Variational Auto-Encoder in Tensorflow

Hamid Karimi

May 10, 2017

Michigan State University
Department of Computer Science and Engineering
428 South Shaw Lane
East Lansing, MI 48824

Variational Auto-Encoder (VAE)

Figure ?? shows Swiss-roll distribution generated by provided transformation.

Figure 1: Real Swiss-roll distribution

Loss function

Equation ?? shows the loss function.

$$L = \frac{1}{2} \times \frac{1}{M} \Bigg(\sum_{k} (x_k - \hat{x}_k)^2 - \sum_{k} \sum_{j=1}^{2} \Big(1 + \log(\sigma_j^k)^2 - (\mu_j^k)^2 - (\sigma_j^k)^2 \Big) \Bigg) \tag{1}$$

M is number of batches, $1 \leqslant k \leqslant M$ is index to batch samples, $\hat{x}_k$ is generated sample by decoder, $1 \leqslant j \leqslant 2$ in z dimensions, $\log(\sigma_j^k)^2$ and (μ_j^k) are obtained from encoder network. Indeed, $\log(\sigma_j^k)^2$ and (μ_j^k) are parameters of the network which their optimal values are to be found by multilayer neural network. The first part of equation ?? is reconstruction loss which is simply sum of square root difference between ground truth value x_k and generated sample $\hat{x}_k$. The second term is KL divergence between $p(z)$ and $q(x|z)$ when $p(z)$ is Gaussian distribution with mean and standard deviation of μ and σ, respectively.

Model design

Table ?? shows network layer design and other settings. At first, I just had one hidden layer with Relu as non-linear function. But I changed it to two hidden layers with tanh which turned out to converge faster.

Training

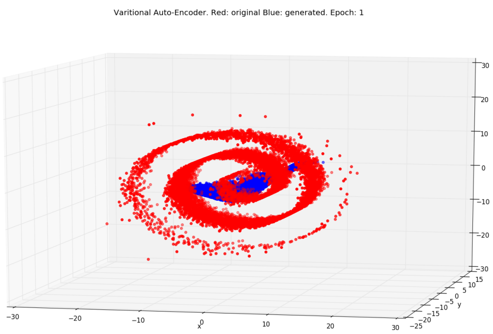
At each iteration, a batch from Swiss-roll distribution is fed into the encoder network. The encoder produces $\log \sigma^2$ and μ for each input instance. The latent variable z is achieved as $z = \mu + \sigma \times \epsilon$ where ϵ is $N(0, I)$ [?]. Decoder takes z and generates (fake) sample $\hat{x}$. The generator generates desired Swiss-roll distribution at epoch 200. However, I did train the network for 1000 epochs.

Results

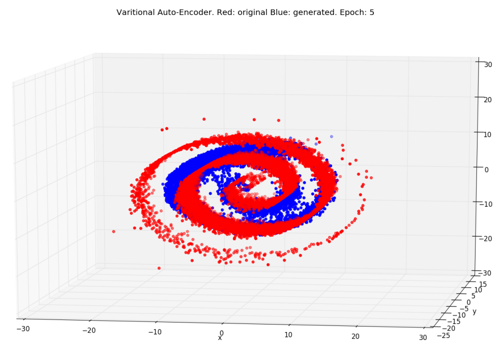
Figure **??** shows original distribution and generated ones in the same plot for different epochs. In epoch 201, they match each other quite well. Figure **??** plots evolution of generated distributions at different epochs. As you can see, as iterations go on the shape is getting more similar to real Swiss-roll distribution . Figure **??** shows plot of Variational Free Energy (VFE) vs iterations. VFE decreases as iteration increases. In order to show the decline more clearly, initial big values of VFE have been removed from the curve.

Encoder	Layer1	100 units. Non-linearity : tanh
	Layer2	100 units. Non-linearity : tanh
	Layer3	4 units. Linear (yielding latent variables μ and $\log\sigma^2$)
Decoder	Layer1	100 units. Nonlinearity : tanh
	Layer2	100 units. Nonlinearity : tanh
	Layer3	3 units. Linear (yielding generated points)
z dimension	2	
Batch size	100	
Learning rate	0.001	
Optimizer	AdamOptimizer	
Number of Swiss-roll points	10000	
Number of trained epochs	1000	
Convergence epoch	200	

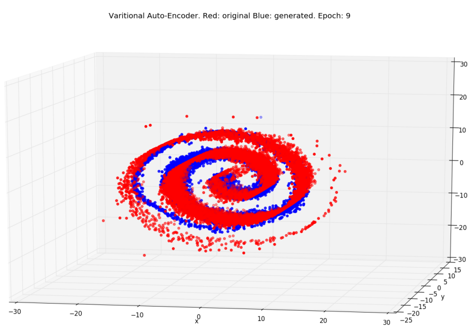
Table 1: VAE network architecture and settings



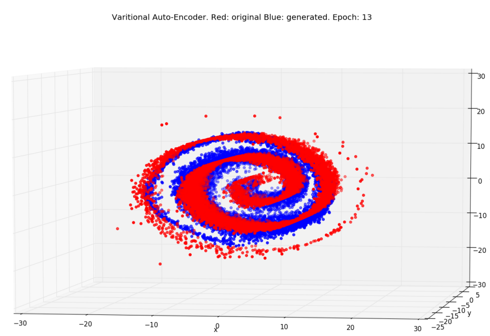
(a) epoch 1



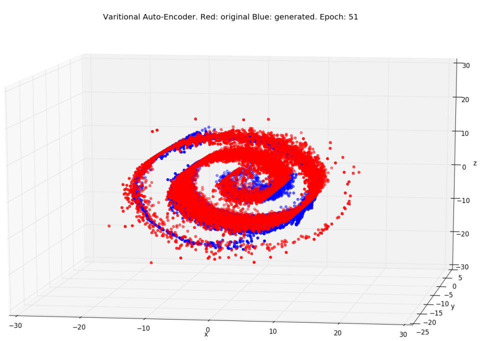
(b) epoch 5



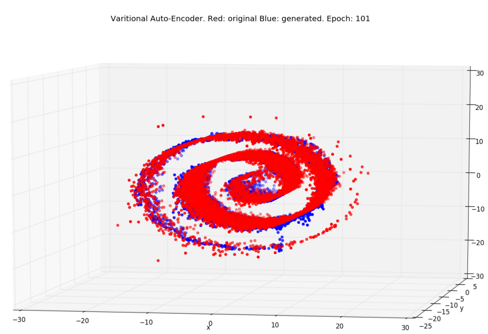
(c) epoch 9



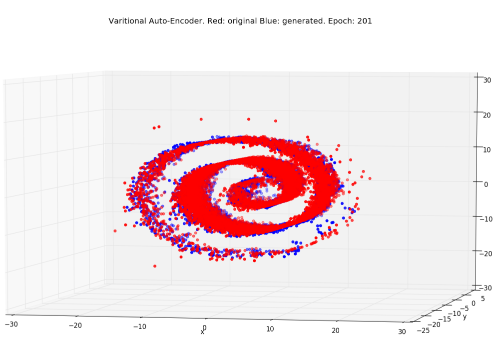
(d) epoch 13



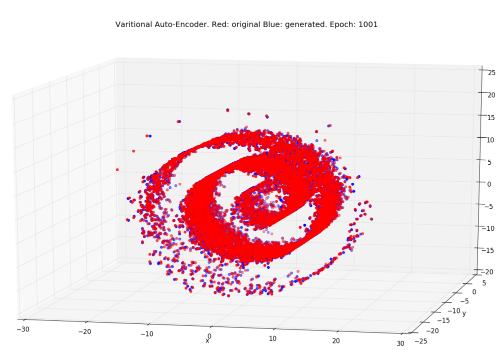
(e) epoch 51



(f) epoch 101

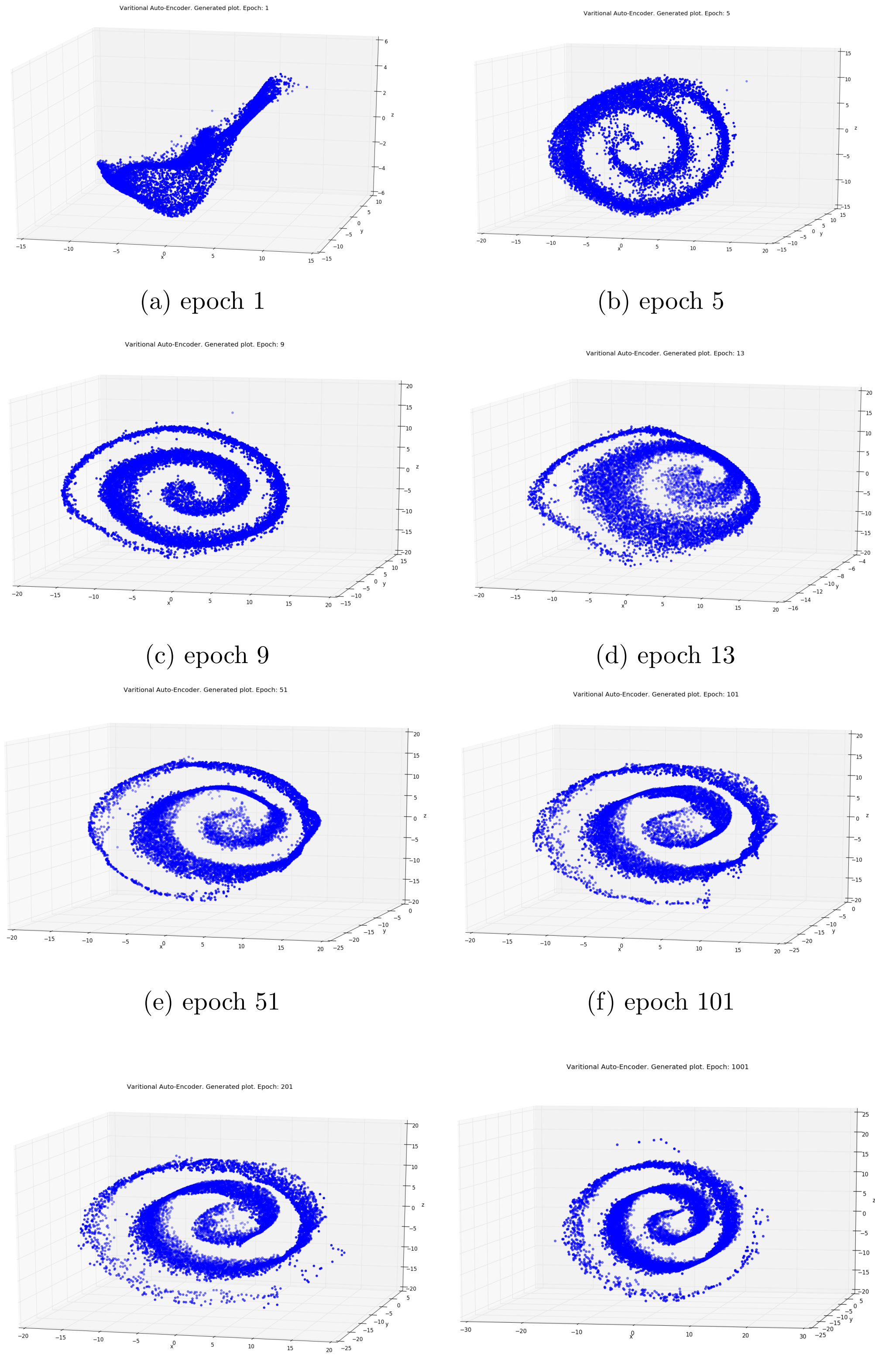


(g) epoch 201



(h) epoch 1001

Figure 2: Variational Auto-Encoder. Red plot is original and blue is generated



(a) epoch 1

(b) epoch 5

(c) epoch 9

(d) epoch 13

(e) epoch 51

(f) epoch 101

(g) epoch 201

(h) epoch 1001

Figure 3: Variational Auto-Encoder generated distributions

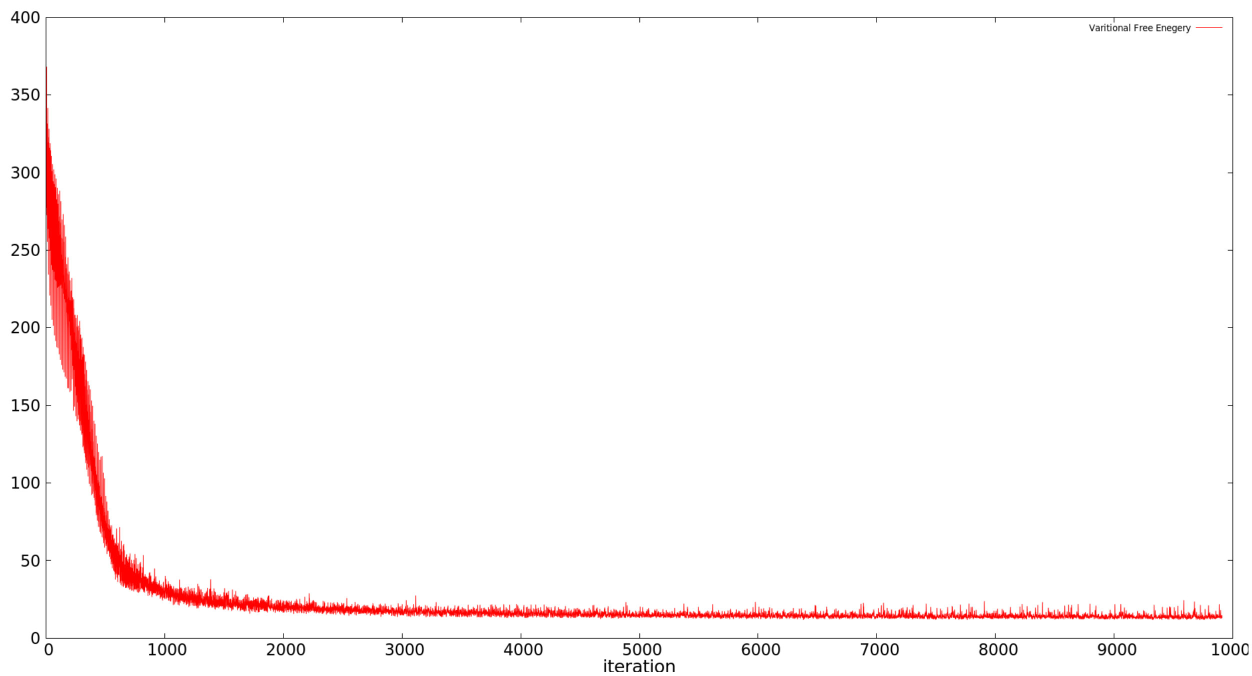


Figure 4: Variational Free Energy