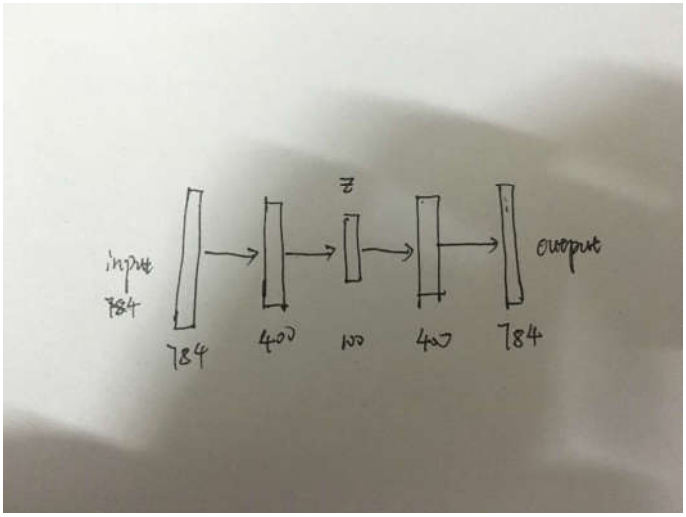
Machine Learning Experiment3

A. Visuliazation method: t-SNE(similar to LDA and PCA, but is better than both)

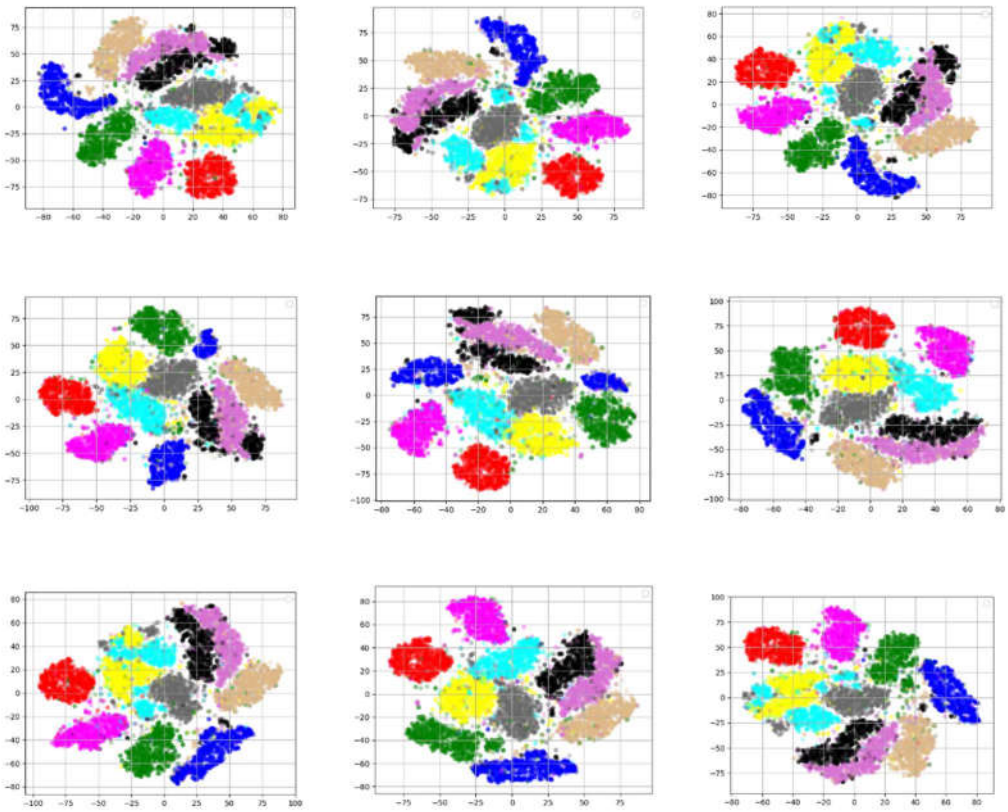
Architecture of autoencoder:



Results:

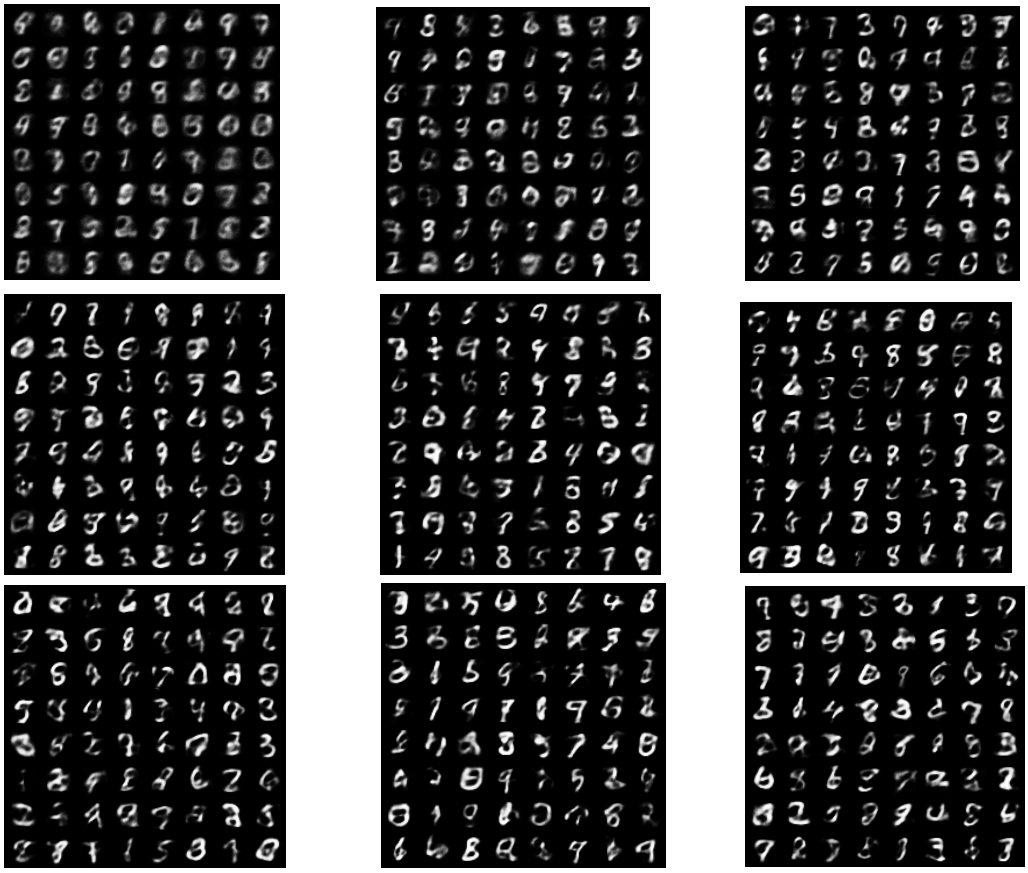
Visulization result:

This is the test dataset visualization results, not the training dataset.



From left to right, top to bottom: epoch 1-9.

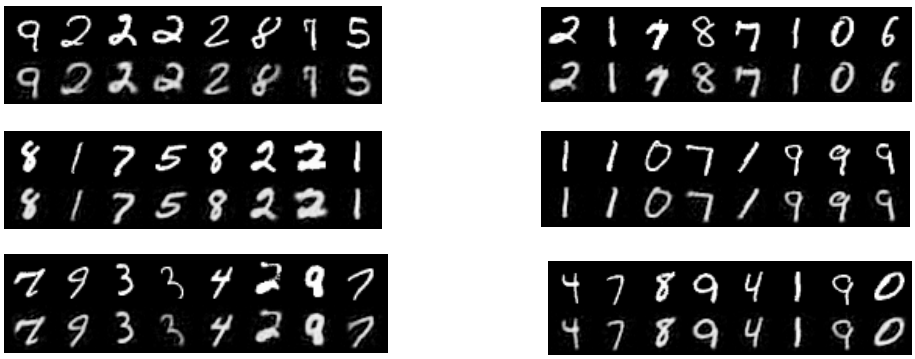
Generation result:



From left to right, top to bottom: epoch 1-9.

Input: noise z from normal distribution

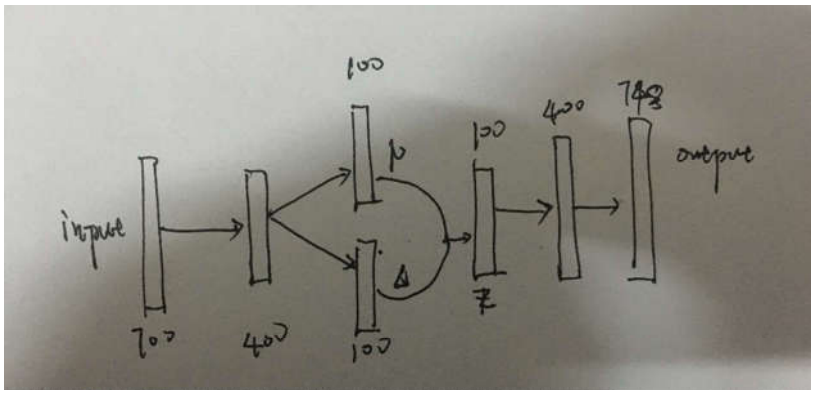
Reconstruction result:



From left to right, top to bottom: epoch 1-6.

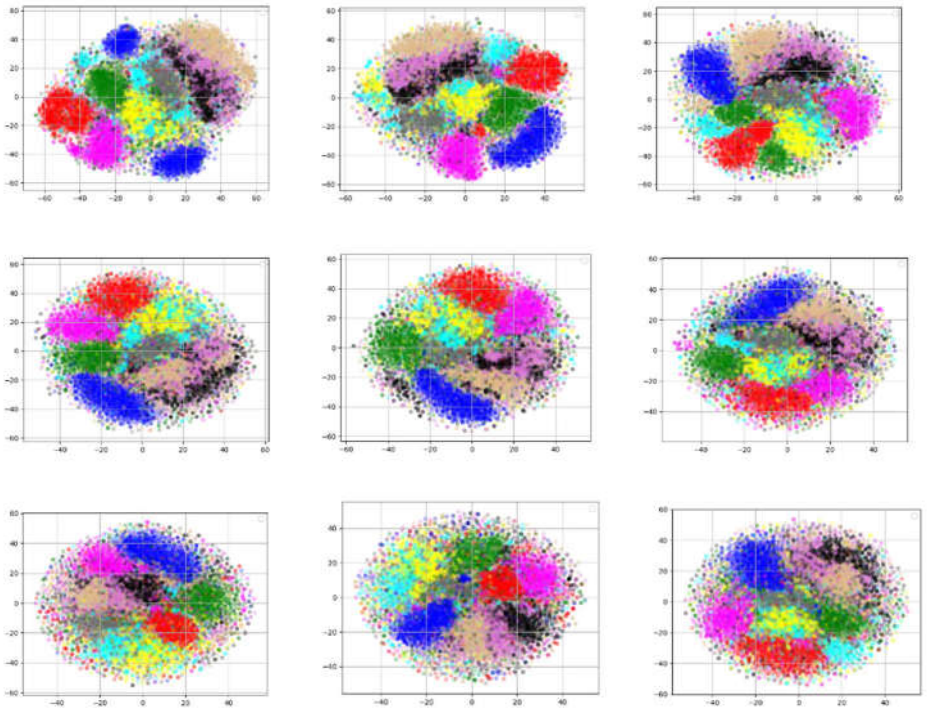
I do not use constractive autoencoder or sparse autoencoder, but use a more powerful and beautiful model- variational autoencoder(VAE)

Architecture of VAE:



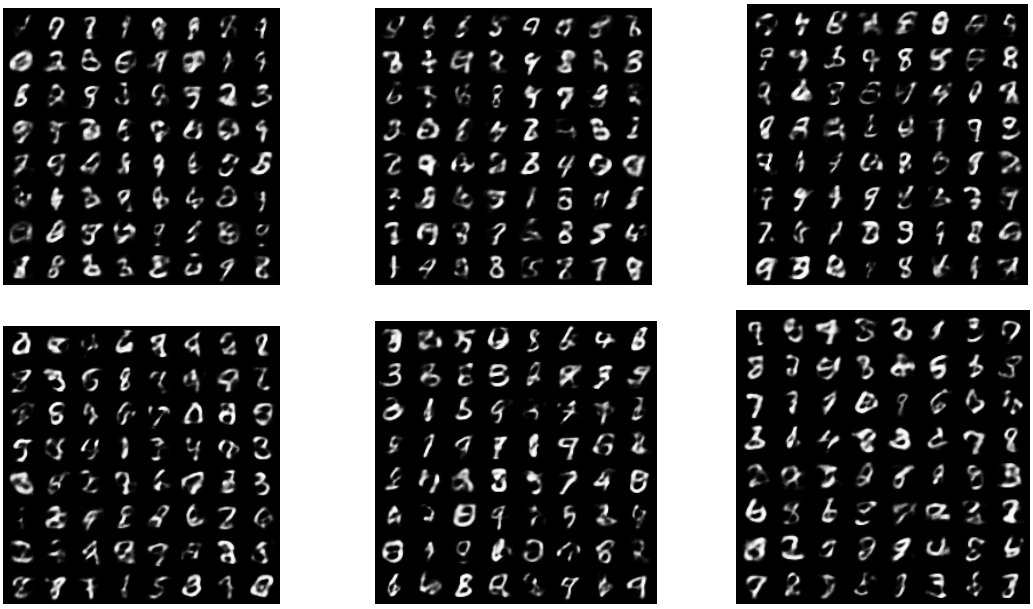
Results:

Visualization result:



From left to right, top to bottom: epoch 1-9.

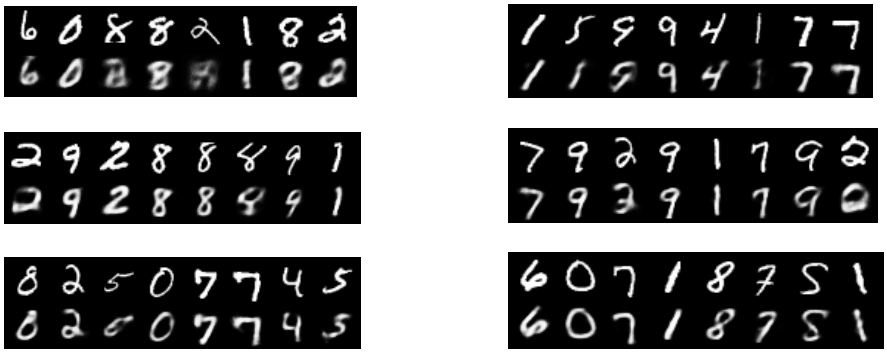
Generation result:



From left to right, top to bottom: epoch 1-9.

Input: noise z from normal distribution

Reconstruction result:



Comparison between autoencoder and variational autoencoder

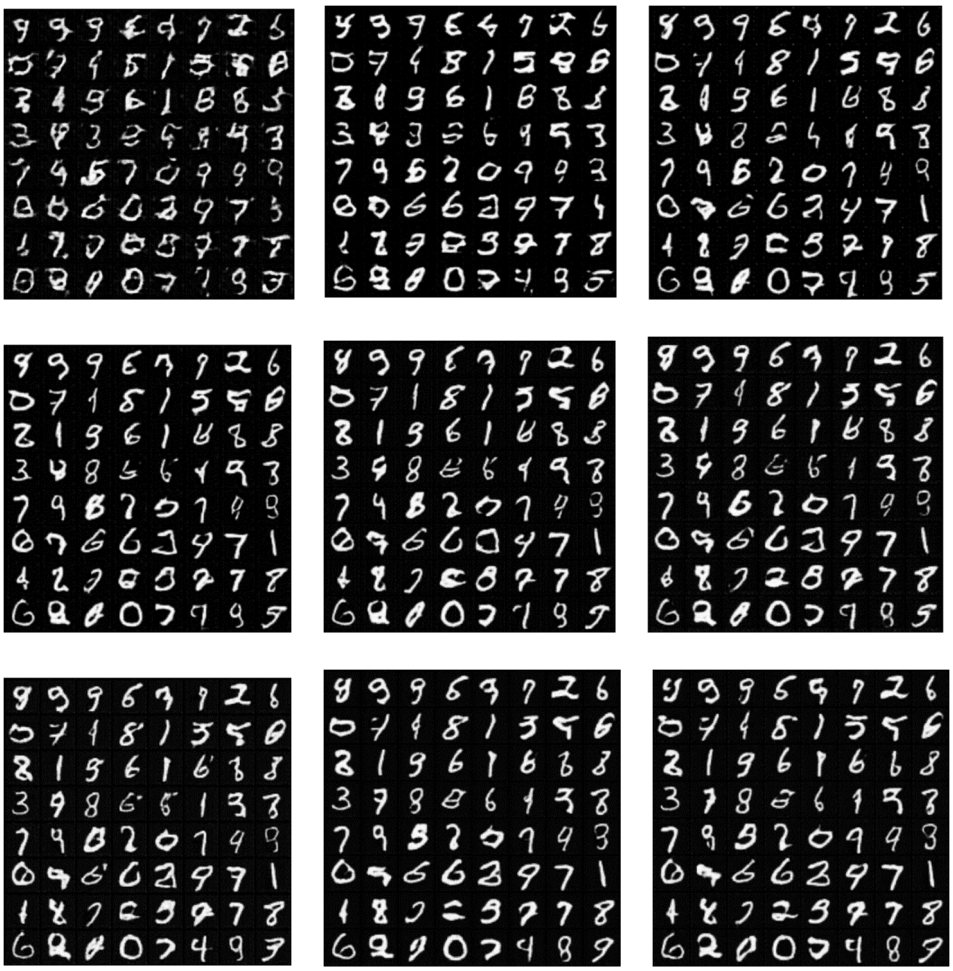
The clustering result: autoencoder is better, because it does not introduce much noise.

The generation result: VAE is better, because it can be continuous.

Reconstruction result: both are similar.

B. DCGAN architectiure is the same as traditional one.

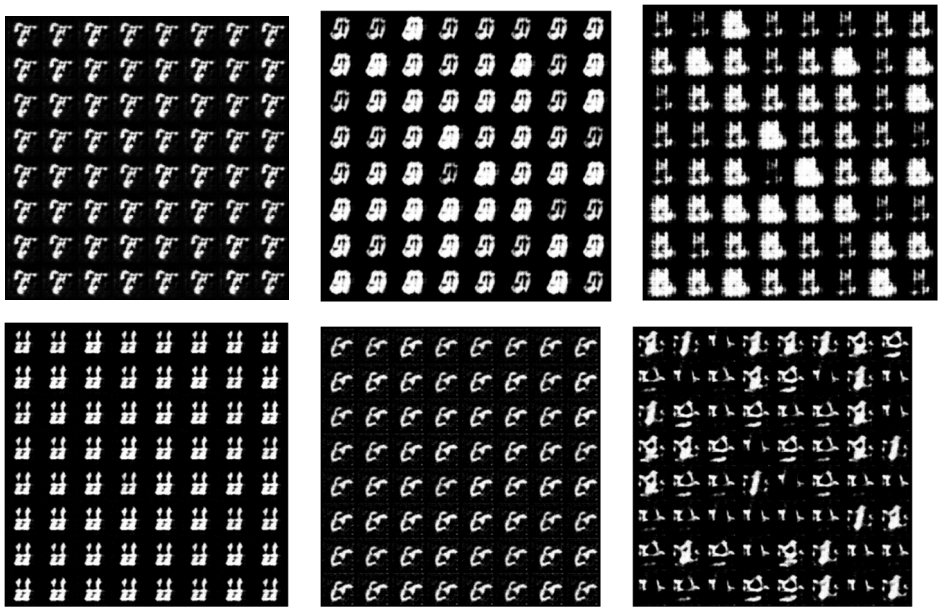
Original DCGAN Results: normal distribution



From left to right, top to bottom: epoch 1-9.

Input: noise z from normal distribution

(1) Results from uniform distribution



From left to right, top to bottom: epoch 1-9.
Input: noise z from uniform distribution

Analysis:

<1> at the first beginning of the training, it takes more time to train the model until it can produce real-like image when using uniform distribution.

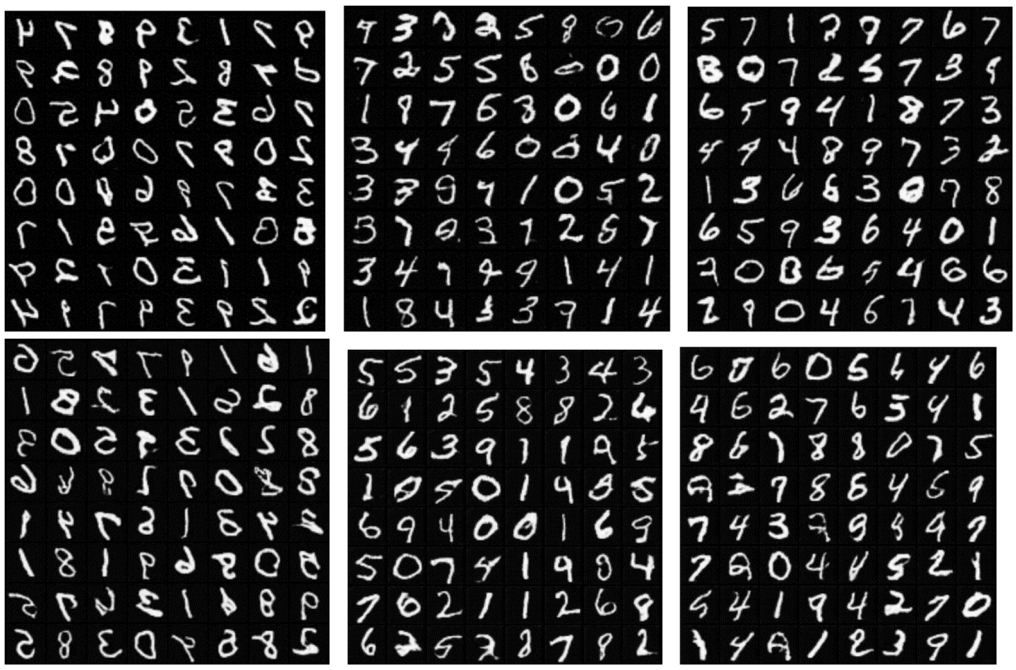
<2> My intuition says that if I learn from a z~N(μ,σ) latent variable that has a small σ, that would mean that the samples my generator would produce would be very sparse and similar to each other; as if I were spanning a smaller space of Generated samples for my generator function G(z) and the discriminator D(x) would have a harder time discriminating between real and fake data if G(z) is very close to x~pdata(x).

On the other hand if z~U[0,1]

, then perhaps the variability is less likely, but I believe that with an equally likelihood of exploring all the [0,1] space then the Generator has a higher exploration space if viewing this from an exploration vs exploitation framework (as G(z) is equally likely to hit all locations in the Generator space, vs a subset).

(2) Using the latent representation of the autoencoder as the source of GAN's z vectors

Results:



Analysis:

When using the input z from autoencoder feature, the gan converages faster to generate real image.
I think the reason is that when using normal or uniform distribution, the generator need to remember the whole distribution which is not possible, however, if we use the autoencoder features as the input z, the generator only needs to remember the autoencoder features which is possible.

C.

replacing the generator from part B with the decoder portion of the autoencoder from partA. Can the training of the modified GAN somehow benefit from the availability of the trained autoencoder (from A).

discovery: (1) when use a pretrained decoder from an autoencoder, the model converages faster when fine tuning.

Epoch 1-4