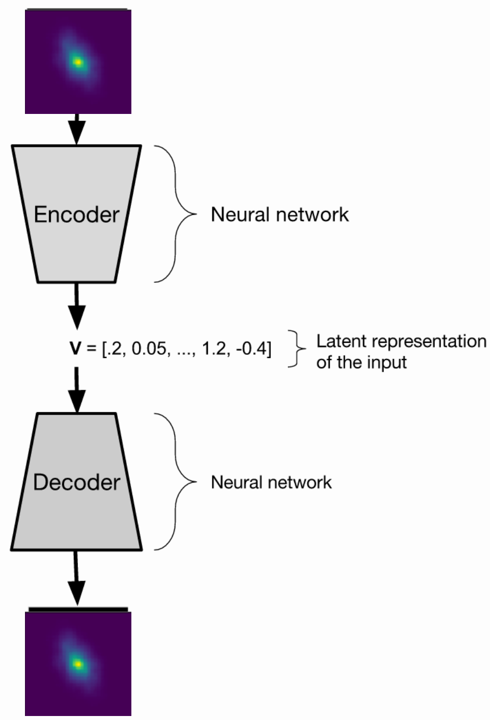
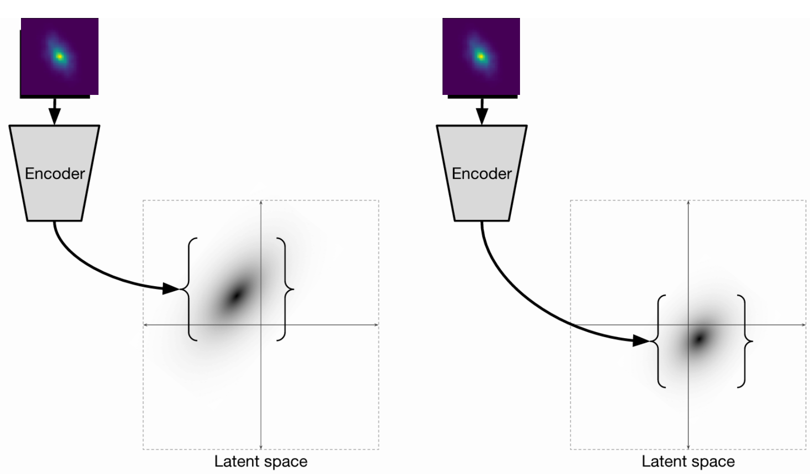
In Auto-encoder(AE), we map the input images to a low dimensional hidden space, like the V layer in the following figure:



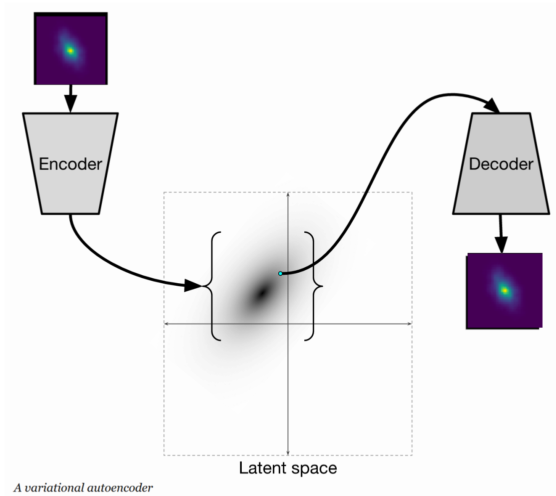
A standard autoencoder

The loss function of AE measures the pixel-wise difference between input images and output images. We may call this self-supervised learning. By performing down-sample and upsample process, hot pixels will be removed.

However, if we want to use a given V to generate a image, the result may be bad. Because here the V space is not that continuous/smooth.



The encoder of a variational autoencoder



A variational autoencoder

To solve this problem(to generate new samples), we may use a variational auto-encoder(VAE). VAE is always used as a generative model to generate new samples.
For the Encoder part of the VAEs, a certain class of input images are mapped to a certain Multi-dimensional Gaussian distribution. And then the Decoder use a resampled hidden value to generate a new image.

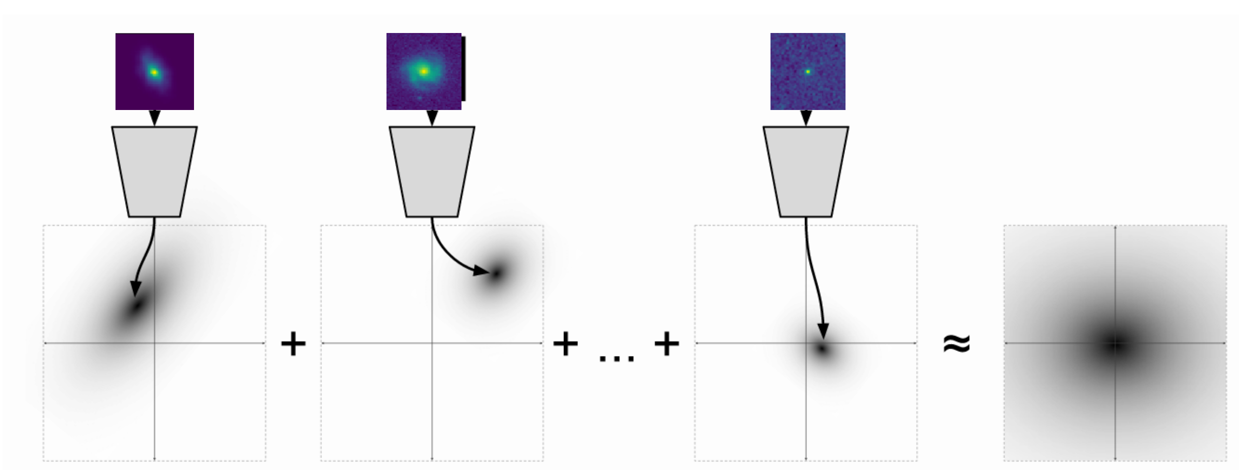
So the first part of loss function here is also the same: pixel-wise difference between input images and output images.

Then, for better performance in generation tasks, we hope that the total distribution can also obey a normal distribution.
We can use KL-divergence to describe the difference between two distributions:

$$P(x) \sim \sum_{i}^{n} N(\mu_i, \sigma_i^2) \quad Q(x) \sim N(0,1)$$

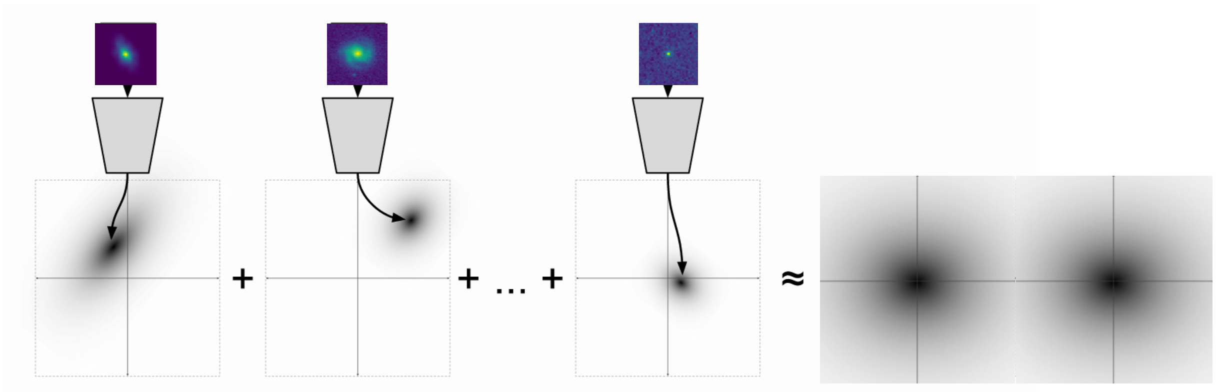
$$D_{KL} = \int_{-\infty}^{\infty} P(x) \log \frac{P(x)}{Q(x)} dx$$

(The optimization process is like using stochastic gradient descent)



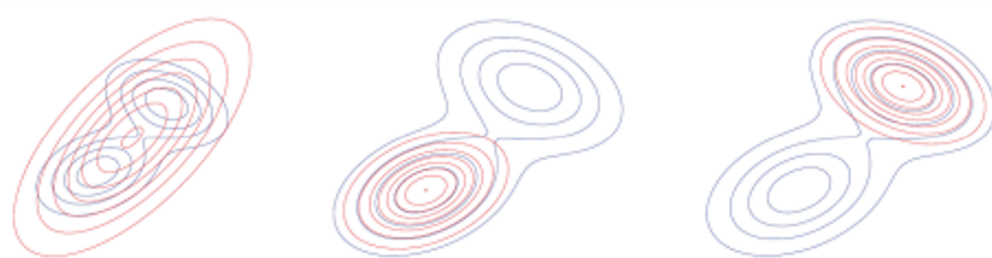
So, the KL-term here act as a regularizer that can restrict V_mean and V_variance. The final result is we can use any point in this hidden space to generate a quite good image that at least looks like one of the several classes.
This N(0,1) priori is good at generating new images, but not conducive to unsupervised classification.

To separate different classes into two or more clusters, a better choice of the priori distribution can be a double-peak Gaussian.



$$P(x) \sim \sum_{i}^{n} N(\mu_i, \sigma_i^2) \qquad Q(x) \sim \frac{1}{2}N(-1,1) + \frac{1}{2}N(1,1)$$

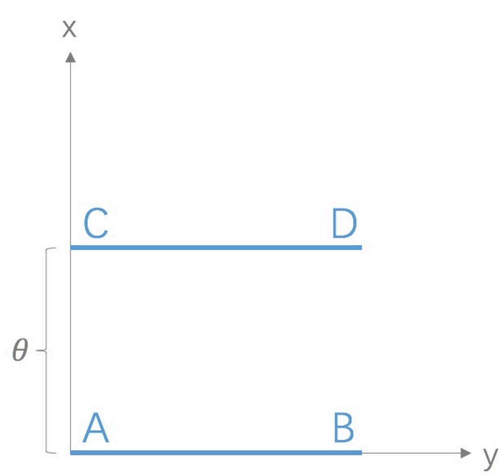
However, KL divergence works bad in such situation.



(a) (b) (c)

1. Asymmetry: D(P||Q) ≠ D(Q||P)
2. Gradient disappearance:
   JS_divergence = log2 = const

Their KL divergence is infinity



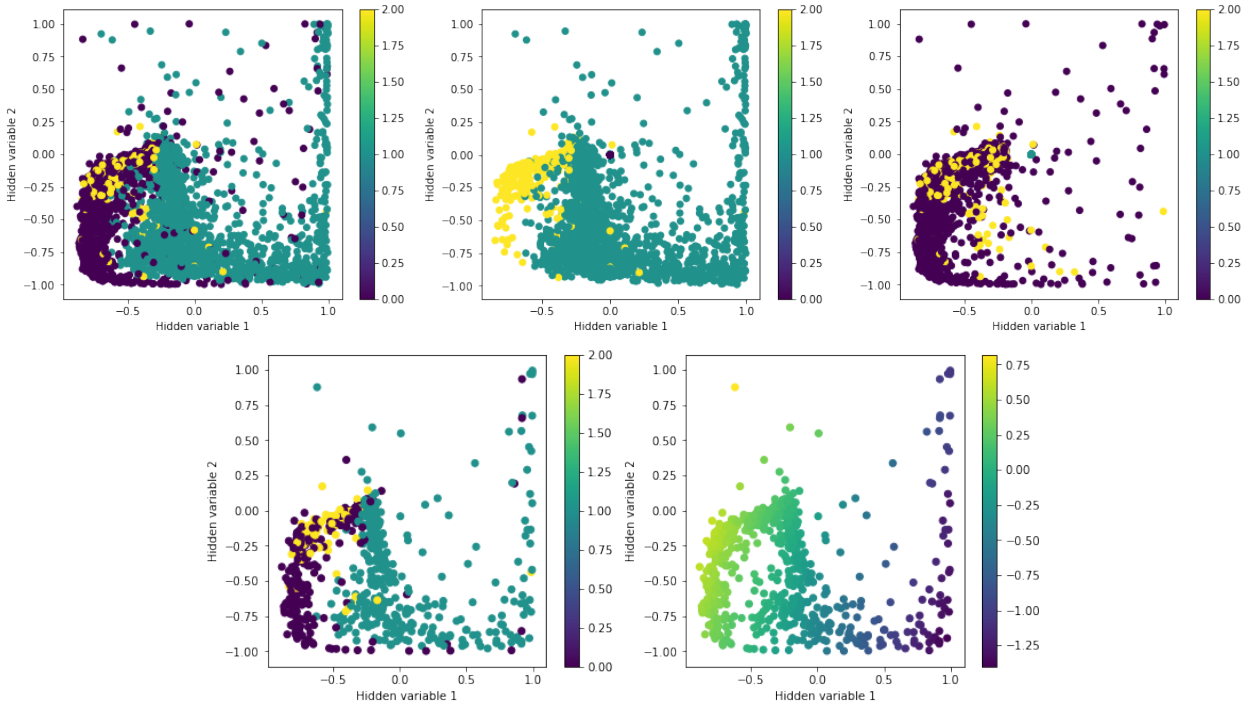
when \theta≠0, and 0 when \theta=0
Their JS divergence is log2 when \theta≠0, and 0 when \theta=0and
In Wasserstein metric,

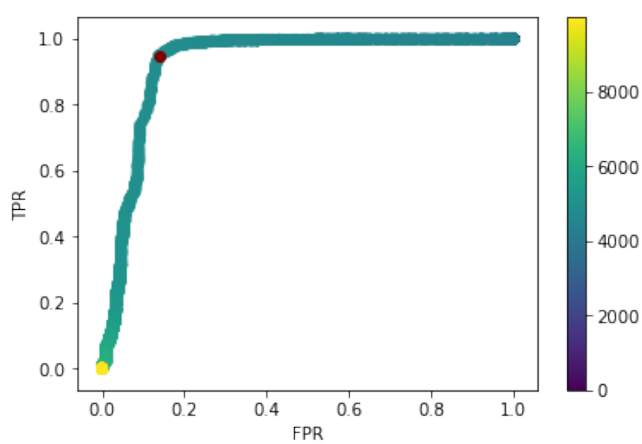
W_loss = |\theta|

The gradient exist even when those two distributions have no overlap.

(TODOs: I need some time to understand Wasserstein metric)

A Wasserstein loss analogy shows quite good improvement, but still not very stable.





AUC: 0.92

In 12 experiment, the AUCs are:

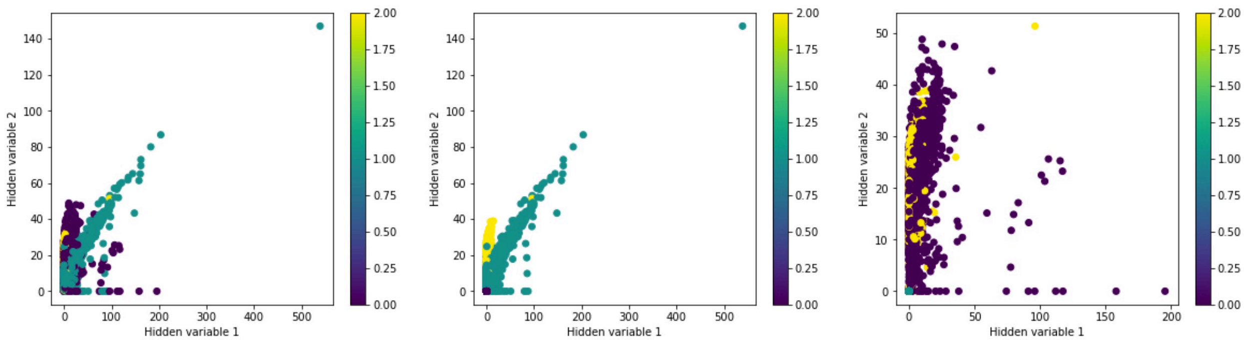
AUC: 0.874056800995
AUC: 0.890954309319
AUC: 0.924916904083
AUC: 0.907576218669
AUC: 0.894452798743
AUC: 0.901569885906
AUC: 0.761481311188
AUC: 0.872700482459
AUC: 0.858911459225
AUC: 0.882133929019
AUC: 0.899523653834
AUC: 0.724179096412

This is much better than before, when I use the single Gaussian priori(in 22 experiment):

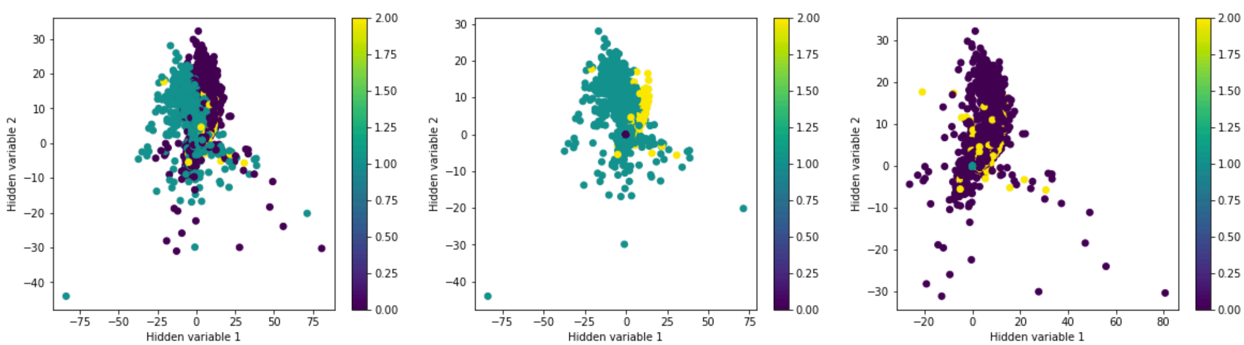
total accuracy is: 0.860203648706
total accuracy is: 0.816574741904
total accuracy is: 0.889619572903
total accuracy is: 0.833050487908
total accuracy is: 0.853415358507
total accuracy is: 0.7910479423
total accuracy is: 0.784896054306
total accuracy is: 0.801371800311
total accuracy is: 0.701456653939
total accuracy is: 0.830787724509
total accuracy is: 0.724225710649
total accuracy is: 0.797765521143
total accuracy is: 0.838141705558
total accuracy is: 0.829585631452
total accuracy is: 0.864304907368
total accuracy is: 0.883326262198
total accuracy is: 0.859142978362
total accuracy is: 0.814029133079

```
total accuracy is: 0.842313675576
total accuracy is: 0.831565549427
total accuracy is: 0.849101965776
total accuracy is: 0.853556781219
total accuracy is: 0.872719558761
total accuracy is: 0.725215669637
```

AE + BN, encoder activation: Relu



AE + BN, encoder activation: LeakyRelu



Summary: AE can help to understand the physical correspondence. And VAE can get better classification performance.