Testing Improvements for Vector Quantized Variational Autoencoders with Novel Data

Shivanshu Gupta **Kolby Nottingham** **Preethi Seshadri**

Abstract

We explore using the Vector Quantized Variational Autoencoder (VQ-VAE) to generate discrete representations for the Kaokore dataset, which contains images of facial expressions from traditional Japanese illustrations (`https://github.com/rois-codh/kaokore`). The framework VQ-VAE is built on, Variational Autoencoders (VAE), learn continuous latent representations. While continuous representations are flexible, many real world attributes are better defined discretely, and some current state-of-the art model architectures, like transformers, only work with discrete data. Additionally, VAEs have been shown to exhibit posterior collapse, which means that latent codes are ignored. In this project, we experiment with VQ-VAEs on a novel dataset and design experiments to test the advantages and disadvantages of multiple VQ-VAE variations. Our results indicate that while the original VQ-VAE algorithm learns faster than some of its successors, it does not achieve the same level of performance.

1 Introduction

Generative machine learning models learn a distribution $p(x, y)$ for data instances x and labels y. Compared to discriminative machine learning models that learn $p(y|x)$, generative models can be more difficult to learn but come with added benefits. One use case for generative machine learning models is mapping data instances to a latent space. Once mapped to a latent space, latent variables can be used as a compressed representation of a data instances or to sample and generate new data instances.

One generative machine learning model that learns latent variable representations of data is the variational autoencoder (VAE). VAEs learn a latent variable representation of each datapoint with an encoder module. Simultanously, a decoder module learns to reconstruct the original image from the latent variables.

Traditional VAEs learn a latent space that is associated with a probablility distribution of continuous variables. Vector quantized variational autoencoders (VQ-VAE) learn discrete latent variables instead. Discrete variables are cometimes advantageous because real world data can often be summarized by categorical features.

In this work, we compare two variations of VQ-VAEs. The second variation (VQ-VAE2) was released as a follow up to VQ-VAEs and introduces a hierarchical structure for the model. We go into further detail comparing the algorithms in section 2. We implement each of these methods and run experiemnts with each on the Kaokore dataset. To the best of our knowledge, this is the first time VAEs have been used to learn a latent space for this dataset.

2 Methods

In this section, we will describe the various models used in this paper: VAE, VQ-VAE, and VQ-VAE2. While the continuous latent representation of VAEs offers benefits such as latent interpolation, having

Preprint.

discrete latent representations can be a more natural fit for certain modalities. Additionally, it has been shown that VAEs can suffer from posterior collapse, a phenomenon in which the learned latent space becomes uninformative. VQ-VAEs have been formulated to address these concerns, while still offering similar performance to VAEs.

2.1 VAE

Similar to vanilla autoencoders, variational autoencoders include an encoder which yields a lower dimensional latent representation and a decoder which reconstructs the input. However, VAEs compute an additional loss term that computes the Kullback-Leibler (KL) divergence between the encoder's distribution $q_\theta(z|x)$ and $p(z)$, where $p(z)$ is typically specified as a normal distribution with zero mean and unit variance. This term serves as a regularizer that keeps similar inputs' latent representations close together.

From a generative modeling perspective, imagine data point *i* is sampled by 1) drawing latent variable z_i and 2) drawing datapoint x_i based on z_i. Therefore, we would like to infer a good latent representation z given input *x*. This is given by the posterior:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{1}$$

However, computing the denominator $p(x)$ requires marginalizing over latents z and is often computationally intractable. Instead, variational inference allows us to approximate *p(z|x)* through another distribution *q(z|x)*, which is referred to as the approximate posterior. In other words, we would like to optimize the divergence between the approximate and original posteriors:

$$\min KL(q(z|x)\|p(z|x)) \tag{2}$$

The expression above cannot be optimized directly, since it requires computing *p(x)*. While we do not go through the derivation here, minimizing the KL divergence between *q(z|x)* and *p(z|x)* is equivalent to maximizing the evidence lower bound (ELBO), which is computationally tractable.

$$ELBO = E_{q_\theta(z|x)} \log p_\phi(x|z) - KL(q_\theta(z|x)||p(z)) \tag{3}$$

In practice, VAEs are modeled using neural networks; the approximate posterior is computed from the inference network (encoder) with parameters θ and the likelihood is computed from the generative network (decoder) network with parameters ϕ.

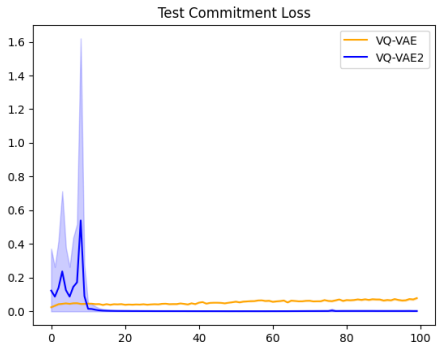
2.2 VQ-VAE

In order to learn discrete latent representations, the posterior and prior distributions are categorical instead of gaussian, and samples from these distributions are drawn by indexing an embedding table. The latent embedding space is $e \in R^{K \times D}$, where *K* is the size of the discrete latent space (i.e., a *K*-way categorical) and *D* is the dimensionality of each latent embedding vector e_i.The discrete latent variables z are calculated by a nearest neighbor look-up using the embedding table in a process referred to as Vector Quantization (VQ). Therefore, the forward computation pipeline can be viewed as a standard autoencoder with a non-linearity that maps the latents to one of *K* embedding vectors. Since performing a nearest neighbors lookup is non-differentiable, straight-through gradient estimation is used so that the gradient passed back is the same before and after quantization.

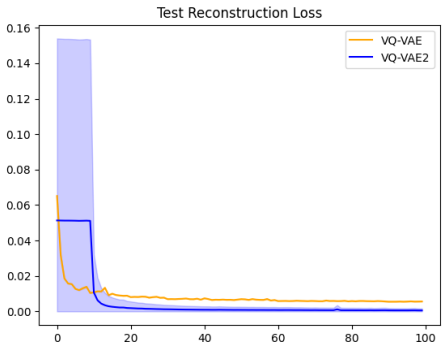
The loss function consists of three terms:

1. Reconstruction loss: Encourages original and reconstructed images to be similar
2. Codebook loss: Encourages the embedding vectors *e* to be close to the encoder output
3. Commitment loss: Encourages the encoder output to be close to the embedding vectors *e*

$$L = \log p(x|z_q(x)) + \|\text{sg}[z_e(x)] - e\|_2 + \beta\|z_e(x) - \text{sg}[e]\|_2 \tag{4}$$



(a) Commitment loss over 100 epochs.



(b) Reconstruction loss over 100 epochs.

The decoder optimizes the first term only, the encoder optimizes the first and last terms, and the embeddings are optimized by the middle term.

2.3 VQ-VAE2

VQ-VAE2 builds on VQ-VAE by using a hierarchical VQ-VAE model to encode images onto a discrete latent space, followed by learning a powerful PixelCNN prior. In this hierarchical architecture, the top-level latent code models global information and the bottom-level latent code, conditioned on the top-level latent, is responsible for representing local details. As a result the learned prior is also hierarchical; the top-level PixelCNN prior is conditioned on the class label, and the bottom level PixelCNN is conditioned on the class label as well as the top-level latent code. The authors show that the VQ-VAE2 framework produces samples that rival perceptual quality of GANs, while not suffering from lack of diversity.

3 Experiments

In a series of experiments, we compare the original VQ-VAE implementation to its variant VQ-VAE2. For VQ-VAE2, we implement the algorithm ourselves. For VQ-VAE we use an implementation from `https://github.com/nadavbh12/VQ-VAE`.

3.1 Kaokore Dataset

The Kaokore dataset consists of facial expressions extracted from pre-modern Japanese artwork. The dataset focuses on cropped face images extracted from Japanese artwork from the Late Muromachi Period (16th century) to the Early Edo Period (17th century) to facilitate research on art history and artistic style. The most recent version contains 8848 colored images, and each image is annotated with gender (male, female) and social status (noble, warrior, incarnation, commoner) labels. The paper associated with the dataset looks at classification as well as generative modeling, such as neural painting and style transfer models.

3.2 Results

	Commitment Loss	Reconstruction Loss
VQ-VAE	0.0793 (0.0027)	0.0056 (0.0001)
VQ-VAE2	0.0025 (0.0035)	0.0005 (0.0008)

Table 1: Commitment loss and reconstruction loss on the Kaokore testset for both algorithms. The numbers in paranthesis show standard deviation over three runs.

As indicated in table **??** and figure **??**, the primary advantage of VQ-VAE2 is the lower loss. This is achieved due to the better ELBO via the heirarchical model structure. The overall loss is composed

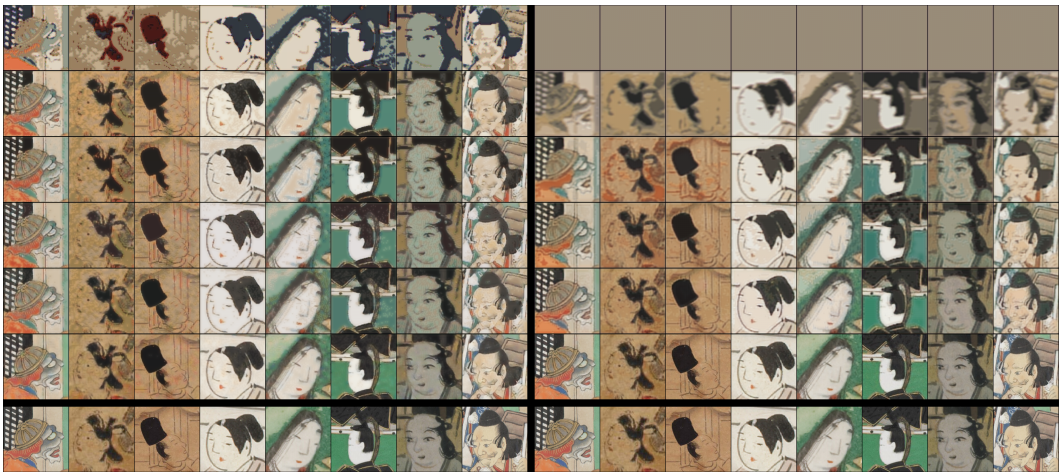


Figure 2: The first six rows of images show samples at 1, 12, 14, 16, 18, and 100 epochs for VQ-VAE (left) and VQ-VAE2 (right). The final row shows the cooresponding images from the testset.

of the reconstruction loss plus the commitment loss multiplied by a scalar. This trade off is the cause of the commitment loss increasing slightly as reconstruction loss lowers.

Another observation from these results is that VQ-VAE2 starts off training slower than VQ-VAE. We cannot find any reason to justify this in the algorithm itself, so we assume it is a result of untuned hyperparameters or a small bug in our code. Time permitting we would look into this further.

This delay in learning can also be seen in figure **??**. The first row shows that VQ-VAE2 starts off outputting noise for the first few epochs. Rows two through five show samples from the epochs where the loss of VQ-VAE2 changes the fastest. During these epochs, VQ-VAE2 starts to outperform VQ-VAE. Also note that VQ-VAE2 starts to learn lines and image structure before learning colors.

4 Conclusion

VQ-VAEs have quickly become a popular algorithm for generative machine learning. The improvements of VQ-VAE2 lower the reconstruction loss of the algorithm resulting in better performance. Our experiments using the new Kaokore dataset add additional support to this claim. Additional work is needed to verify whether VQ-VAE2 suffers from slower learning than the original implementation, but it is clear that VQ-VAE2 consistantly lowers the loss of the algorithm.