Chapter 5

Variational Autoencoders: theory and implementations

In this short chapter, we discuss the differences between the theoretical model and common implementations of VAEs. Those differences are there for a reason; they fix some problems that appear when implementing a VAE in its simplest form. First, we define the simple VAE and we provide a visualization of the problems this model suffers from. Second, we demonstrate empirically that common implementations manage to circumvent these problems. Third, we argue why these modifications empirically benefit the simple VAE based on the literature. Fourth, we demonstrate how these implementations do not respect the theory and why this is a problem. Finally, we propose potential solutions to those issues which would lead to an updated theory along with concordant implementations. However, those solutions have not been tested yet; we are demonstrating the existence of this gap between the theory and implementations to begin. In other words, only the first three steps of this project are completed at the moment of writing the thesis.

This chapter is not based on a published or submitted research paper yet; it is based on work that has begun a few years back and that is still ongoing. We are currently working on adapting the content of this chapter into publishable work. This chapter contains background information and observations we will refer to in the chapters that follow.

5.1 The simple variational autoencoder

In this section we define what we refer as the *simple VAE* in this chapter. It is actually the VAE model introduced as an example in section 2.2.3. We use this model to illustrate the difference between what was theoretically proposed and the common implementations.

The model is composed of a set of observed variables, which are identified as $\mathbf{x}$ and a set of unobserved latent variables, identified as $\mathbf{z}$. We assume $p(\mathbf{z})$ to be $N(0, I)$ and that $\mathbf{x}|z \sim N(\mu_x, \sigma_x)$. We also suppose that the dimension of $\mathbf{z}$ is d which is much lower than the dimension of $\mathbf{x}$, m.

Furthermore, in this model the parameters of the observed distribution $((\mu_x, \sigma_x))$ are continuous functions of the latent variable $\mathbf{z}$; $\theta = [\mu_x, \sigma_x] = f_x(z)$, to use a short notation, we identify $\mu_x(z)$ as the

function that takes z as input and return the parameters μ_x associated with this value and same for $\sigma_x(z)$ or simply $\theta(z) : \mathbb{R}^d \to \mathbb{R}^m \times \mathbb{R}^m_+$.

To ensure that this link function is as flexible as possible, a NN is used; $\theta(z)$ is a NN. It allows for a maximum amount of flexibility but in turn makes the posterior of the latent $p_\theta(\mathbf{z}|x)$ intractable and consequently the EM algorithm cannot be used. The proposed solution is to approximate $p_\theta(\mathbf{z}|x)$ with $q_\varphi(\mathbf{z}|x)$ a distribution of our choice.

For this simple variational autoencoder we will use a normal distribution: $\mathbf{z}|x \sim N(\mu_z, \sigma_z^2)$. Here again the parameters μ_z and σ_z are function of $\mathbf{x}$: $\varphi = [\mu_z, \sigma_z] = f_z(x)$ or $\varphi(x) : \mathbb{R}^m \to \mathbb{R}^d \times \mathbb{R}^d_+$. This function is once again a NN.

The simple VAE model is graphically represented as follows :

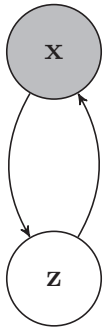
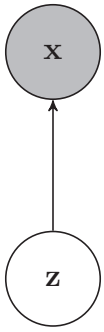


Figure 5.1: A graphical representation of the VAE architecture.

In Figure 5.1 the upward arrow represents the generative model (p) and the downward arrow represents the discriminative model (q) or inference network. Though not very useful now, it is practical for more complex model to separate both networks:



(a) The generative model assumes $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z})p_\theta(\mathbf{x}|z)$.

(b) The inference model. Given observations x we can infer the latent variable using $q_\varphi(\mathbf{z}|x)$.

5.1.1 Maximization of the ELBO

As previously mentioned in chapter 2, because it is impossible to compute the posterior distribution of the latent $p_\theta(\mathbf{z}|x)$ we cannot compute $\mathbf{E}_{p_\theta(\mathbf{z}|x)}[\ln p_\theta(\mathbf{z}, \mathbf{x})]$ and thus EM is not a viable solution here. The proposed solution is variational inference: we replace $p_\theta(\mathbf{z}|x)$ with an approximate distribution $q_\varphi(\mathbf{z}|x)$

and attempt to maximize the ELBO

$$\mathcal{L}(\varphi, \theta) = \mathbf{E}_{q_\varphi(\mathbf{z}|x)} \left[\ln p_\theta(\mathbf{z}) + \ln p_\theta(\mathbf{x}|z) - \ln q_\varphi(\mathbf{z}|x) \right].$$

The common strategy is to run a gradient-based optimizer on a Monte Carlo sample of the ELBO

$$\ln p_\theta(\mathbf{z}) + \ln p_\theta(\mathbf{x}|z) - \ln q_\varphi(\mathbf{z}|x) \quad z \sim q_\varphi(\mathbf{z}|x),$$

where we draw a new Monte Carlo sample at every steps of the optimization. To discuss further the current successful implementations, let us reorganize the terms in the ELBO

$$\begin{aligned} \mathcal{L}(\varphi, \theta) &= \mathbf{E}_{q_\varphi(\mathbf{z}|x)} \left[\ln p_\theta(\mathbf{z}) + \ln p_\theta(\mathbf{x}|z) - \ln q_\varphi(\mathbf{z}|x) \right] \\ &= \mathbf{E}_{q_\varphi(\mathbf{z}|x)} \left[\ln p_\theta(\mathbf{x}|z) - (\ln q_\varphi(\mathbf{z}|x) - \ln p_\theta(\mathbf{z})) \right] \\ &= \mathbf{E}_{q_\varphi(\mathbf{z}|x)} \left[\ln p_\theta(\mathbf{x}|z) \right] - \mathbf{E}_{q_\varphi(\mathbf{z}|x)} \left[\ln q_\varphi(\mathbf{z}|x) - \ln p_\theta(\mathbf{z}) \right] \\ &= \underbrace{\mathbf{E}_{q_\varphi(\mathbf{z}|x)} \left[\ln p_\theta(\mathbf{x}|z) \right]}_{\text{Reconstruction error}} - \underbrace{KL\left(q_\varphi(\mathbf{z}|x) | p_\theta(\mathbf{z}) \right)}_{\text{Regularization term}}. \end{aligned} \tag{5.1}$$

It is common to perceive the ELBO with respect to those two terms. We can see this as a regularized optimization problem where we want to maximize the first term, the observed-data likelihood, and where the second term works as penalization that discourages $q_\varphi(\mathbf{z}|x)$ from drifting far away from a $N(0, I)$ which represents the effect of the prior with a Bayesian flavour.

5.1.2 Practical uses

Dimensionality reduction and representation learning

A VAE is an unsupervised learning model like k-means clustering, GMM or Principal Component Analysis (PCA). Just like these other techniques, VAE can be used for dimensionality reduction. If the code z is a of much lower dimension, given the fitted encoding function φ and decoding function θ we can easily encode large observations x into the parameters of their lower-dimension representation z and also decode this representation to get the parameters of the reconstructed observation distribution. In contrast to k-means or PCA, VAE offers a probabilistic dimensionality reduction rather than a deterministic one which is similar to what GMM or pPCA offers in that aspect.

Dimensionality reduction is very useful for storage purposes or message transmission. A VAE can be directly used for lossy compression where z is the compressed representation x. VAEs can also be used as building blocks in more complex compression schemes, for instance Townsend et al. [145] constructed a lossless compression algorithm (BB-ANS) using VAEs. Besides, it is also quite common to apply supervised learning techniques to the latent representation itself, as discussed later.

Manifold learning is often a synonym of non-linear dimensionality reduction and in the context of machine learning is often viewed as a feature extraction step. Furthermore, the lower-dimension representation itself can be analysed sometimes [86, 114, 121, 83] to visualize the compression process. Since compression functions are continuous, it also allows us to better understand distances in the high-dimensional space $\mathcal{X}$.

Generator

A VAE is a generative model [83]. Indeed, since a prior distribution is assumed for the latent variable, $\mathbf{z} \sim N(0, I)$ we have a fully defined joint distribution $p_\theta(\mathbf{x}, \mathbf{z})$ and it is possible to generate new *observations* using ancestral sampling. Ancestral sampling [15] is the generative procedure for graphical models. The graphical representation of VAEs, a Bayesian network [88], represents a set of factorization and conditional independence assumptions which induces a natural sequence of events. In the simple VAE case the assumed factorization is :$p(x, z) = p(z)p(x|z)$. This leads to the following ancestral sampling procedure:

Algorithm 6: Ancestral Sampling with the simple VAE
INPUT: n desired size of the generated sample 1) Sample z from $N(0, I)$. 2) Process z through the NNs θ to get $\mu_x(z)$ and $\sigma_x(z)$. 3) Sample x from $N(\mu_x(z), \sigma_x(z))$. 4) Return x OUTPUT: a sample x of size n

Doing so allows us to generate new data points x according to its assumed distribution.

Semi-Supervised learning

VAEs can also be solutions to semi-supervised problems as was proposed by Kingma [85, 83] shortly after the release of the introductory paper [86]. Various semi-supervised VAE models have been proposed [85, 120, 107, 155]; and Rastgoufard [121] offers a thorough analysis of these models applied to various semi-supervised tasks. Semi-supervision aims at learning the *supervised relation* between the observations x and the labels y given a data set where multiple observations are not labelled. In other words, given a standard labelled data set can we improve the classifier by incorporating additional unlabelled observations ? It is quite an important problem since expert labelling is far more costly than the process of collecting raw data [121].

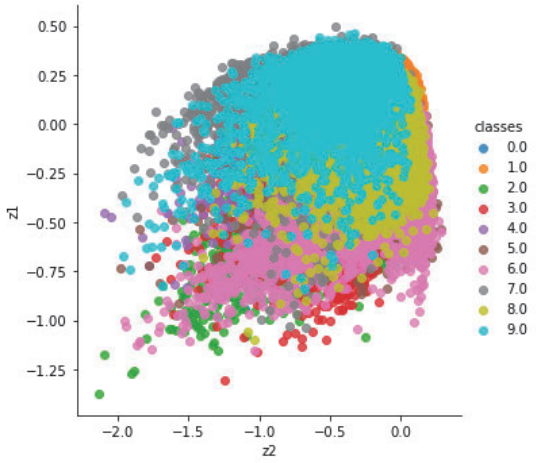
Observations from different classes are likely to cluster in different regions of the latent representation. In other words, observations x attached to different labels y will probably have different latent representation z. We can use the labelled observation to find an appropriate prediction function h that takes z as input and return a predicted label $\hat{y}$. The strategy behind semi-supervised learning is to leverage the large amount of unlabelled point to improve the encoding function $q_\varphi(\mathbf{z}|x)$ thus improving the classification mechanism.

In fact, the encoder q can be perceived as the feature-extraction step [153, 1, 122]; $\mathbf{z} \sim q(\mathbf{z}|x)$ is the vector of features extracted from the image x and it is easier to classify observations using those features than when using the original observations x. The classifier h is trained on the features z using labelled data set $S_l = \{(x_i, y_i)|i \in 1..n_l\}$ but we can use the additional unlabelled data $S_u = \{x_j|j \in 1...n_u\}$ to improve the feature extractor q.

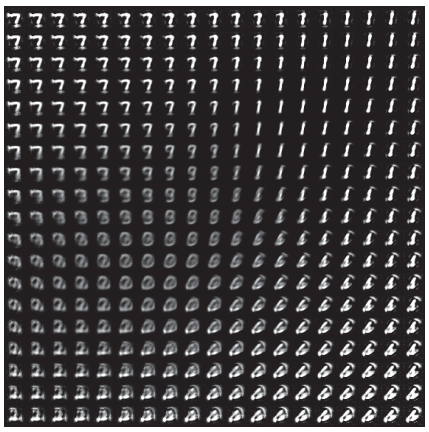
5.2 Visualization of the simple VAE

In this section we demonstrate problems when implementing the simple VAE directly. The VAE and it's associated generative procedures are implemented in Python and we will use the well-known MNIST data set [95] to visualize some of these problems. In hand-written document analysis, the MNIST data set introduced by LeCun & al. [95] quickly became a benchmark for hand-written digits recognition and is now a rite of passage for computer vision algorithms. It contains more than 60,000 images in shades of grey of hand-written digits of size 28 by 28 pixels.

The following images and plots are visual supports for our arguments about the simple VAE problems. We produce plots and images that illustrates how VAE performs in tasks mentioned before; compression and generation. Semi-supervised applications are left out for now, but VAEs limitations in semi-supervision problems has been extensively studied in Rastgoufard's thesis [121]. For compression we look at the latent space and its associated reconstruction $\mu_x(z)$, possible if $d = 2$, as well as an example of reconstruction to observe the *loss* incurred by the compression and decompression process.



(a) Observations x projected onto its latent representation using $z \sim N(\mu_z(x), \sigma_z(x))$.



(b) Decoded latent space using $\mu_x(z)$.

Figure 5.3: Latent space visualization of a simple VAE with latent space of dimension $d = 2$

Figure 5.4: Images x on the top row and its reconstruction $\mu_x(q_\varphi(x))$ on the bottom row produce from a simple VAE with latent space of dimension $d = 2$

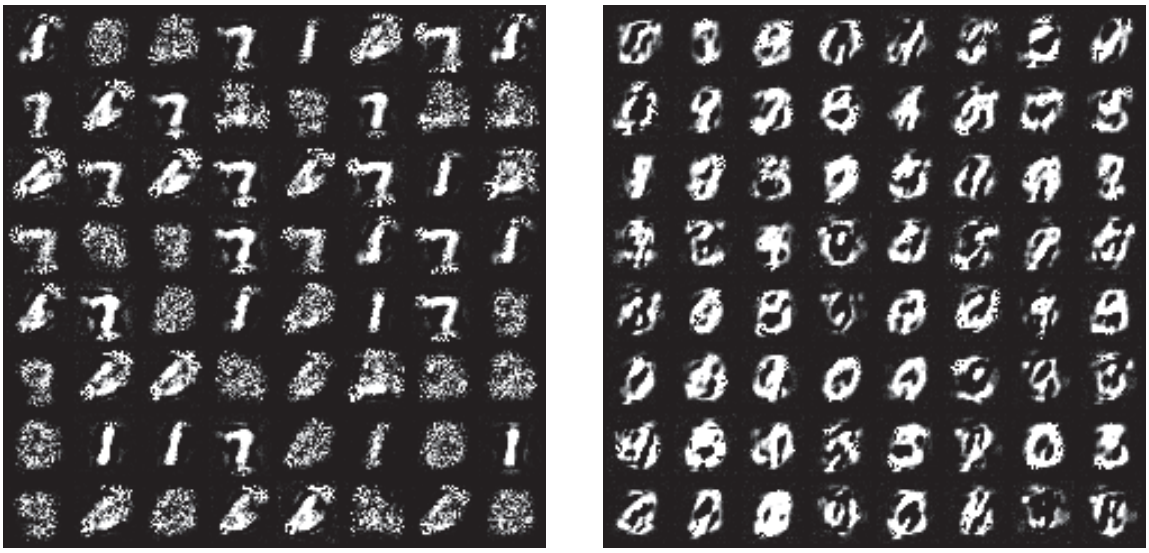
Figure 5.5: Images x on the top row and its reconstruction $\mu_x(q_\varphi(x))$ on the bottom row produce from a simple VAE with latent space of dimension $d = 20$

Figures 5.4 and 5.5 contain images and their associated reconstruction. We see multiple imperfections, blurry images and sometimes the reconstructed digit is a completely different digit. Of course, we do expect to lose some details when reducing the dimension from 784 to 20, but we know NNs allows for function complex enough and we hope to achieve better results than what is obtainable with PCA which compresses the data with linear combination.

Figure 5.6: Examples of reconstruction produced by PCA included in Bishop's book [15]. The image to the left is a real image and other images are reconstruction with latent space of size $d = 1$, $d = 10$, $d = 50$ and $d = 250$ respectively.

PCA uses simple linear combination for compression and decompression. However, it achieves reconstruction of similar quality with a latent space of size $d = 10$ (third image of Figure 5.6) than the simple VAE with a latent space of size $d = 20$ who relies on NN as for compression and decompression. This is a disappointment.

For generation, we use ancestral sampling to produce a sample of 64 images:

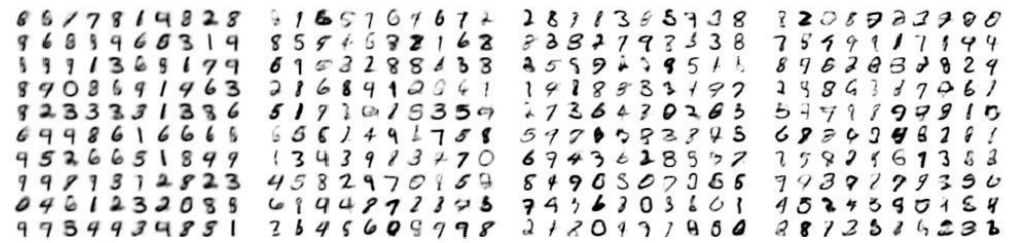


(a) $d = 2$ (b) $d = 20$

Figure 5.7: Sample obtained from the ancestral sampling described in the previous section.

Now we observe another major problem; the images generated are blurry, contains multiple imperfection and lack diversity. A human eye would judge harshly those images; they do not look like realistic hand-written digits. As it stands, with an exact implementation of the simple VAE, the compression

and reconstruction abilities of such models are equivalent to PCA and the generated images are not impressive. None of these problems are mentioned in the papers that originally presented this model [86, 73]. For instance, here are the samples available in Kingma's thesis [83] :



(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Figure 5.8: Snapshot of the result section of Kingma's thesis.

It is unlikely that neither of the first authors did not encounter any of these problems. By avoiding this discussion, it falls on the users to figure out how to implement the needed modifications and this prevents the model to be used reliably on real-data problems as is.

5.3 Algorithmic solutions

As we observed in the previous section, a direct implementation of the simple VAE proposed in the literature suffers from problems for both reconstruction and image generation. The figures of section 5.2 reveal some problems of the simple VAE. However figures much more flattering were published in articles discussed above. To produce those images, some important modifications were done by researchers *under the hood* of the proposed VAE of section 2.2.3, we will discuss those modifications in their respective subsection.

In this section we explore successful implementations of VAEs and we highlight the differences between the simple VAE proposed and the common implementations. We discuss these differences and their impact on the resulting model; how they fix some problem but drastically steer the model away from its original proposed form. We named this section *algorithmic solution* since the explored modification are algorithmic rather than theoretical. Our main objective in this chapter is to raise awareness and motivate further research in this area.

5.3.1 Tradeoff between reconstruction and regularization

Remember that

$$\begin{aligned}\mathcal{L}(\varphi,\theta) &= \mathbf{E}_{q_\varphi(z|x)}\left[\ln p_\theta(z) + \ln p_\theta(x|z) - \ln q_\varphi(z|x)\right] \\ &= \underbrace{\mathbf{E}_{q_\varphi(z|x)}\left[\ln p_\theta(x|z)\right]}_{\text{Reconstruction error}} - \underbrace{KL\left(q_\varphi(z|x)|p_\theta(z)\right)}_{\text{Regularization term}}. \qquad (5.2)\end{aligned}$$

(5.3)

Figure 5.9: Images x on the top row and its reconstruction $\mu_x(q_\varphi(x))$ on the bottom row produced with a β-VAE with latent space of dimension $d = 20$

In multiple implementations we observed a wide range of modifications to the objective function where both the reconstruction error and the regularization term are considered separately. To begin we will address the balance, or lack thereof, between the two components of the ELBO.

When perceiving the ELBO as a regularized optimization problem as defined in section 5.1.1, the need for an hyper-parameter controlling the strength of the regularization might seem beneficial. It is now common to add a hyper-parameter, say β in the objective function to allow us to control the balance between the reconstruction error and the regularization

$$\mathbf{E}_{q_\varphi(z|x)}\left[\ln p_\theta(x|z)\right] - \beta KL\left(q_\varphi(z|x)|p_\theta(z)\right). \tag{5.4}$$

Models optimized with the objective function of Equation 5.4 are known as β-VAEs [63, 24] and they were developed to improve the disentanglement of the latent representation. In fact, their ability to form a disentangled representation has been well studied [63, 24]. However, there is little discussion on the effect of this hyper-parameter on the generative abilities of β-VAEs and how to select the hyper-parameter β appropriately.

The authors indicate that a large β is *putting a stronger constraint on the latent bottleneck than in the original VAE formulation which... should encourage the model to learn the most efficient representation of the data.* They also claim that the regularizing term in the objective function encourages conditional independence in q_φ. However this is done to the detriment of the reconstruction term and to the detriment of variability in generated samples.

Simply put, Equation 5.4 directly implies that small β leads to a more accurate reconstruction and large β to more regularization. In other words, small β enables the algorithm to compress the data to a lower-dimensional space and reconstruct an almost perfect image:

In Figure 5.9 we observe much better reconstruction that previously in Figure 5.5.

5.3.2 Reconstruction term

Secondly, let us discuss the implementation of the reconstruction term. If we assume that $\mathbf{x}|z \sim N$ then

$$\begin{aligned}\ln p_\theta(x|z) &= \ln\left(\frac{1}{\sqrt{2\pi\sigma(z)^2}}\exp\left(\frac{-(x-\mu(z))^2}{2\sigma(z)^2}\right)\right)\\ &= -\frac{1}{2}\ln\left(2\pi\sigma(z)^2\right) - \frac{(x-\mu(z))^2}{2\sigma(z)^2}.\end{aligned} \tag{5.5}$$

Common implementations do not maximize the reconstruction term of Equation 5.5. Instead the NN θ returns an output of the same size as x and minimize the mean squared error (MSE) between x and

the reconstructed $\bar{x} = \mu(z)$. The motivation is that minimizing the MSE is equivalent to maximizing the log-likelihood for a normal distribution with a fixed $\sigma_x = 1$ (I)

$$-\frac{1}{2}\ln(2\pi) - \frac{(x-\mu(z))^2}{2} \propto -(x-\mu(z))^2.$$

Based on empirical result fixing $\sigma(z)$ produces better reconstructed images:

Figure 5.10: Images x on the top row and its reconstruction $\mu_x(q_\varphi(x))$ on the bottom row produce from a simple VAE with latent space of dimension $d = 2$ and $\sigma = 1$

Figure 5.11: Images x on the top row and its reconstruction $\mu_x(q_\varphi(x))$ on the bottom row produce from a simple VAE with latent space of dimension $d = 20$ and $\sigma = 1$

In Figures5.10 and 5.11 we have noticeably better reconstruction than in Figures 5.4 and 5.5.

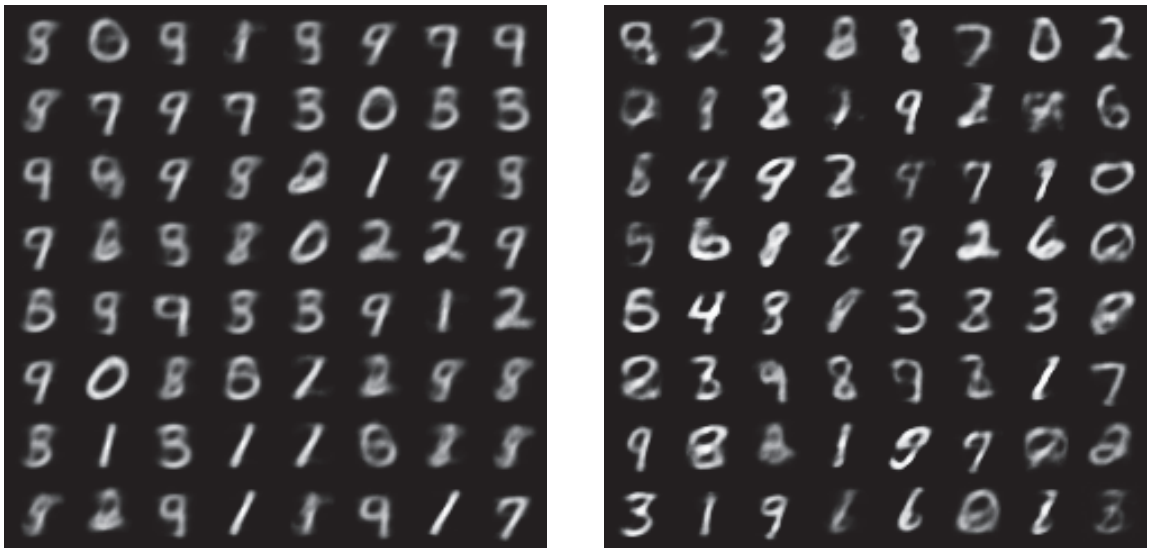
5.3.3 Modification to the ancestral sampling procedure

Finally, let us introduce a modification done to the data generation technique. We previously discussed how VAEs were presented as generative models and that the graphical representation suggested the Ancestral Sampling technique detailed in Algorithm 6.

However, all the implementations found online, including Kingma's implementation [85], do not rely on ancestral sampling. Actually, none of the implementation we found sample images from $p(x|z)$, instead x is deterministic given z, which is why we coined this technique *deterministic sampling*.

Algorithm 7: Deterministic Sampling
INPUT: n desired size of the generated sample 1) Sample z from $N(0, I)$. 2) Process z through the NNs θ to get $\mu_x(z)$ and $\sigma_x(z)$. 3) Return $\mu_x(z)$ OUTPUT: a sample of means (μ_x) of size n

The only difference between the samples of Figures 5.7 and Figures 5.12 is the sampling technique. In other words, we have trained a simple VAE, as introduced in section 5.1, but instead of generating x from $p_\theta(\mathbf{x}|z)$, we simply returned $\mu_x(z)$. The difference in the quality is noticeable on eyesight.



(a) $d = 2$ (b) $d = 20$

Figure 5.12: Sample obtained from $\mu_x(z)$ where $z \sim N(0, I)$.

Figure 5.13: Images x on the top row and its reconstruction $\mu_x(q_\varphi(x))$ on the bottom row produced from a simple VAE with latent space of dimension $d = 2$ and $\sigma = 0.0001$

Figure 5.14: Images x on the top row and its reconstruction $\mu_x(q_\varphi(x))$ on the bottom row produced from a simple VAE with latent space of dimension $d = 20$ and $\sigma = 0.0001$

5.3.4 Effect on the model optimized

The three algorithmic solutions discussed above have one thing in common; they all directly affect how the total observed variance is distributed in the resulting model.

For β-VAE, the β parameter influences the variance of $q_\varphi(\mathbf{z}|x)$. Large β pushed $q_\varphi(\mathbf{z}|x)$ towards a $N(0, I)$ distribution while small β allows $q_\varphi(\mathbf{z}|x)$ having a much large variance with correlated dimensions.

Similarly, when fixing $\sigma_x = I$, the distribution of the total variability is now fixed by the user. Fixing σ_x constrain the variance of $p_\theta(\mathbf{x}|z)$. Like β can be adjusted to distribute the variability in different way, σ_x can be fixed to different values. For instance, if we wanted to force the latent variable to take on larger portions of the variability we could fix σ_x to a small value, this was also suggested by Lucas et al. [105].

Figure 5.14 shows impressive reconstruction. These images are, on eyesight, as good as those of Figure 5.9 but using totally different approaches. Earlier with selected a small β thus relaxing the constraint applied by the regularization term and here we have fixed a small σ_x. Those two techniques allow for most of the variability to be explained by the latent space. The fact that adjusting σ_x leads to similar fit than adjusting β was mentioned by Lucas [105] but it was not mentioned that this is due to both having similar effect on how the total variability is distributed across both components, latent and observed, of the VAE.

Finally, the last modification discussed constrains the variance of $p_\theta(\mathbf{x}|z)$ when generating. For a fixed z then $x = \mu_x(z)$ which is equivalent to fixing $\sigma_x = 0$; this turns the generating distribution from a Normal to a Dirac Delta distribution.

Another way to perceive this constraint is that it ensures a correlation between the pixels. When using $\mu_x(z)$ every pixel has the exact same distribution value of 0.5. This is because the mean of a normal distribution is also its median; $F(\mu) = 0.5$ where F is the distribution function of any normal distribution. Consequently, pixels are perfectly correlated in their distribution value. We can produce images that look just as good by sampling $z \sim N(0, I)$ and then outputting $\mu_x(z) + \alpha \times \sigma_x(z)$ for any α as seen in Figure 5.15.

(a) $\mu_x(z) - 0.1 \times \sigma_x(z)$ (b) $\mu_x(z) + 0.1 \times \sigma_x(z)$

Figure 5.15: Samples obtained from a simple VAE with $z \sim N(0, I)$.

To summarize, these three algorithmic solutions improve either the compression/decompression abilities or generative abilities of the simple VAE by putting constraints on the variance either by modifying the objective function while training or by modifying the sampling procedure when generating new observations. In the next section we argue that these modifications create new problems and we demonstrate that the model resulting from those modifications does not respect the theory any longer which, in turn, makes these new problems hard to solve.

5.4 Issues with algorithmic solution

5.4.1 Application issues

The common algorithmic solutions discussed above solve some issues of the simple VAE as illustrated in section 5.3, however, new problems also appear.

First, selecting the β parameter is a complicated task where the user has to define how important is the reconstruction relatively to the regularization of q_φ. To this day there is no automated way to select the right value for β. Similarly, if we desire to fix σ_x the value of this fixed variance has to be established heuristically.

Second, the improvement in reconstruction observed when fixing a small β or a small σ_x comes to the detriment of the generative abilities of VAE. In fact, small β or small fixed σ_x leads to a q_φ with high variance as explained earlier. This is problematic from a generative perspective. Remember that we optimize a Monte Carlo sample of the ELBO, thus we train $p_\theta(\mathbf{x}|z)$ using zs sampled from $q_\varphi(\mathbf{z}|x)$. In other words, the NN function θ is trained with zs generated from $q_\varphi(\mathbf{z}|x)$. Consequently if $q_\varphi(\mathbf{z})$ and $p_\theta(\mathbf{z})$ have drastically different supports then we do not know how does the NN θ will react when receiving inputs z sampled from $p_\theta(\mathbf{z})$.

Another generative problem arises with large β, this leads to a lack of variability in generated images. The lack of variability in the generated data happens when $q_\varphi(\mathbf{z}|x)$ resembles too much the prior $p_\theta(\mathbf{z})$, instead of getting close to the intractable posterior $p_\theta(\mathbf{z}|x)$, and this problem has been recently coined *posterior collapse*. As a matter of fact, if $q_\varphi(\mathbf{z}|x) \approx p_\theta(\mathbf{z})$ then $q_\varphi(\mathbf{z}|x)$ is *independent* of x; $q_\varphi(\mathbf{z}|x)$ does not vary as x varies. This is not intended; we want the latent representation of x obtained through $q_\varphi(\mathbf{z}|x)$ to contain information about x and thus to be different for different xs. This leads to a latent space that does not contain information about the observed space and this leads to an homogenized reconstruction.

Figure 5.16 provides a visualization of the problems caused when β is either too large or too small. In Figure 5.16a we see the high variance latent space comparatively to the much more constraint counterpart of Figure 5.16c. The effect of posterior collapse, with too large β, can also be observed in Figure 5.16d where $\mu_x(z)$ is constant in z and the resulting image is the average of all digits.

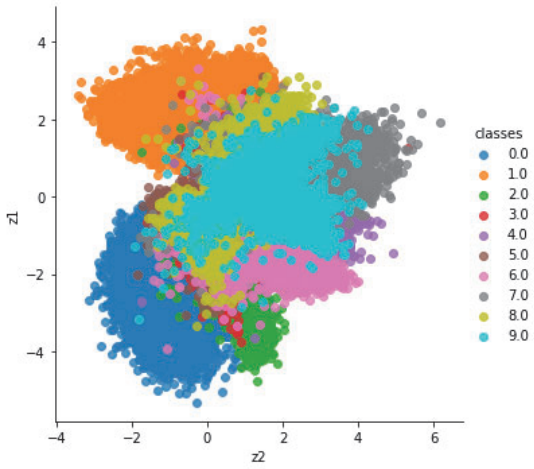
These problems are easier to observe when using VAEs on a single digit data set:

We see bigger variability in the images spanned by the latent space in Figure 5.17b but also more imperfections. This happens when θ has to process zs unobserved in the training process. This contrast with Figure 5.17d where all images are relatively good but they all look alike. This is a symptom of posterior collapse.

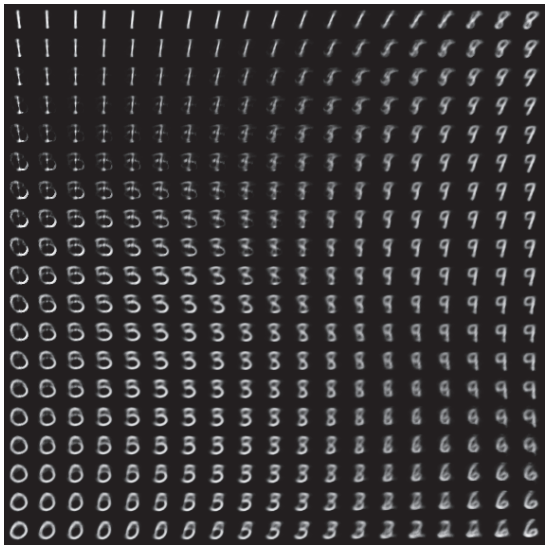
Similarly we have high variance for $q_\varphi(\mathbf{z}|x)$ when fixing σ_x to a small value such as observed in Figure 5.18.

5.4.2 Theoretical issues

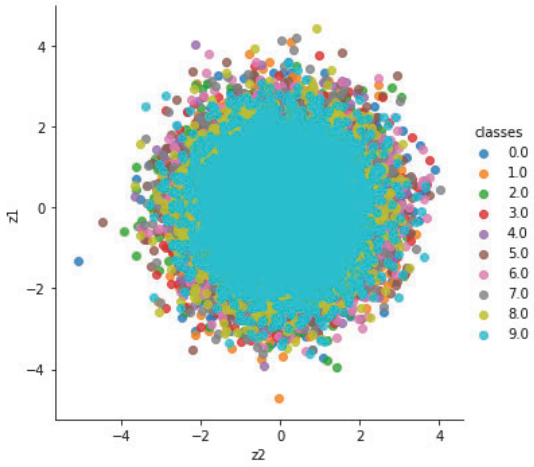
Additionally, we want to raise awareness towards theoretical issues with the algorithmic solutions detailed in the previous section.



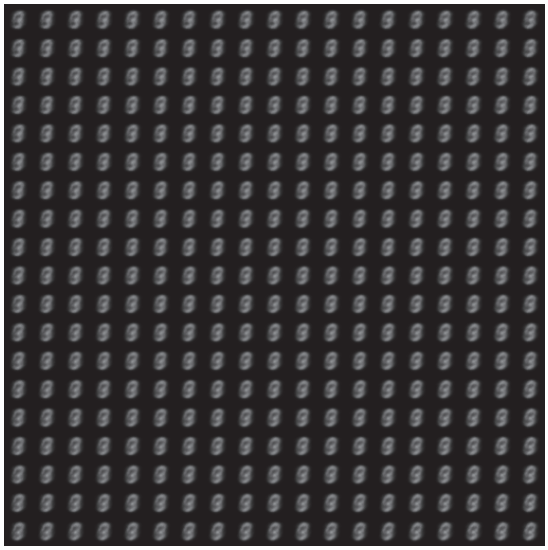
(a) Observations x projected onto its latent representation using $z \sim N(\mu_z(x), \sigma_z(x))$ with small β



(b) Decoded latent space using $\mu_x(z)$ with small β

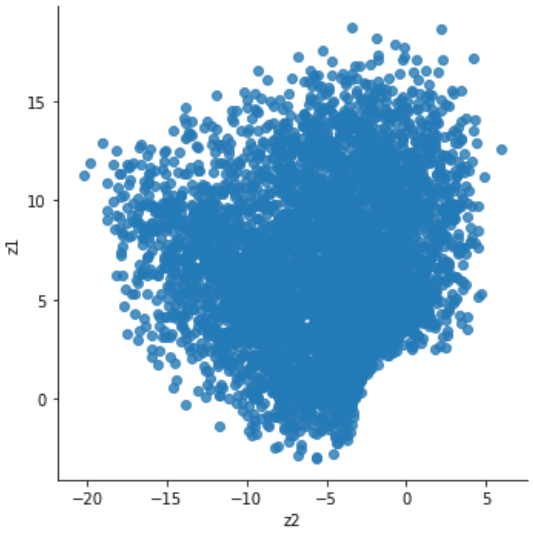


(c) Observations x projected onto its latent representation using $z \sim N(\mu_z(x), \sigma_z(x))$ with large β

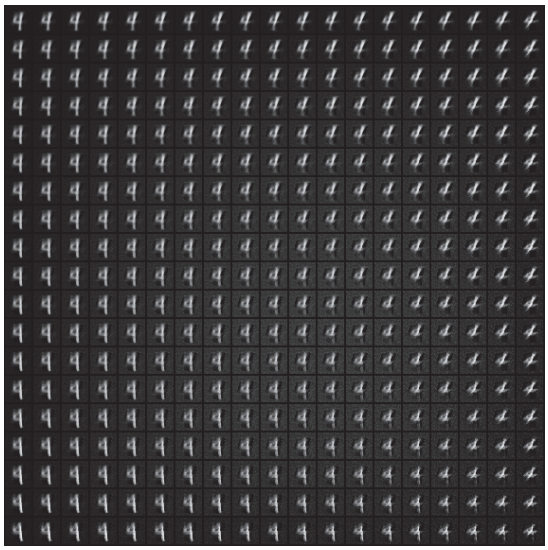


(d) Decoded latent space using $\mu_x(z)$ with large β.

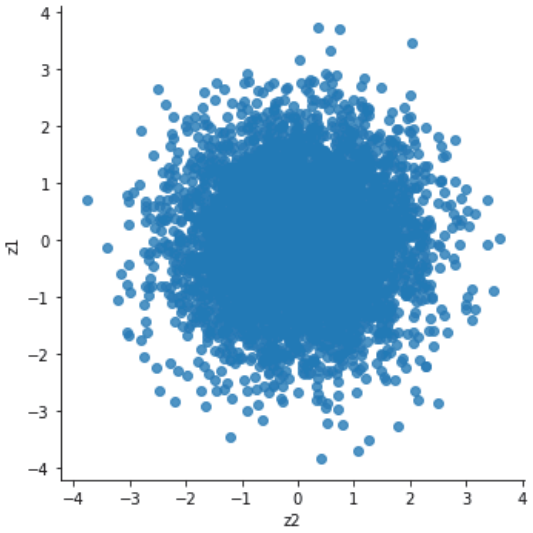
Figure 5.16: Visualization of the latent representation.



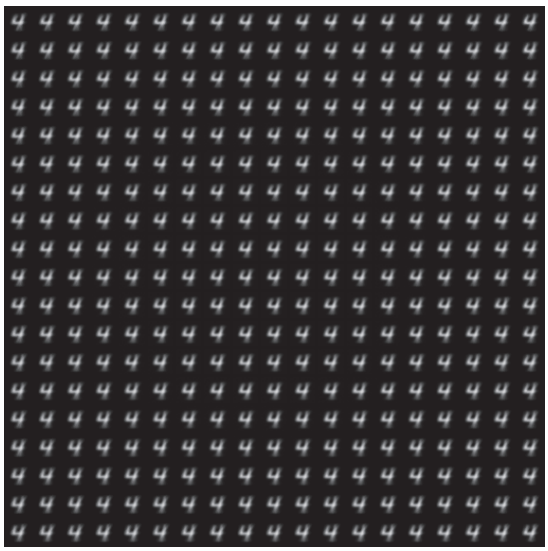
(a) Observations x projected onto its latent representation using $z \sim N(\mu_z(x), \sigma_z(x))$ with small β.



(b) Decoded latent space using $\mu_x(z)$ small β.

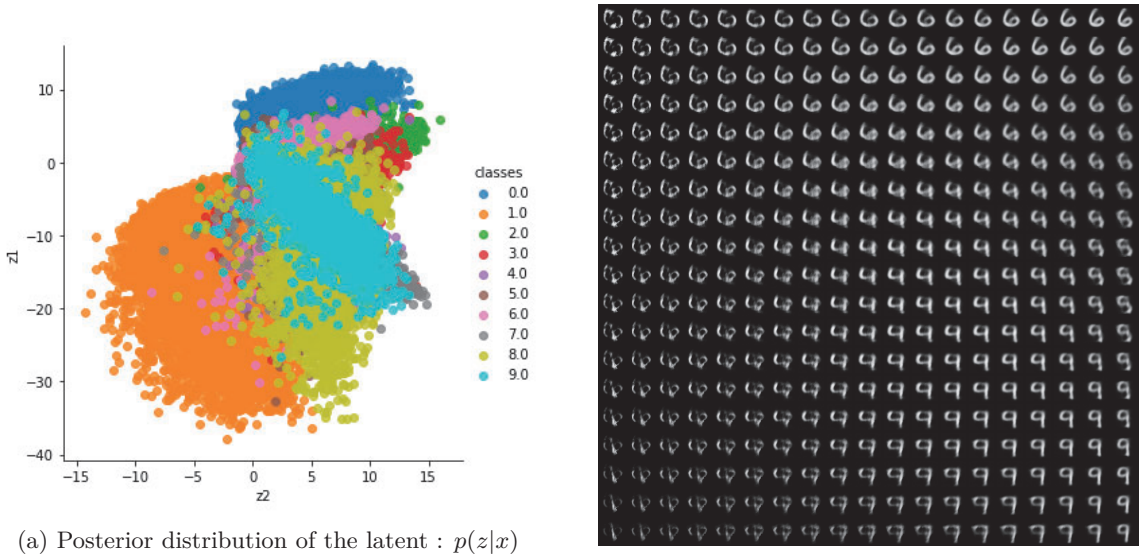


(c) Observations x projected onto its latent representation using $z \sim N(\mu_z(x), \sigma_z(x))$ with large β



(d) Decoded latent space using $\mu_x(z)$ with large β

Figure 5.17: Visualization of the latent representation.



(a) Posterior distribution of the latent : $p(z|x)$

(b) Decoded latent space using $\mu_x(z)$.

Figure 5.18: VAE with latent space of dimension $d = 2$ and $\sigma = 0.0001$

To begin, we quickly address the choice of observed data distribution. Though the VAE model is very flexible in the form $p_\theta(\mathbf{x}|z)$ can take, lots of implementations use the Bernoulli distribution [86, 137, 91, 40]. This is a problem since the Bernoulli distribution supports binary variable and thus it should not be used to model pixels continuously distributed in $(0, 1)$. Similarly, we also encounter a support problem with normal distribution. Pixels are continuously distributed in $(0, 1)$ but the Normal support is infinite. However these are more data compatibility problems rather than theoretical issues with the proposed algorithmic solutions.

When strictly considering the solutions discussed, the biggest problem is the violation of certain theoretical properties and guarantees of the simple VAE. For the β-VAE, by selecting a $\beta < 1$ the resulting objective function is no longer a lower bound of the marginal log-likelihood $\log p(x)$ and thus we are losing an important theoretical guarantee of the model.

We can take the β-VAE concept to its limit and fix $\beta = 0$, this produces the best reconstruction possible but also eliminates one of the novelties of VAEs; the distribution of the latent variable z. In fact, when $\beta = 0$ the parameters of $q_\varphi(\mathbf{z}|x)$ are not estimated anymore. The resulting model is much closer to an AE fitted by maximizing a likelihood function.

A similar problem comes with combining the use of the MSE as the reconstruction error and the deterministic sampling procedure. If we optimize $\mu_x(z)$ by minimizing the MSE, thus fixing σ_x when training and the data generated is $\mu_x(z)$ itself, thus fixing σ_x when generating then we got rid of the probabilistic components of x. Indeed, the θ parameters can be reduced to μ_x and the variance of $p_\theta(\mathbf{x}|z)$ is not considered at any point in time during training nor generation. In other words, the resulting model is totally deterministic in x given z.

Combining these modifications altogether, and taking the β-VAE to its extreme case we now have

the following objective function:

$$(x - \mu_x(z))^2 \quad \text{where } z \sim q_\varphi(\mathbf{z}|x) \tag{5.6}$$

When maximizing the resulting objective function of Equation 5.6, only $\mu_x(z)$ is trained. The model is now an AE with a NN decoder optimized by minimizing the mean-squared reconstruction error and an untrained probabilistic encoder.

5.5 Future work

As explained in section 5.3.4 the algorithmic solutions have one thing in common: they all influence how the total variance is distributed between the latent variables and observed variables. Hence we believe the cause of the problems observed in section 5.2 with the simple VAE is the lack of identifiability between the variability attributed to both the observed and the latent component. In short, when fitting latent variable models the total variability within the observed data x is split between the variance of the latent variable and the variance of the observed variable and there exist infinitely many ways to split the total variance in two. To solve this identifiability problem it is common to fix the variance of one of the components or to decide how to distribute the variability between the two components when establishing the optimization procedure.

For instance, in PCA the latent representation is designed to take on as much variability from the observed data space as possible. In PCA, the variance of the latent representation z of size d is the average of the d largest eigenvalue of the covariance matrix of the observed data.

We want to propose a new theoretical formulation, along with concordant implementation that solves this variance identifiability problem. We believe having implementations concordant to the theory is beneficial as it helps to generalize the model to new applications and it will allow us to rely on the theory if problems come up, which is not the case with the modification now that the resulting model has strayed away from the theoretical formulation. We also believe that fixing this variance identifiability problem would be beneficial as it would better grasp the natural variability in a wide range of data sets which is ignored in a deterministic AE.

Our goal is to allow for the total variance to be expressed differently than it is right now. We want to take a closer look at probabilistic PCA (pPCA) [144, 143]. In this model, the variance of the latent representation z of size d is the average of the d largest eigenvalue again and the variance of $p_\theta(x|z)$ is the average of the leftover eigenvalues. Compared to PCA where the maximum variance projection is enforced by the model, the solution in pPCA happens naturally without specific constraints. This is a strong result we hope to use in our future work on VAE in order to balance naturally both components' variance. Based on the results of PCA and pPCA where a maximum amount of the variability is attributed to the latent variable and on the results we observed with β-VAE it seems like for the simple VAE to produce better reconstruction and generations it needs to shift some of the observed distribution variance to the latent representation variance. Similarly, we could study GMMs, or any other model-based clustering. Studying the similarities and differences between VAEs and GMMs could provide us with interesting insight regarding the problems that VAEs suffer from.

Additionally, once this balance is fixed, we believe it is important that the observed distribution variance can be expressed in a more complete manner; we want to drop the conditional independence assumption.

$$\mathrm{Var}(\sum_{j=1}^{m} \mathbf{x}_j) = \sum_{j=1}^{m} \mathrm{Var}(\mathbf{x}_j) + \sum_{j \neq i}^{m} \mathrm{Cov}(\mathbf{x}_j, \mathbf{x}_i) \tag{5.7}$$

In the simple VAE, the normal distribution that models observations has a diagonal covariance matrix, which bottlenecks all of the observed distribution variance on the diagonal as suggested by the simple decomposition of the total variance of Equation 5.7.

In other words, even if the total variance was distributed optimally between $\mathbf{z}$ and $\mathbf{x}|z$ we would also need to let some of the covariance term of Equations 5.7 to be non-zero otherwise it will result in high individual variable variance. This should also better model real data such as images where pixels in a neighbourhood are highly correlated. In order to optimally fit a covariance matrix we are currently exploring ideas of spatial statistics.

5.6 Related literature

We faced those problems within the first few years of VAE's existence in early 2016 and slowly started working on this chapter. Back then, none of the literature available mentioned those problems neither how the small coding tricks established earlier were actually drastically changing the model.

Based on our research, the *posterior collapse* problem is the problem addressed the most in the literature and it is now fully recognized as a problem and received a lot of attention in the last few years. Though it was not in an attempt to solve the issue the paper presenting the β-VAE formulation [63, 24] was among the first to discuss the effect of the regularization term of the ELBO and it's potential effect of the variability in the images it produces. He et al. [62] recently provided an insightful investigation of posterior collapse; they suggest that the cause of posterior collapse is the inference of the approximate distribution lagging behind the true posterior at the early stages of training.

Alemi et al. [3] directly discussed the posterior collapsed problem with an information theory approach. The problem is indeed that z does not contain enough *information* about x and they propose to optimize VAEs in a way that maximizes the mutual information between the observed variables and the latent variables. Not only did they address the problem but they also encouraged research in that regard.

After a publication from Dai et al. [37] discussing the relationship between PCA and VAEs, Lucas et al. [105] made connections with the pPCA model. They demonstrate that the regularization term is only partially responsible for posterior collapse but mostly that the variance parameter of the decoding distribution was playing a huge role. This confirms what we suspect. The authors make a thorough analysis of the effect of the variance term and suggest that the optimization procedure naturally favours too much observed-data variance and suggest way to reduce it to solve the posterior collapse problem. Overall this paper is a great contribution towards solving some of the VAE issues.

Lucas et al. [105] also show that, for a linear VAE, the ELBO has the same global maximum as the log likelihood and thus the solution has scaled principal components as the columns of the decoder network.

They also show that using the ELBO objective does not introduce new local maxima. Finally, after establishing a metric for posterior collapse they demonstrate how fixing a small σ_x makes the posterior collapse problem completely disappear. Additionally, we have shown in this chapter that restraining the variance in $p(x|z)$ drastically improves the reconstruction abilities of VAEs. Now, many solutions and formulations have been proposed [4, 151, 128]recently to discuss the variance problems and we are excited to see such keen interest towards this problem.

Although we have only noticed few articles discussing the issue with the current generative problem, Dorta et al. [42, 41] came up with a similar observation that we did; $\mu_x(z)$ is commonly used to generate images because of the poor performances of ancestral sampling which is caused by the lack of correlation between pixels. The solution they proposed is a fully parametrizes covariance matrix for $p(x|z)$.

5.7 Conclusion

The VAE model as defined in the literature [86, 83] is built upon a rigorous theory and the described model is both innovative and a big contribution to the fields of machine learning and statistics alike. It extends latent variable models to allow for more flexible functions between the latent and the observation space and has empirically performed well on some real-data problems.

However it seems there are big difference between the theory and the popular implementations. The current implementations fix some of the problems encountered when using VAEs but they do so by taking out the components that made VAEs special. In this chapter, we demonstrated how most of the simple fixes we found online are progressively transforming a VAE into an AE. We demonstrated that these fixes also come with new problems. Finally, we provided a taste of the solutions we are currently working on.