Variational Lossy Autoencoder (VLAE)

Raghav Mishra

1 First Pass: Core Concepts

- The VLAE aims to extract **useful features** from observed data for downstream tasks such as classification.
- The **Variational Lossy Autoencoder (VLAE)** combines **VAEs** with **neural autoregressive models** (e.g., PixelCNN) to improve generative modeling performance.
- VLAE leverages autoregressive models as both the **prior distribution** $p(z)$ and the **decoding distribution** $p(x|z)$ to greatly improve generative modeling performance of VAEs.
- The model is designed to be a **lossy compressor** of observed data. The analysis identifies the conditions under which the latent code (z) in a VAE should be used for autoencoding.
- VLAE has the appealing properties of **controllable representation learning** and improved density estimation performance, allowing control over what the global latent code can learn.
- VLAE is **slower at generation** due to the sequential nature of the autoregressive model used in the decoder $p(x|z)$.

2 Second Pass: Key Results

- Achieved new state-of-the-art results on **MNIST**, **OMNIGLOT**, and **Caltech-101 Silhouettes** density estimation tasks, as well as competitive results on CIFAR10.

- The conditional distribution $p(x|z)$ is implemented with a small receptive-field **PixelCNN** (e.g., 6 layers of masked convolution with filter size 3).
- Reported marginal **Negative Log-Likelihood (NLL)** is estimated using Importance Sampling with 4096 samples.
- For the Statically Binarized MNIST model, the converged expected DKL is $\mathbb{E}[\mathrm{DKL}(q(z|x)\|p(z))] = 13.3$ nats $= 19.2$ bits.
- VLAE can learn a **lossier compression** than a VAE with a regular factorized conditional distribution.

Table 1: Statically Binarized MNIST: Model NLL Test Results (nats/dim)

Model	**NLL Test**
Normalizing flows (Rezende & Mohamed, 2015)	85.10
DRAW (Gregor et al., 2015)	< 80.97
Discrete VAE (Rolfe, 2016)	81.01
PixelRNN (van den Oord et al., 2016a)	79.20
IAF VAE (Kingma et al., 2016)	79.88
AF VAE	79.30
VLAE	**79.03**

Table 2: Dynamically Binarized MNIST: Model NLL Test Results (nats/dim)

Model	**NLL Test**
Convolutional VAE + HVI (Salimans et al., 2014)	81.94
DLGM 2hl + IWAE (Burda et al., 2015a)	82.90
Discrete VAE (Rolfe, 2016)	80.04
LVAE (Kaae Sønderby et al., 2016)	81.74
DRAW + VGP (Tran et al., 2015)	< 79.88
IAF VAE (Kingma et al., 2016)	79.10
Unconditional Decoder	87.55
VLAE	**78.53**

Table 3: OMNIGLOT: Model NLL Test Results (nats/dim)

Model	**NLL Test**
VAE (Burda et al., 2015a)	106.31
IWAE (Burda et al., 2015a)	103.38
RBM (500 hidden) (Burda et al., 2015b)	100.46
DRAW (Gregor et al., 2015)	< 96.50
Conv DRAW (Gregor et al., 2016)	< 91.00
Unconditional Decoder	95.02
VLAE	**90.98**
VLAE (fine-tuned)	**89.83**

Table 4: Caltech-101 Silhouettes: Model NLL Test Results (nats/dim)

Model	**NLL Test**
RWS SBN (Bornschein et al., 2014)	113.3
RBM (Cho et al., 2011)	107.8
NAIS NADE (Du et al., 2015)	100.0
Discrete VAE (Rolfe, 2016)	97.6
SpARN (Goessling et al., 2015)	88.48
Unconditional Decoder	89.26
VLAE	**77.36**

- VLAE was applied to the **CIFAR10** dataset of natural images.
- VLAE models attain new state-of-the-art performance among other variationally trained latent-variable models.
- The DenseNet VLAE model also outperforms most other tractable likelihood models including Gated PixelCNN and PixelRNN and has results only slightly worse than the then state-of-the-art PixelCNN++.

Table 5: CIFAR10: Test set bits/dim for various models. Likelihood for VLAE is approximated with 512 importance samples.

Method	**bits/dim**
Tractable likelihood models	
Uniform distribution [**?**]	8.00
Multivariate Gaussian [**?**]	4.70
NICE [**?**]	4.48
Deep GMMs [**?**]	4.00
Real NVP [**?**]	3.49
PixelCNN [**?**]	3.14
Gated PixelCNN [**?**]	3.03
PixelRNN [**?**]	3.00
PixelCNN++ [**?**]	2.92
Variationally trained latent-variable models	
Deep Diffusion [**?**]	5.40
Convolutional DRAW [**?**]	3.58
ResNet VAE with IAF [**?**]	3.11
ResNet VLAE	3.04
DenseNet VLAE	**2.95**

3 Third Pass: Model Theory

3.1 Introduction: Variational Inference and Coding

- The ultimate goal is for the model to uncover and untangle those causal sources of variations that are of interest.
- An autoregressive model of the data may achieve the same log-likelihood as a variational autoencoder (VAE).
- Notably, an autoregressive model has no stochastic latent variables at all.
- A VAE is frequently interpreted as a **regularized autoencoder**.
- Let x be the observed variables and z the latent variables. Let $p(x, z)$ be the parametric model of their joint distribution, called the **generative model**. Given

a dataset $X = \{x_1, \ldots, x_N\}$, we wish to perform maximum likelihood learning of its parameters:

$$\log p(X) = \sum_{i=1}^{N} \log p(x^{(i)}), \tag{1}$$

where $p(x^{(i)}) = \int p(x^{(i)}, z)\, dz$ is typically intractable for complex models, motivating the use of variational inference.

- Let $q(z|x)$ be a parametric inference model (also called the encoder or recognition model) defined over the latent variables. We optimize the **variational lower bound (ELBO)** on the marginal log-likelihood of each observation x:

$$\log p(x) \geq \mathbb{E}_{q(z|x)} \left[\log p(x, z) - \log q(z|x)\right] = \mathcal{L}(x; \theta) \tag{2}$$

where θ indicates the parameters of both the generative model p and the inference model q.

- The ELBO $\mathcal{L}(x; \theta)$ can be re-arranged:

$$\begin{aligned} \mathcal{L}(x; \theta) &= \mathbb{E}_{q(z|x)} \left[\log p(x, z) - \log q(z|x)\right] \\ &= \mathbb{E}_{q(z|x)} \left[\log p(x|z)\right] - \mathrm{DKL}\left(q(z|x) \| p(z)\right) \end{aligned} \tag{3}$$

- A more powerful $p(x|z)$ will make VAE's marginal generative distribution $p(x)$ more expressive.

- The issue of latent code collapse: when a very powerful autoregressive decoder (like an RNN) is used for $p(x|z)$, it can in theory model the entire data distribution $p(x)$ without using z. Consequently, z is sometimes completely ignored, and the model regresses to be a standard unconditional autoregressive distribution.

- The goal of designing an efficient coding protocol is to minimize the expected code length of communicating x.

- **Naive Two-Part Coding**: VAE implies a two-part code: $p(z)$ (essence/structure) and $p(x|z)$ (modeling error/deviation). The expected code length is:

$$C_{\text{naive}}(x) = \mathbb{E}_{x \sim \text{data},\, z \sim q(z|x)} \left[-\log p(z) - \log p(x|z)\right] \tag{4}$$

- In this scheme, information that can be modeled locally by the decoding distribution $p(x|z)$ without access to z will be encoded locally, and only the remainder (long-range dependency) will be encoded in z. This relates to the concept of **lossy compression** achieved by VLAE.

3.2 VLAE: Model Definition and Architecture

- VLAE presents two complementary classes of improvements to VAE that utilize autoregressive models to explicitly control representation learning and improve density estimation.

- **Conditional Decoder $p(x|z)$ with Restricted Receptive Field:**
  - Instead of conditioning the full decoder on z, VLAE restricts the receptive field of the autoregressive decoder to a small local window $x_{\text{WindowAround}(i)}$ around the pixel x_i: $p(x_i|x_{<i}, z) \approx p(x_i|x_{\text{WindowAround}(i)}, z)$.
  - Since $x_{\text{WindowAround}(i)}$ is smaller than $x_{<i}$, $p(x|z)$ won't be able to represent arbitrarily complex distributions over x without dependence on z, as not all distributions over x admit such factorizations with limited receptive fields.
  - **Feature Learning**: Local statistics of 2D images like texture will likely be modeled completely by the small local window. In contrast, global structural information of images (like shapes of objects) is a long-range dependency that **must** be communicated through the latent code z, thus forcing z to learn useful global representations.

- **Autoregressive Flow (AF) Prior $p(z)$:**
  - The VAE objective encourages the approximate posterior $q(z|x)$ to match the prior $p(z)$, leading to inefficient latent codes if $q(z|x)$ is too expressive. The mismatch between $q(z|x)$ and $p(z)$ can be exploited to construct a lossy code.
  - VLAE parametrizes the prior distribution $p(z)$ using an **Autoregressive Flow (AF)** from some simple noise source (e.g., spherical Gaussian), which can theoretically reduce inefficiency in Bits-Back coding.
  - **AF definition**: For an autoregressive flow f, a continuous noise source ϵ is transformed into the latent code z: $z = f(\epsilon)$. Assuming the density function for the noise source is $u(\epsilon)$, the log-prior is $\log p(z) = \log u(\epsilon) - \log \left|\det \frac{df^{-1}}{dz}\right|$. (The sign in the determinant log is often opposite to the noted $\log\det \frac{df}{d\epsilon}$ depending on convention).
  - **IAF/AF Equivalence**: The AF prior $p(z)$ is equivalent to using an **Inverse Autoregressive Flow (IAF)** to model the approximate posterior $q(z|x)$.
  - **IAF form**: $\epsilon_i = (z_i - \mu_i(z_{<i}))/\sigma_i(z_{<i})$, where $\mu_i(\cdot)$ and $\sigma_i(\cdot)$ are transformations generated autoregressively.

– **Benefit**: Using an AF prior makes the model more expressive without incurring extra cost during training time for the ELBO computation (under the expectation of $z \sim q(z|x)$) because the transformation complexity is the same as that of an IAF posterior.
– **Density Estimation**: The AF prior $p(z)$ is more expressive than a simple factorized prior, allowing the model to better match the aggregated posterior $q(z) = \int q(z|x)p_{\text{data}}(x)\,dx$ and thus improve density estimation performance.
– IAF posterior has a shorter decoder path $p(x|z)$ whereas AF prior has a deeper decoder path $p(x|f(\epsilon))$.