spectral norm of each layer activation while SR only works with the weights. In common between SR and SN, is that the potential insensitivity in latent codes should stabilize the KL training.

The NVAE network was implemented using the machine learning library Tensorflow. The overall architecture is illustrated in figure 2.2. The input to the network are raw MNIST images of size 28x28x1. The output has the same size, but they are instead reconstructions in logits. The preprocessing step contains two preprocessing blocks and three preprocessing cells within each blocks, where each cell contains batch normalizations, swish activations, convolutional layers followed by squeeze and excitation, and a skip connection. This step converts the input samples to the size 32x32x3. The postprocess instead reverts the samples to the original input size with three postprocessing blocks where each block contains two postprocessing cells followed by an Exponential Linear Unit (ELU) activation and a convolutional layer. The postprocessing cells contain same type of layers as the preprocecing cells but in different order.

Each hierarchical group contains one or more residual cells, denoted with m, which in our case is 1. This model contains 15 hierarchical groups hence produces 15 latent variables z. Each group belongs to one latent variable scale. A scale is the spatial dimensionality of z. There are two scales used for MNIST. The five bottom groups have 4^2 and the latter 10 groups 8^2 in scales for z. After the last group in each scale, the dimensionality is rescaled by convolutional layers to the desired dimensions.

With the hyperparameter settings presented by [1], an ablation study was done on three models. One was trained using SR and step-based KL warm-up. The two others were trained using SN with either step-based or epoch-based warm-up. The networks were trained for 400 epochs[1] and minimizes the loss described in equation 1 with model parameters θ, original sample x and the reconstruction $\tilde{x}$. The loss function is composed by three terms: a mean drawn from reconstruction loss and KL loss, batch normalization loss and SR loss scaled by λ. The λ was fixed to 0.01. For models with SN, the

[1]Except the step-based SN model which was trained for 370 epochs due to an unexpected crash.

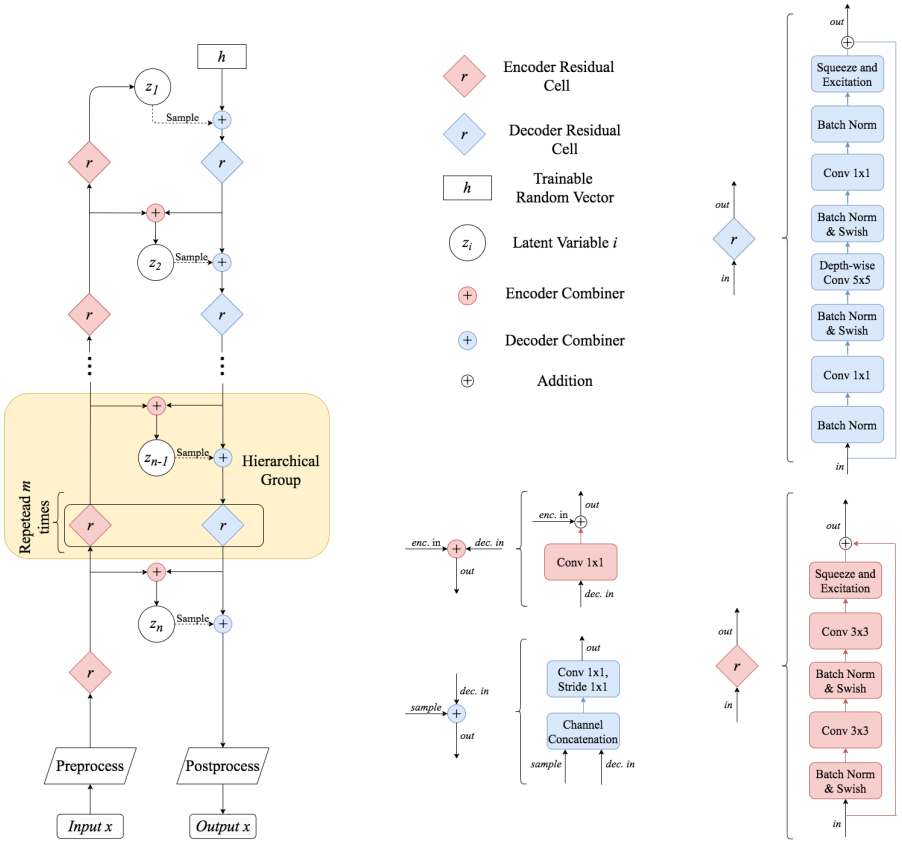

Figure 1: Illustration of n-group hierarchical VAE with m residual cells per group.

SR term is replaced by applying a spectral normalization layer after each convolutional layer. During training, KL warm-up was ran for the first 30% of training.

$$L_{NVAE}(x, \tilde{x}, \theta) = \frac{L_{recon}(x, \tilde{x}) + L_{KL}(\theta)}{2} + \lambda L_{BN}(\theta) + \lambda L_{SR}(\theta) \tag{1}$$

2.3 Evaluation

Following are a set of popular metrics chosen for evaluating our NVAE implementation. The metrics cover quality in generated samples, sensitivity in latent space, comprehensiveness of samples in representing the prior space, among others. This section moreover goes into implementation detail of these evaluation metrics.

2.3.1 Negative log-likelihood

The negative log-likelihood (NLL) is a density estimation metric that gives the likelihood that an image would have been generated by the model. By calculating the NLL of the held-out test dataset, it can be used as a metric for how well the model has learned the data distribution.

2.3.2 Precision and recall

Precision and recall (PR) is typically used in discriminative learning for evaluating how reliable positive classifications are and to which degree the model handles the complete space of positive cases. PR has an intuitive translation for generative models. Precision would for a generative model denote the resemblance between generated and real samples, whereas recall captures how well the training data in its entirety is covered.

Figure 2: Sampled digits with temperature 1 from the step-based SN model. The digits are uncurated apart from picking the first sample of each digit as the model is unconditional.

It's clear from above description there may be several implementation interpretations possible. We've used the implementation from Karras et al. [7] available at `https://github.com/kynkaat/improved-precision-and-recall-metric`. The proposed algorithm aims to estimate the manifolds of the prior, i.e. the complete space of input samples, and of the generated samples. The manifolds can then be used to measure precision and recall analogously to the discriminative case. Approximating the manifold is accomplished with pair-wise Euclidean distances between VGG-16 embeddings to find the k-th closest neighbor of each such embedding, the distance to which is the radius of a sphere defining part of the approximate manifold. With two approximate manifolds of generated and true samples, we may now measure precision and recall

2.3.3 Fréchet Inception Distance

The dependency on human evaluation for generative models is highlighted in [13]. As a solution, the authors propose the Inception score (IS) they found to correlate well with human evaluation. IS is calculated using a pre-trained Inception model [5] from which generated samples Inception class posteriors $p(y|x)$ are used to measure meaningfulness ($p(y|x)$ with low entropy) and diversity ($p(x)$ with high entropy).

Fréchet Inception Distance (FID) extends IS by involving test data of real samples [4]. The main idea is to gather the Inception pool layer representations, instead of Inception-v3 as in 2.3.2, and to model these as multivariate Gaussians; if the Fréchet distance between these distributions is low it means the Inception model perceive the real and generated samples as similar.

3 Experiments

Table 1: Performance metrics on binarized MNIST. NLL and PR are calculated over ten passes of the test split. FID is calculated over 10000 samples with temperature 1, i.e. a standard normal prior distribution. Arrows indicate if lower or higher values are better.

Model	NLL (nats) ↓	FID ↓	Precision ↑	Recall ↑	Training time (h) ↓
Step + SN	$87.06(\pm 2.18)$	8.87	$0.8950(\pm 0.0999)$	$0.9227(\pm 0.0879)$	49[2]
Step + SR	$80.33(\pm 2.01)$	30.37	$0.8559(\pm 0.0608)$	$0.8803(\pm 0.0546)$	104
Epoch + SN	$98.92(\pm 1.83)$	20.85	$0.7541(\pm 0.152)$	$0.8828(\pm 0.114)$	71[3]
Vanilla VAE	86.76[14]	$28.2(\pm 0.3)$[12]	-	-	-
NVIDIA's NVAE w/o flow [1]	78.01	-	-	-	-
NVIDIA's NVAE w/ flow [1]	78.19	-	-	-	-
GLF [12]	-	$5.8(\pm 0.1)$	0.981	0.963	-
PixelCNN [15]	81.30 [10]	-	-	-	-
LMCONV [10]	77.58	-	-	-	-

As shown in table 3, the step-based model with SR which was implemented exactly as the NVAE paper[1] has achieved a NLL score very close to their result of 78.01. The SN models performed worse in this metric. However, SN improved the FID performance both for the step-based and epoch-based models relative to SR, and the step-based SN model was the best performing of the three for both FID, precision and recall. Ten samples from the step-based SN model representing each digit 0-9 are shown in figure 2.

[2]Linearly extrapolated from the average epoch time during the 370 training epochs.

[3]Training suddenly slowed down by 50% after approximately 150 epochs for unknown reasons. This should likely be closer to step-based SN.

4 Discussions

The difference between our results with SR and those of the source NVAE paper [1] may be due to our binarization of the MNIST dataset being applied a single time for each image as a preprocessing step, while the NVIDIA implementation applies this binarization each time an image is drawn from the dataset. As the binarization is a random sampling from the Bernoulli distribution that the original gray scale image forms, performing the binarization each time the image is drawn will lead to a slightly different image each time and can act as a form of data augmentation. Another reason could be because of the reconstruction loss during our training was calculated for the padded images while the [1] removed the padding which could have affected the optimization.

Using an epoch-based KL-divergence warmup was shown to lead to a large decrease in performance in all metrics compared to step-based warmup with the same architecture, and should therefore be avoided. An important observation is the considerably reduced training time through the use of spectral normalization, which in itself is a large benefit of the architecture. While using SR was the superior architecture as measured by NLL, the SN models outperform the SR model for the other metrics that were used. This indicates a trade-off between density estimation performance and generated sample quality, which is supported by [16]. NLL is a metric for density estimation, while the remaining metrics measure properties more relevant for the generative process. As the hyperparameters were tuned by [1] in favor of NLL, we need to take into consideration that this observed difference in generative and density estimation performance, might result from the SR-tailored hyperparameters not being as appropriate for our SN models or for the sampling task.

However, as [1] show that the performance of NVAE is not drastically affected by the removal of SR meaning the hyperparameters should be fair for SN. [1] also specify that their hyperparameter search was not extensive, so the hyperparameters are assumed to not be excessively specialized for NLL at the cost of the other metrics. We therefore assume that the largest difference in performance is due to the change in the architecture, and not due to the hyperparameters being less tuned for certain metrics or models.

4.1 Challenges

4.1.1 Implementation

Implementation of the NVAE architecture was a significant challenge. Perhaps the largest challenge that was faced was that NVAE is a heavily optimized architecture with many aspects of the implementation which are not listed in the paper. Some examples of this include the pre- and post-process blocks which are not present at all in the paper despite introducing new hyperparameters and multiple new types of layers.

While the official NVAE implementation is publicly available on GitHub, it is written in a highly specifically optimized manner for distributed computation across multiple GPUs. In fact, we were not successful in running the code on our development environments with single GPUs. The code is also written in an excessively abstracted dynamic manner, where it is very difficult to follow the implementation when reading the code. This lead to a large number of minor errors being discovered only after investing time into training, something that became especially troublesome as training was very computationally intensive.

4.1.2 Computational Costs

The computational requirements also lead us to not be able to perform any hyperparameter searching, so the used hyperparameters were those that were proposed by [1]. These hyperparameters are optimized for the spectral regularization case, so the spectral normalization based models may perform better with different hyperparameters.

Precision and Recall [7] proved to be, relative to other used evaluation metrics, a computationally heavy task. For 10 attempts providing new model-sampled images, PR used the entire MNIST test data of 10 000 images. In total, over 10 hours were required for each such evaluation. PPL usually required around 5 hours. Working with these time scales proved a challenge as errors were discovered in our implementations. This further on affected our chances at reducing variance in test results, as even more time would be needed.

4.1.3 PPL

The perceptual path length metric that was proposed by [8] has as far as we know not been applied on MNIST before our work. This makes it difficult to draw any conclusions regarding NVAE's performance with PPL, as we have no baseline to compare to. Furthermore, the values that were found were orders of magnitude larger than what is observed when PPL is measured on datasets such as CelebA. This might in turn be explained by our slight variation to the originally proposed PPL method, in that the VGG-16 activations are directly used for measuring the euclidean distance, without the specifically tuned attached head that [8] use to make a distance metric that is more in line with human perception.

Table 2: PPL calculated over 1000 samples on MNIST. Lower is better.

Model	PPL $\downarrow$
Step + SN	$7.351 * 10^6(\pm 6.31 * 10^5)$
Step + SR	$7.391 * 10^6(\pm 5.88 * 10^5)$
Epoch + SN	$6.602 * 10^6(\pm 5.30 * 10^5)$

4.2 Ethical considerations

Being a variational autoencoder, NVAE has the attractive quality of having a continuous latent space that can be explored in order to gain a deeper understanding of why the model generated the data that it did. This increased transparency is a key attribute towards helping machine learning applications in society meet SDG 16.6 for increasing transparency and accountability in institutions. In theory, variational autoencoders are also more resilient towards mode collapse than GANs. This is due to the fact that a VAE must learn to represent the entire input space, while a GAN only has to learn to fool the discriminator and can focus on generating a single image that is successful at this task. The societal impact of this can be enormous, as reducing bias can be a key tool in reducing inequality within and between countries where generative models are applied for key tasks (SDG 10) and also reduce inequality between genders (SDG 5).

Our proposed extension of using SN instead of SR lead to a large decrease in training time, and therefore power consumption. This directly benefits SDG 7.3 regarding increasing energy efficiency, as this extension allows potentially greater business value while still having lower energy consumption. Furthermore, decreased computational costs make the technology more accessible to all, which can help promote the use of machine learning in developing countries and reduce inequality within and among countries (SDG 10).

4.3 Self Assessment

We have successfully re-implemented NVAE without normalizing flows. Our experiments reproduced similar NLL values to [1], and we performed an ablation study with variations that investigate the influence of different components and sensitivity to the design choices. These contributions should satisfy the requirements for a good-very good project. Moreover, this paper goes beyond the scope and provides results for four additional evaluation metrics: FID, PPL (slightly simplified), Precision and Recall. In the ablation study, the modification of KL warm-up to be epoch-based and introduction of SN are both extensions to the source paper. Our NVAE implementation in TensorFlow also merits a bonus as there is no public implementation in the framework yet. Furthermore, the complexity of this network architecture and the heavily optimized details of the already many components increases the implementation difficulty significantly, and hence should also be a bonus for this project. The broader impacts of this project have also been investigated thoroughly. These aforementioned extensions together with this written report should bring this study to an excellent project of grade A.

5 Conclusions

NVAE is a generative architecture that we have shown performs competitively in sample-based metrics such as FID with other alternatives while maintaining many of the benefits of being a VAE. Our results indicate that by adjusting the architecture slightly through the use of spectral normalization,

the quality of the samples can be increased as a trade-off against the density estimation performance without changing hyperparameters, while gaining substantial improvements in computational costs. As such, the architecture can be customized through small changes in order to better match the intended application.

The most important next step is to increase the confidence of the results by performing a larger number of tests in order to decrease the variance and a proper hyperparameter search for each architecture to ensure optimal performance. Furthermore, our code base should be extendable to other data sets where the performance of NVAE can be analyzed in other settings such as the CelebA dataset of human faces.

References

[1] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

[2] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[3] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[6] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[7] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, pages 3927–3936, 2019.

[8] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[9] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[10] Ajay Jain, Pieter Abbeel, and Deepak Pathak. Locally masked convolution for autoregressive models. In *Conference on Uncertainty in Artificial Intelligence*, pages 1358–1367. PMLR, 2020.

[11] Biwei Dai and Uros Seljak. Sliced iterative generator, 2020.

[12] Zhisheng Xiao, Qing Yan, and Yali Amit. Generative latent flow. *arXiv preprint arXiv:1905.10485*, 2019.

[13] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

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

[15] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

[16] Shuyu Lin, Stephen Roberts, Niki Trigoni, and Ronald Clark. Balancing reconstruction quality and regularisation in elbo for vaes, 2019.