Summary of VAE_CartPole.1.1

Xupeng Zhu

16th May 2019

1 Introduction

In this code, a variational auto-encoder(VAE) was implemented and trained on screenshots of cart-pole environment on OpenAI gym environment. The image data set was collected as gray scale on the gym environment. Then the encoder was build and trained over the data set. New screenshots were reconstructed by the auto-encoder with respect to the 2D latent variable grid. Comparison was made between different dimension of the latent variable in the VAE.

2 Loss Function

The loss function used in the code is ELBO, which stands for the Evidence Lower BOund:

$$ELBO(\lambda) \;=\; \frac{1}{N}\sum_{i}(\mathbb{E}_{q_\lambda}(z|x_i)[\log p(x_i|z)] - \mathbb{KL}(q_\lambda(z|x_i)))$$

Where N is the size of training set, i is the ith image in the training set. There are two terms in the ELBO loss, the binary cross entropy loss term and the KL divergence term. The posterior $q_\lambda(z|x)$ that the encoder network approximates is Gaussian distribution, and has parameters μ and $\log\sigma$.

The encoder network has three fully connected layers. The decoder network has mirrored structure, except that the last layer of encoder outputs $2 * dim$ dimensions correspond to μ and $\log\sigma$ while the input layer of decoder is dim. dim is the dimension of latent variable z. The input gray scale images (40×90) are flattened and feed into the network. The training data set is consists of $20,000$ screenshots. The training uses Adam optimizer and SGD with batch size 100 and randomly sampled batch. After 30 epochs the loss was reduced, as showed in fig.1.

3 Example images of reconstruction

Fig.2 shows the screenshots that randomly selected from the training set. Fig.3 shows the reconstructed images according to 2D latent variables $z_1,\ z_2$. The

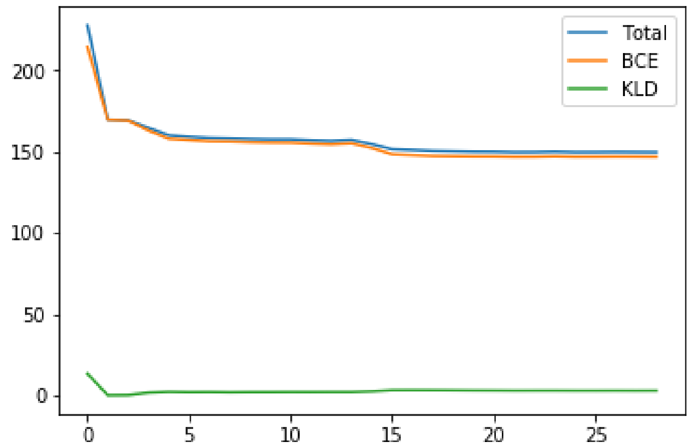

Figure 1: In this figure, Total stands for total ELBO loss, BCE stands for binary cross entropy loss, and KLD stands for Kullback-Leibler divergence.

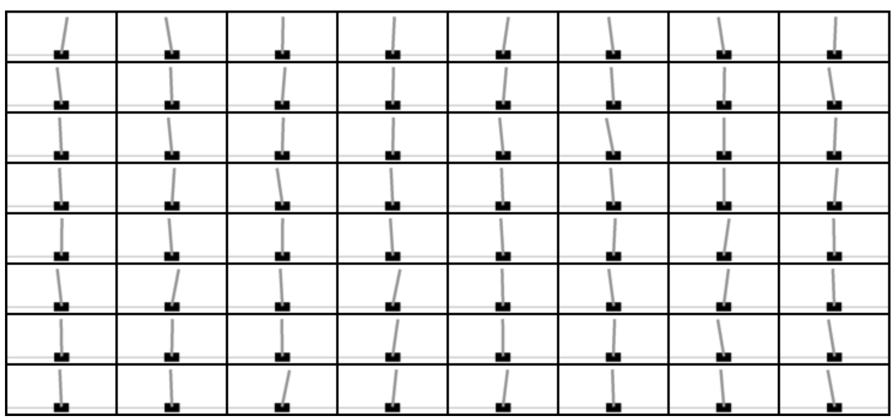

Figure 2: Randomly draw samples from the training set.

latent variables range from $(3,\ 3)$ to $(-3,\ -3)$. The $x,\ y$ -axis correspond to $z_1,\ z_2$ respectively.

From the fig.3 we can see that the posture of the cart pole only relies on z_1. This may due to the cart pole system only have one degree of freedom, and one variable z_1 is enough for the representation.

4 Evaluate the number of dimensions

In this section, several different dimension of latent variables (i.e., $dim = 2, 3, 10$) are compared.

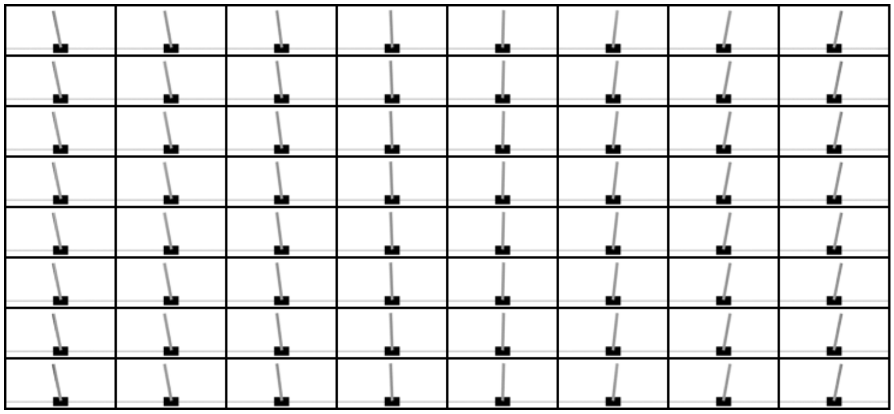

Figure 3: Reconstructed images according to $z_1,\ z_2$.

The loss function of 3 dimensions of latent variables converges most fast, follows by 2 and 10 dimensions ones. From the fig.4 we can see that binary cross entropy dominates the loss function and is keeping decreasing. All of the loss function converge to around 150 nats. This may due to some constant error.

Fig.5 shows the reconstruction by different dimensions of variables. Each finger nail image corresponds to an randomly sampled vector z obeys to standard normal distribution. The 2 dimensions VAE has most finger nail images that are blurry, the 10 dimensions VAE has less, and the 3 dimensions VAE has the most clear reconstructed images. Thus there should have a best fit dimensions for one kind of specific images.

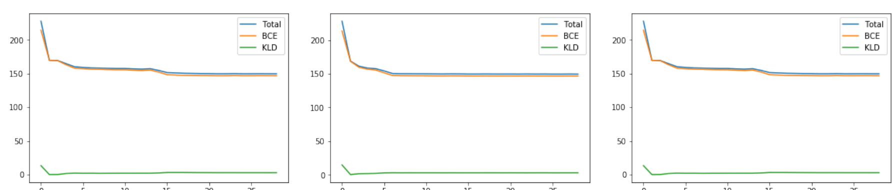

Figure 4: Comparison of loss function of latent variables with different dimensions during training. They are: 2, 3, and 10 dimensions, respectively.

5 Different coefficient on BCE and KLD terms

Here we define the coefficient $\alpha \in [0, 1]$:

$$ELBO(\lambda) \;=\; \frac{1}{N}\sum_{i}(2\alpha\mathbb{E}_{q_\lambda}(z|x_i)[\log p(x_i|z)] - 2\times(1-\alpha)\mathbb{KL}(q_\lambda(z|x_i)))$$

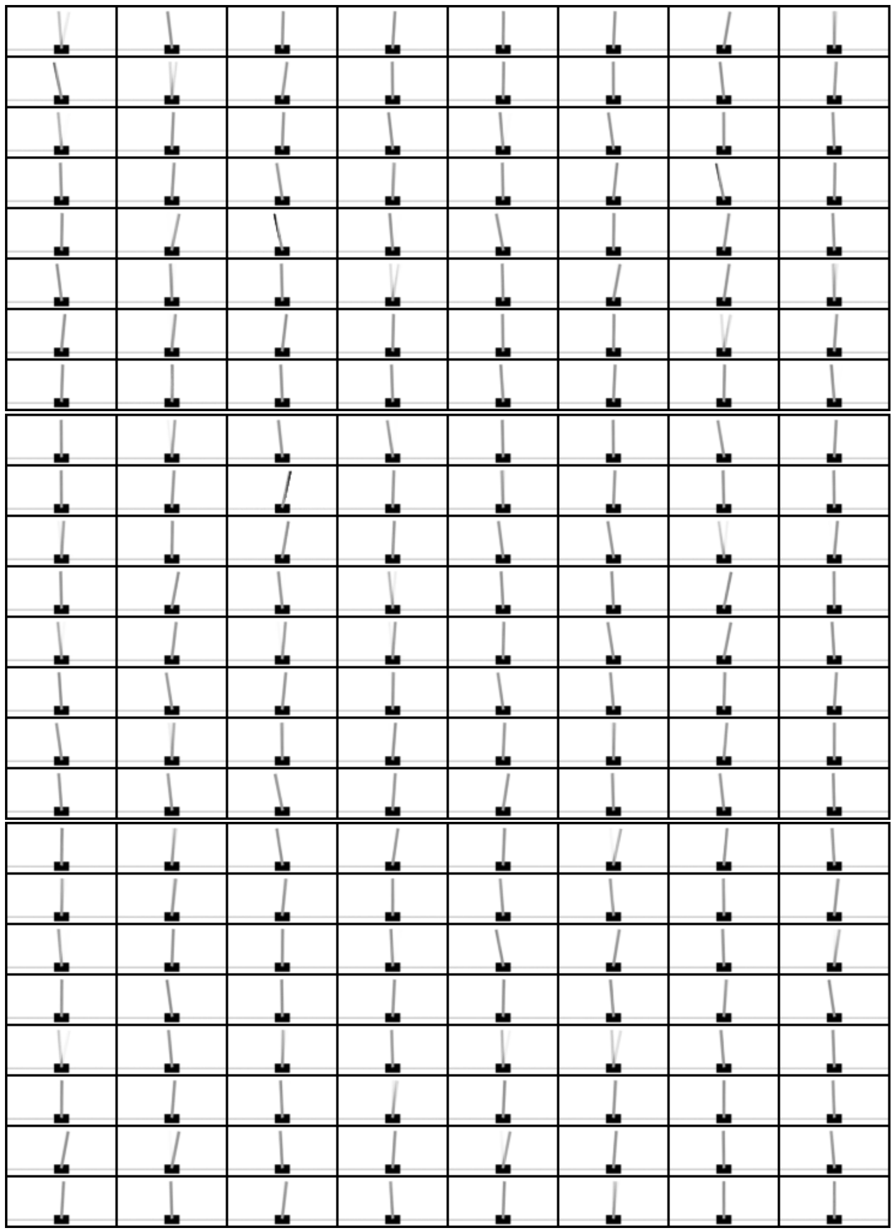

Figure 5: Comparison of VAE reconstruction by randomly sampled latent variables with different dimensions. They are: 2, 3, and 10 dimensions, respectively.

Comparison are made among $\alpha = 0.2, 0.5, 0.8$. Fig.6 shows the result. For each α, we trained three times for persuasion. The best result is $\alpha = 0.2$. Because fig(3, 1) and fig(2, 1) demonstrate these VAE have relate z_1 to the pole angle more clear than the VAE with other value of α. At this time the ELBO loss emphasises the KL divergence, which means the loss forces each latent variable follow the aimed distribution(i.e., Gaussian distribution). So the VAE learned a clear classification, i.e., classify z_1 to related to the pole angle.

6 Observation of two latent variable case

In this subsection, a VAE with two latent variables is trained with the images have movable cart and rotatable pole. Due to both the cart and the pole are free to move, we collected a data set that is 7.5 times lager as the previous one, say $150,000$ images. We also find that this VAE needs to be trained 500 epochs to obtain poorly-performance. The result is showed in fig.7. The latent variables range from $(10,\ 10)$ to $(-10,\ -10)$. The $x,\ y$ -axis correspond to $z_1,\ z_2$ respectively. This fig is the best result we had obtained after a lots of tries. Generally, we can observe that the left part of the figure relate to the cart in the center and the pole is lean to the right. The right part of the figure is interesting. In the right part, from top to bottom related to the cart in the center, the cart in the right-most, then the cart in the left-most, and the cart in the center. Moreover, the upper part related to the pole lean to the left while the lower part related to the pole lean to the right. However, the idea correspondence of the cart-pole and the latent variable is: the position of the cart and the angle of the pole should related to z_1, z_2 or z_2, z_1 respectively.

In this scenario, the VAE dose not perform well. We think the reason may be the training set. The training set was obtained by cart-pole with random action, the random action may not cover all possible state of the cart-pole, for example the cart in the right-most but with the pole lean to the right. More over, there are a large portion of the data set that the cart is near the center and the pole is vertical, this may lead to the bias in the training.

7 Visualization of distribution of 2-d latent variable (new)

In the following sections, new training and testing data set was used. Both data sets were collected by randomly initialize the CartPole env. Both the cart position, cart speed, pole position, pole speed were randomly initialized. Training set has $100,000$ images and testing set has 5000 images. Training set and testing set were collected separately and should have no overlaps.

Fig.9 shows the scatter plot of the mu generated by 2d latent variable VAE. A 2d VAE neural network was trained on the training set. Then 1e4 images were randomly sampled from the training set and feed into the VAE, the output mu$\in R^2$ for each image, were plotted in the first figure. The z variables for each image were generated by the VAE's encoder, according to the mu and variance. In the second figure, the yellow dots show the z corresponding to the 1e4 images. For comparison, mu for 1e3 images were plotted in blue.

Amazingly, the first image in fig.9 shows the mu distribution has a pattern. The dots corresponding to the mu were distributed along radius randomly but were distributed discretely along circumference. Moreover the distribution is like a 2d manifold in 3d space, then flatten into 2d plot. If the manifold can be unfold, it should be a rectangle, or we say it's topology is a 2d cylinder. This cylinder may be an idea state space. The second image in fig.9 is the sampled z. The sampled z distribute similarly to mu, but loss the discretization, due to the assumption the z follows Gaussian distribution.

8 Comparison on different dimensions of latent variable (new)

We compared the loss of training on VAEs that has latent variable dimension in 2, 4, 8, 16 and 32. When the latent variable dimension is 8, the loss decreased the fast. All of the loss of the VAEs converged to the same value eventually, except the VAE has 2 dimension. This may due to the small dimension cannot contain enough information.

9 Reconstruction comparison for 8-d latent variable (new)

We choose the best performance VAE, i.e., has 8 dimension latent variable. The VAE was trained on the training set for 100 epoch, the loss is show in fig.10. Then we randomly sampled images from the testing set, feed into the VAE and reconstructed the new images. The first figure in fig.11 shows the randomly sampled images, the second figure shows the corresponding reconstructed ones.

The VAE can reconstruct a clear image. However, there are several cart-pole configuration were reconstructed wrong. They were circled with red rectangle in fig.11. From the fig.11, for all of the three wrongly reconstructed images, the car position is correct, but the pole angle is obviously in the wrong side. This may lead to problem.

References

[1] Reinforcement learning (dqn) tutorial.

[2] Tutorial - what is a variational autoencoder? – jaan altosaar.

[3] Lyeoni. lyeoni/pytorch-mnist-vae, Oct 2018.

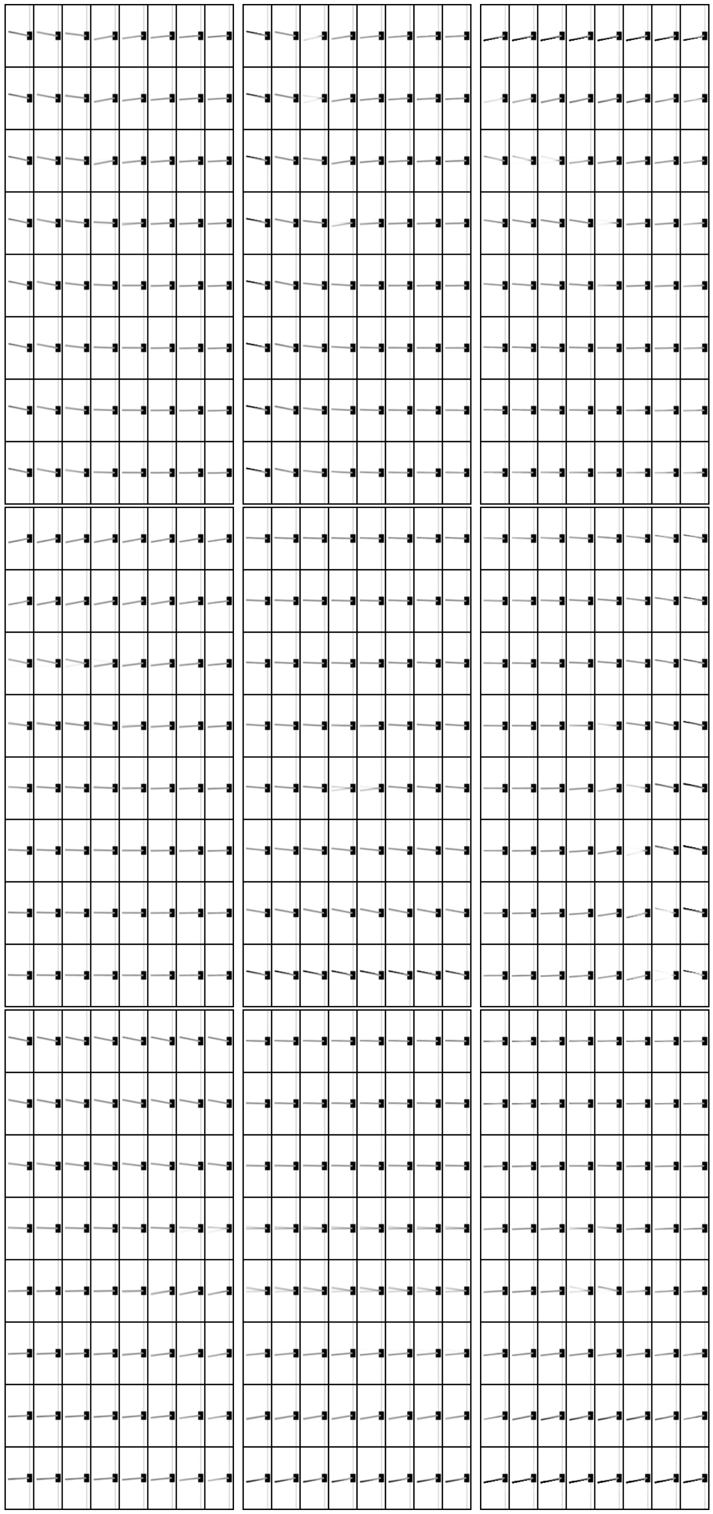

Figure 6: Comparison among $\alpha = 0.2, 0.5, 0.8$. The columns along the length of the paper are for $\alpha = 0.2, 0.5, 0.8$, the rows are correspond to the first, second, and the third training results.

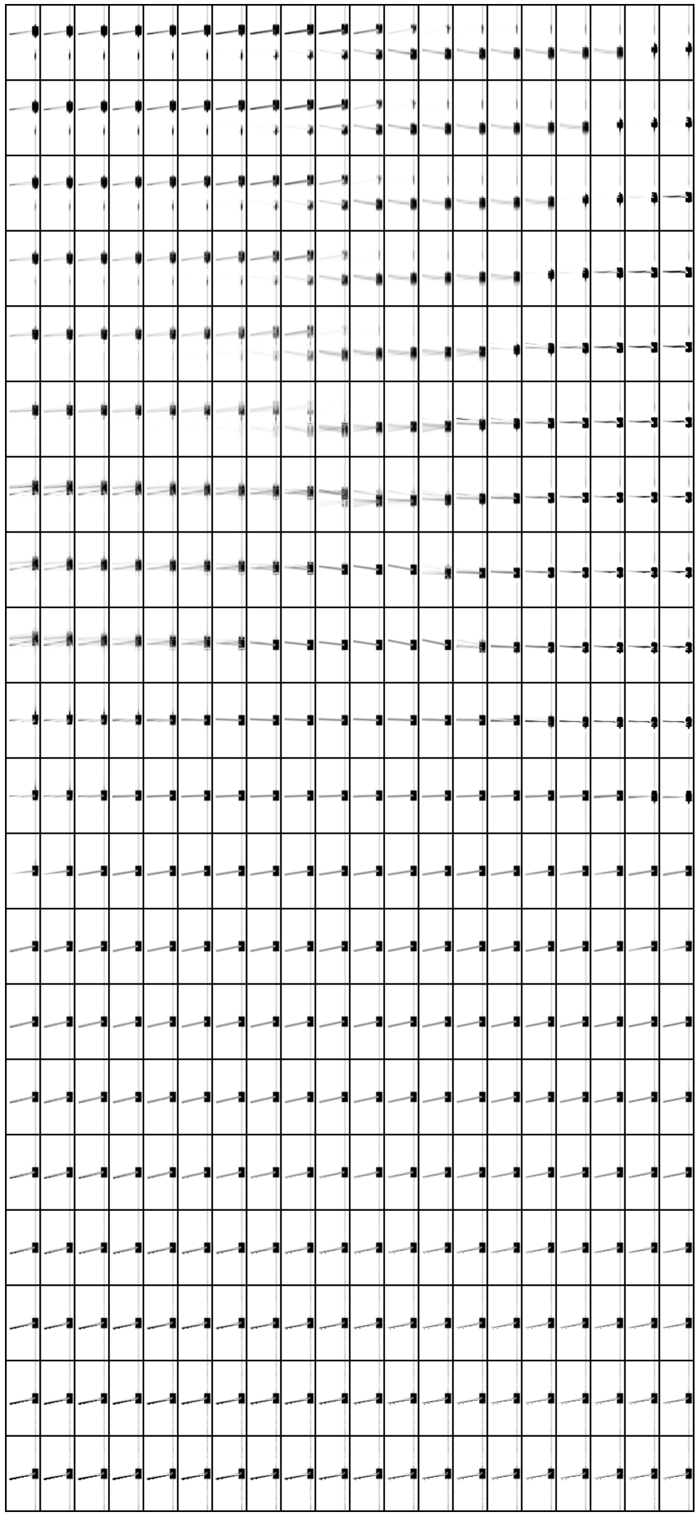

Figure 7: Reconstructed images according to $z_1,\ z_2$. Notice that here both cart and pole are movable.

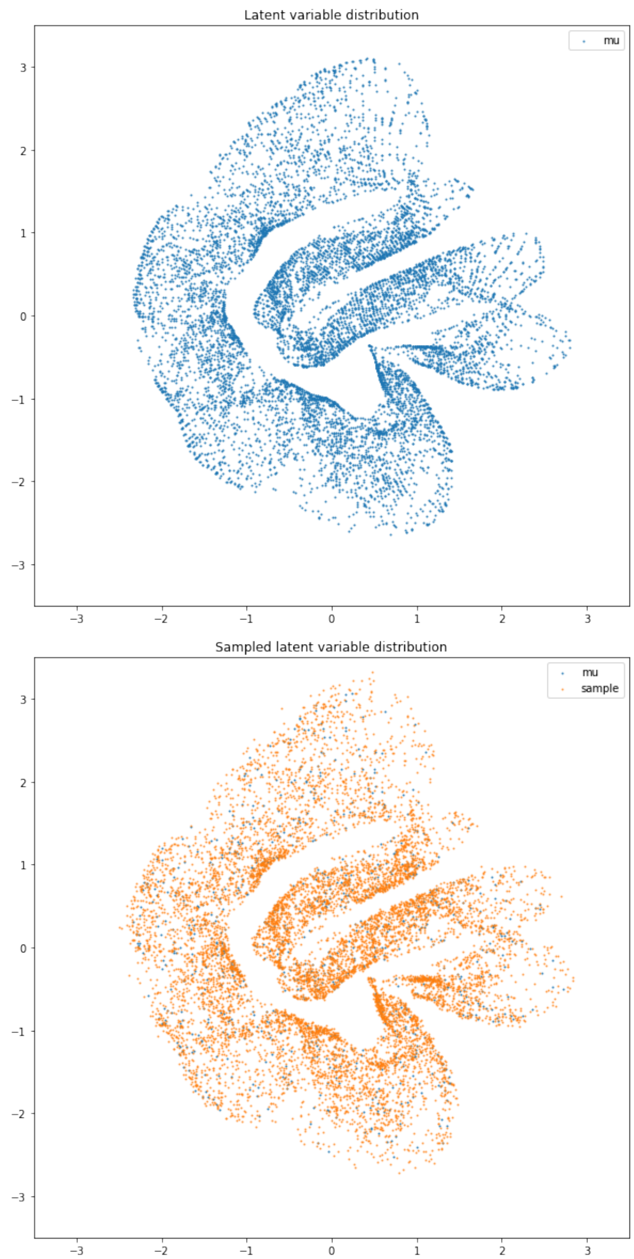

Figure 8: Scatter plot of latent variable, i.e., mu. X-, y- axis correspond to mu1, mu2, respectively.

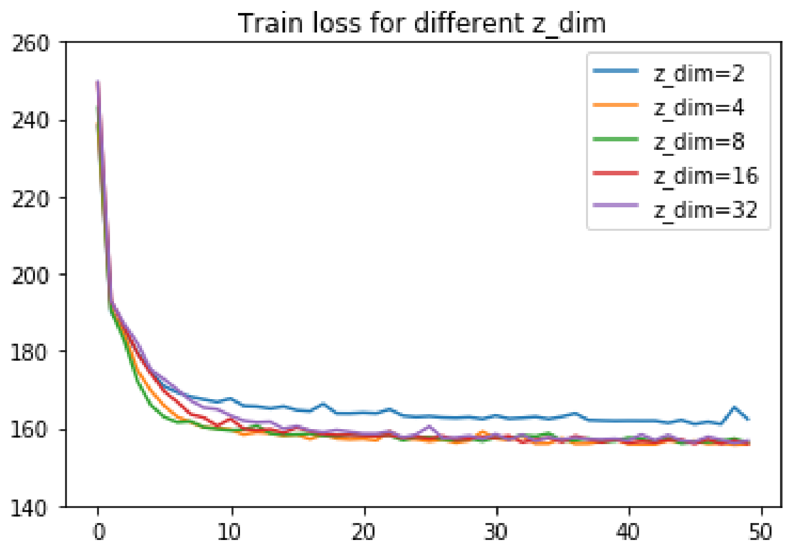

Figure 9: Comparison on VAEs with 2, 4, 8, 16 and 32 dimension of latent variables.

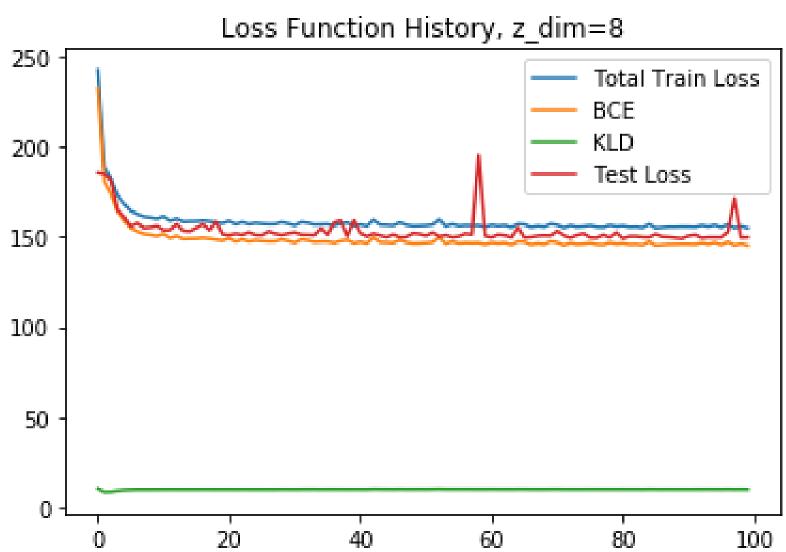

Figure 10: The detailed loss function for VAE with 8 dimension latent varibale.

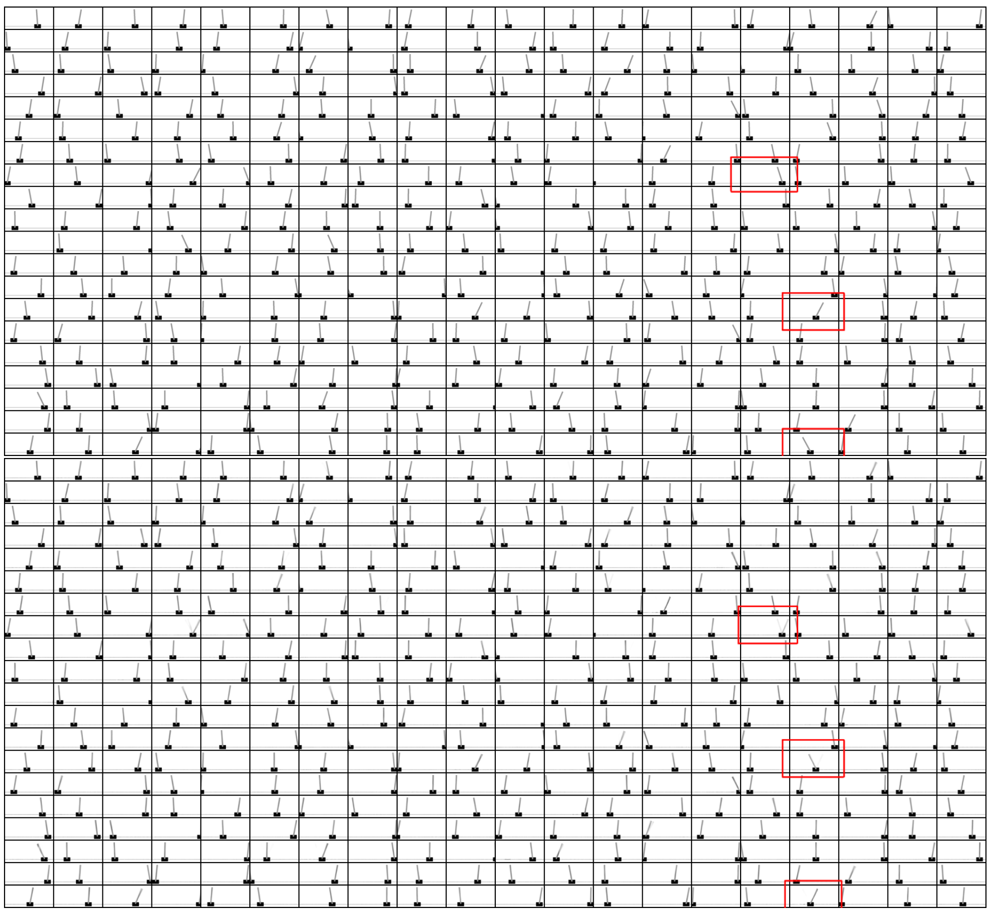

Figure 11: First figure: the origin images in test set. The second figure the reconstructed images, corresponding the ones in the first figure.