Autoencoder and Variational Autoencoder

Zhe Chen, Dijing Zhang

K-Means Clustering

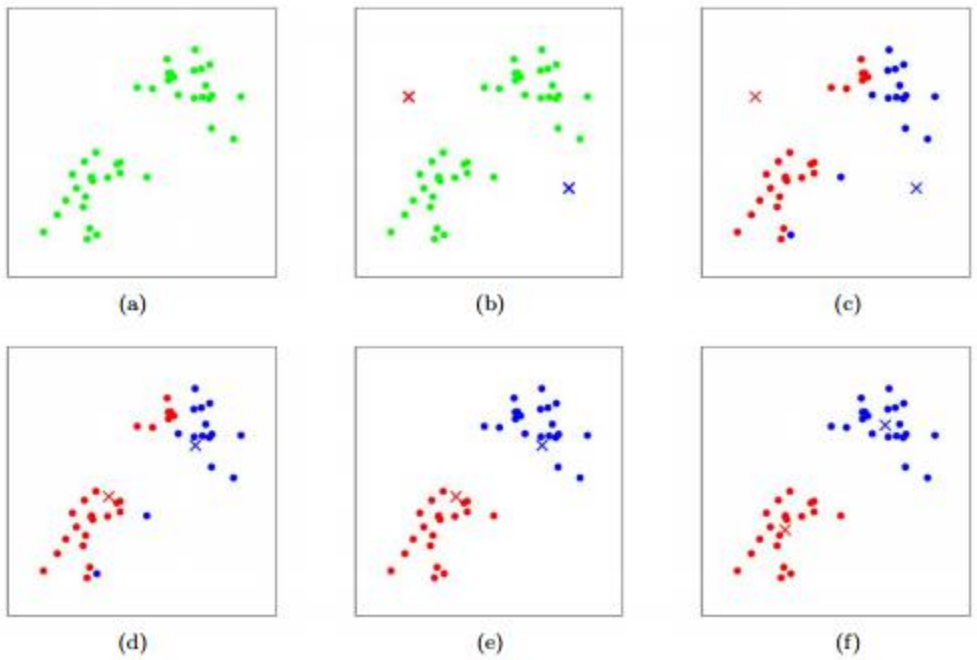


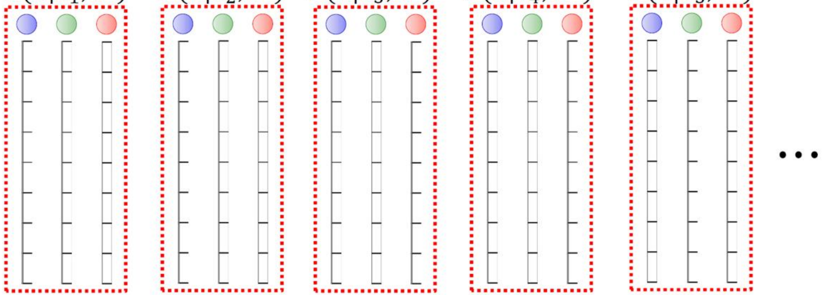
Figure 1: K-means algorithm. Training examples are shown as dots, and cluster centroids are shown as crosses. (a) Original dataset. (b) Random initial cluster centroids. (c-f) Illustration of running two iterations of k-means. In each iteration, we assign each training example to the closest cluster centroid (shown by "painting" the training examples the same color as the cluster centroid to which is assigned); then we move each cluster centroid to the mean of the points assigned to it. Images courtesy of Michael Jordan.

Reference: https://stanford.edu/~cpiech/cs221/handouts/kmeans.html

Expectation Maximization

EM for GMMs

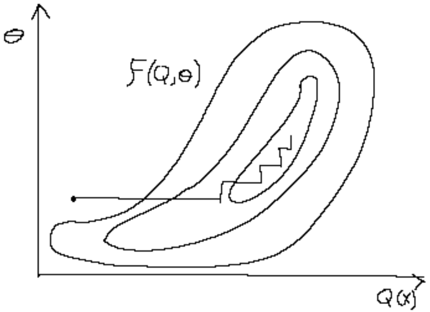
In proportion to $P(k|o_1;\theta^l)$ $P(k|o_2;\theta^l)$ $P(k|o_3;\theta^l)$ $P(k|o_4;\theta^k)$ $P(k|o_5;\theta^l)$



- Initialize μ_k^0 and Σ_k^0 for all k
- Iterate (over l):
  - Compute $P(k|o;\theta^l)$ for all o
    - Compute the proportions by which o is assigned to all Gaussians
  - Update:
  - $\mu_k^{l+1} = \frac{1}{\sum_o P(k|o;\theta^l)} \sum_o P(k|o;\theta^l) o$
  - $\Sigma_k^{l+1} = \frac{1}{\sum_o P(k|o;\theta^l)} \sum_o P(k|o;\theta^l)(o-\mu_k^{l+1})(o-\mu_k^{l+1})^T$

E: complete the data
M: update estimates

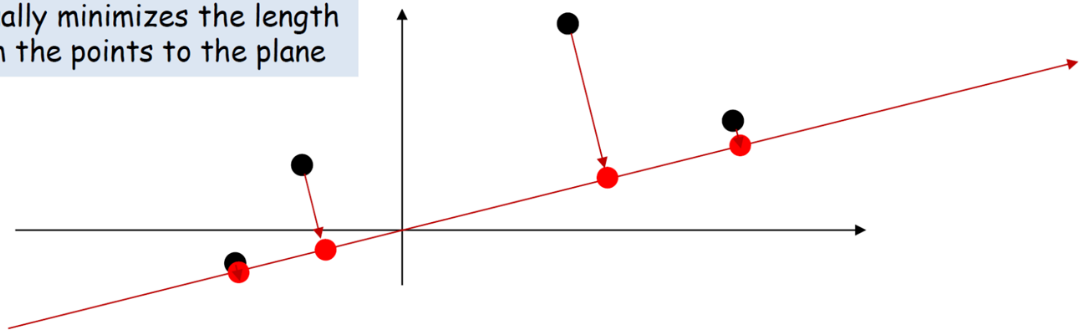
Coordinate ascent

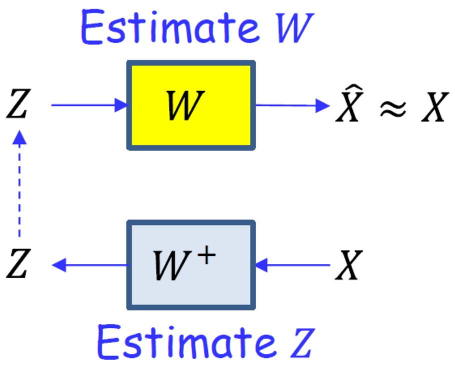


Reference: 10-701 EM Slides by Eric Xing

Linear Autoencoder = Iterative PCA

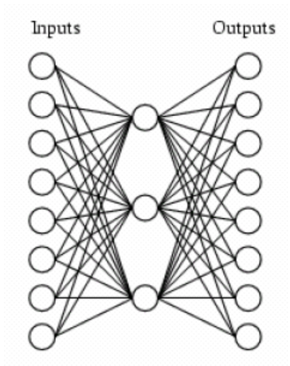
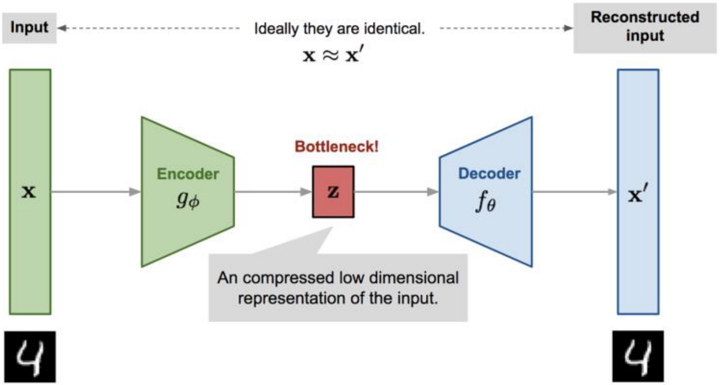
This individually minimizes the length of lines from the points to the plane





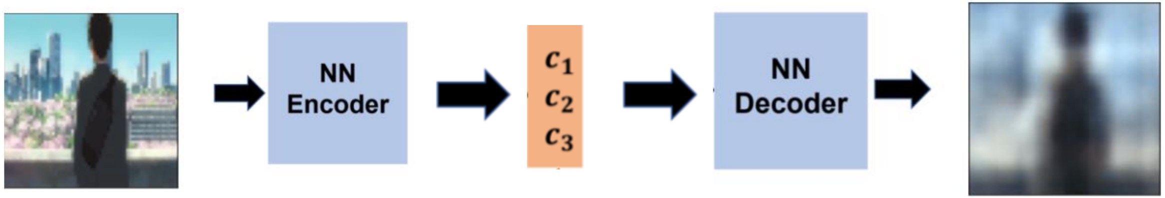
- Initialize a subspace (the basis w)
- Iterate until convergence:
  - Find the best position vectors Z on the W subspace for each training instance
    - Find the location on W that is *closest* to each instance, i.e. the perpendicular projection
  - Let W rotate and stretch/shrink, keeping the arrangement of Z locations fixed
    - Minimize the total square length of the lines attaching the projection on the place to the instance

Examples of Autoencoders

	Binary Autoencoder	MNIST Autoencoder
Network		
Data/Target	One-hot encodings of 0 to 7	MNIST dataset
	Can we get the network to learn a binary latent vector?	How do we improve the quality of reconstructions?

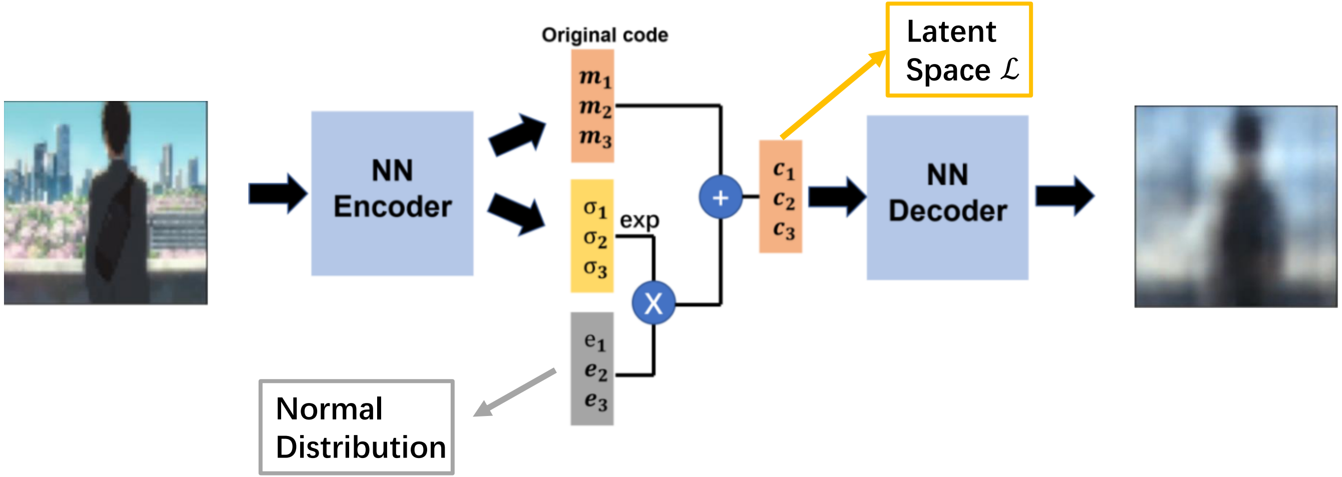
Reference: 10-701 NN Slides by Eric Xing, https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

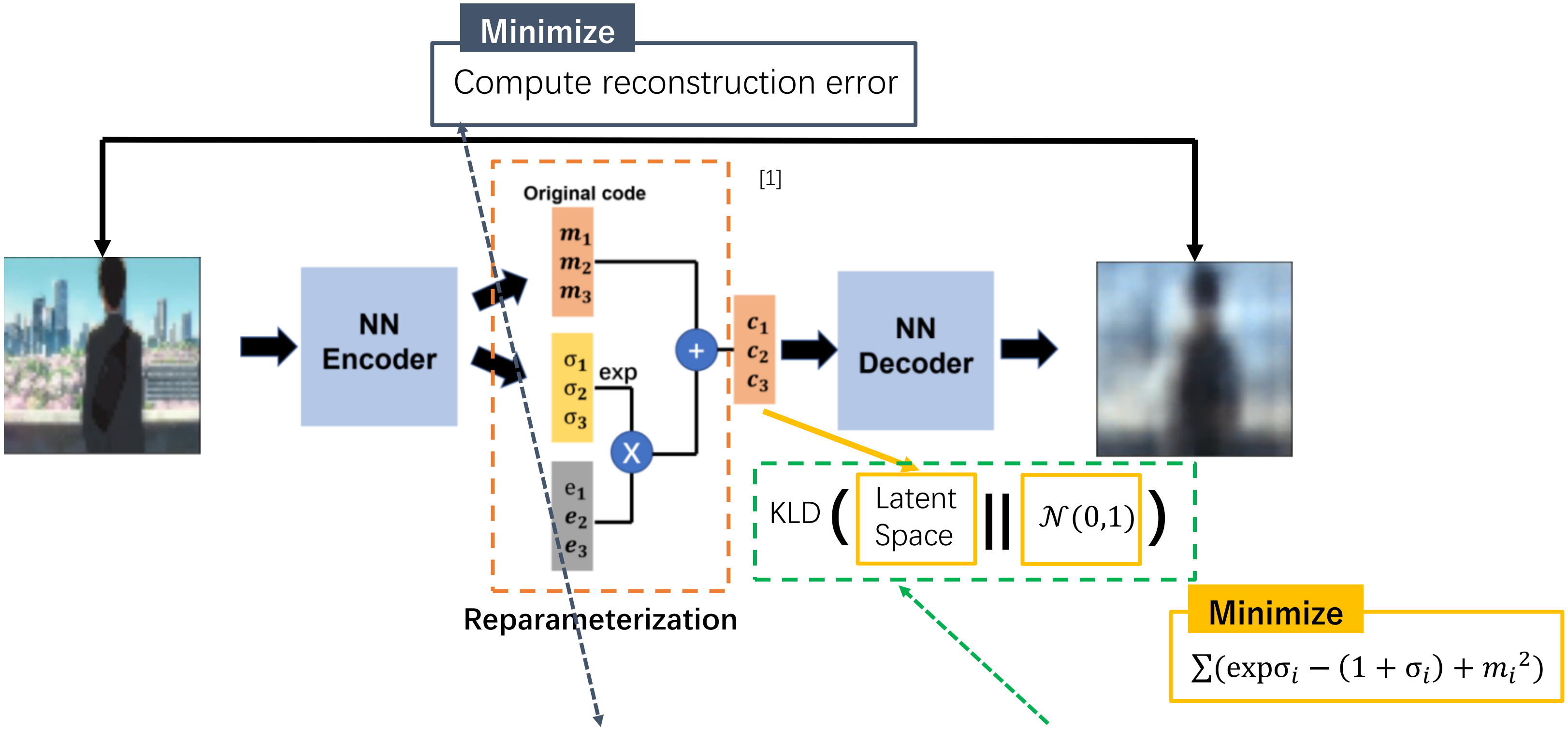
AE



Can hardly learn a continuous latent space if we directly learn the code (ci) here.
But how about we introduce the distribution idea to the latent space, like making the space follow one distribution, so that it can be continuous?

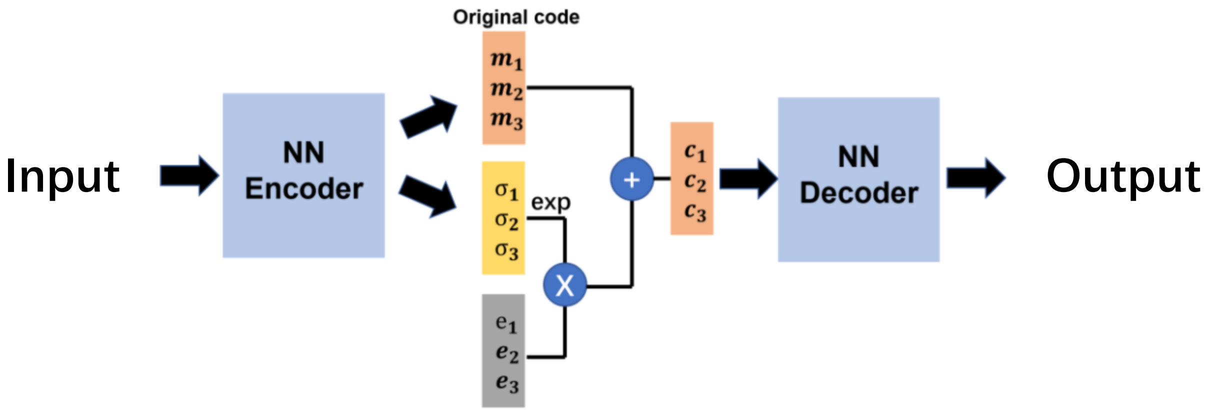
VAE





Loss Function: [2] $\mathcal{L}(\theta, \phi; x, z, \beta) = E_{q_\phi(z|x)}[\log p_\theta(x|z)] - D(q_\phi(z|x)||p(z))$

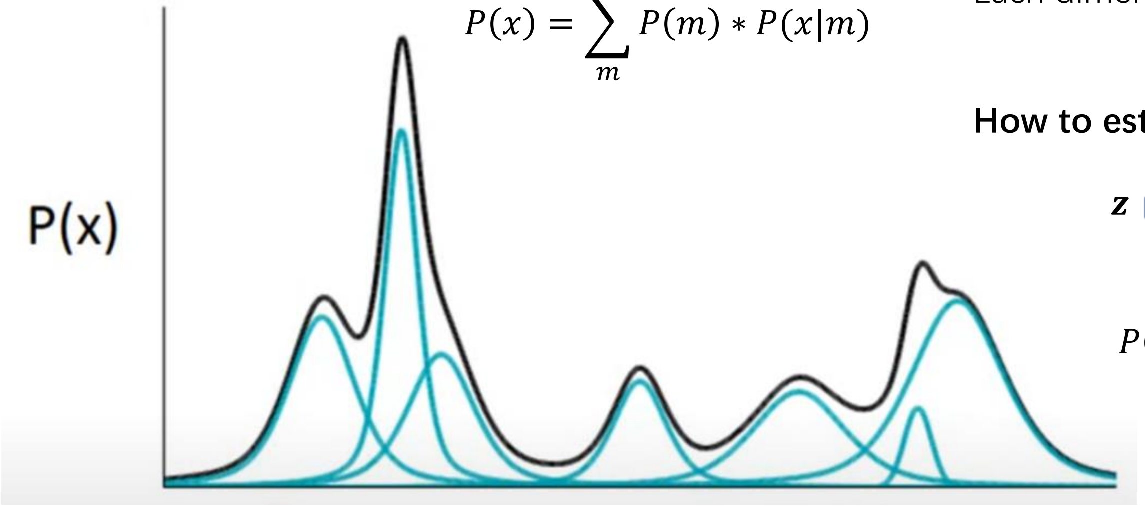
Reference:
[1] Hung-yi Lee's lecture https://www.youtube.com/watch?v=0CKeqXl5IY0&t=1650s
[2] Higgins, Irina, et al. "beta-vae: Learning basic visual concepts with a constrained variational framework." (2016).



Hope to learn the probability distribution of output

Gaussian Mixture Model

$$P(x) = \sum_m P(m) * P(x|m)$$



$z \sim N(0, I)$ z is a vector from normal distribution

Each dimension of Z represents an attribute

$$\boldsymbol{x}|\boldsymbol{z} \sim \boldsymbol{N}(\boldsymbol{\mu}(\boldsymbol{z}), \boldsymbol{\sigma}(\boldsymbol{z}))$$

How to estimate $\boldsymbol{\mu}(\boldsymbol{z}), \boldsymbol{\sigma}(\boldsymbol{z})$?

$$P(x) = \int_z P(z) * P(x|z)dz$$

$$P(x) = \int_z P(z) * P(x|z)dz$$

Tune the NN parameters to max likelihood of the observed x: $L = \sum_x logP(x)$

Decoder

Another distribution $q(z|x)$:

$$\boldsymbol{z|x \sim N(\mu'(x), \sigma'(x))}$$

Encoder

$$logP(x) = \int_z q(z|x)logP(x)dz \quad s.t. \int_z q(z|x)dz = 1 \ \ and\ q\ can\ be\ any\ distribution$$

$$= \int_z q(z|x)log\frac{P(z,x)}{P(z|x)}dz = \int_z q(z|x)\log(\frac{P(z,x)}{q(z|x)} * \frac{q(z|x)}{P(z|x)})dz$$

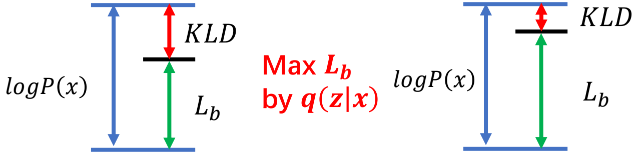
$$= \int_z q(z|x)\log\frac{P(z,x)}{q(z|x)}dz + \boxed{\int_z q(z|x)\log\frac{q(z|x)}{P(z|x)}dz}$$

$$* KLD(p||q) = \sum p * log\frac{p}{q}$$

$$KLD(q(z|x)||P(z|x)) \geq 0$$

$$\geq \int_z q(z|x)\log\frac{P(z,x)}{q(z|x)}dz$$ **Evidence Lower Bound(ELBO)** L_b

$$logP(x) = L_b + KLD(q(z|x)||P(z|x))$$



Find $q(z|x)$ and $P(x|z)$ to maxL_b

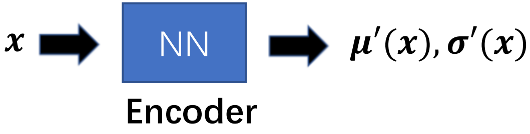
~N(0,I)

$$L_b = \int_z q(z|x) \log \frac{P(z,x)}{q(z|x)} dz = \int_z q(z|x) \log \frac{P(x|z) * P(z)}{q(z|x)} dz$$

$$= \int_z q(z|x) \log \frac{P(z)}{q(z|x)} dz + \int_z q(z|x) \log P(x|z)\, dz$$

$$= -KLD(q(z|x)||P(z)) + \int_z q(z|x) \log P(x|z)\, dz$$

$$z|x \sim N(\mu'(x), \sigma'(x))$$



1. Minimize $KLD(q(z|x)||P(z))$:

Minimize

$$\sum(\exp\sigma_i - (1 + \sigma_i) + m_i^2)$$

2. Maximize $\int_z q(z|x) \log P(x|z)\, dz$

$$= E_{q(z|x)}[logP(x|z)]$$

Minimize

Compute reconstruction error