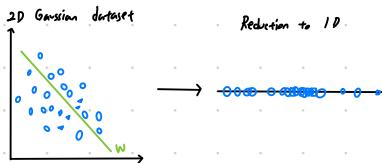


AutoEncoder



Recap - Linear Dimensionality Reduction

Ex)



What would be a good reduction?

① Find w s.t. it maximizes the variance of the projected data

② Find w s.t. it minimizes the reconstruction error.

① = ② : same

① Find w s.t. it maximizes the variance of the projected data

$$\begin{aligned} \text{Var} &= \frac{1}{N} \sum_{i=1}^N (w^T (x^{(i)} - \bar{x}))^2 = \frac{1}{N} \sum_{i=1}^N w^T (x^{(i)} - \bar{x})(x^{(i)} - \bar{x}) w \\ &= w^T \underbrace{\frac{1}{N} \sum_{i=1}^N (x^{(i)} - \bar{x})(x^{(i)} - \bar{x})}_S w = w^T S w \end{aligned}$$

Hence, maximizing the variance can be written as

$$\max_w w^T S w$$

s.t. $w^T w = 1$ by applying Lagrangian function,

we get

$$L(w, \alpha) = w^T S w + \alpha(1 - w^T w) = w^T S w + \alpha - \alpha w^T w$$

We need to find $\frac{\partial L}{\partial w} = 0$ to get optimal w

$$\frac{\partial L}{\partial w} = 2S w - 2\alpha w$$

$$2S w - 2\alpha w = 0$$

$$S w = \alpha w$$

We derived the equation for getting eigenvector & eigenvalue.

★ Variance is a measure of the variability of the data. Therefore, if you were to select a single principal component, you would want it to account for the most variability possible.

How to project a vector x onto the subspace spanned by the unit vector w ?

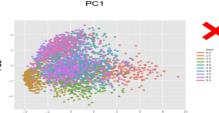
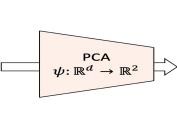
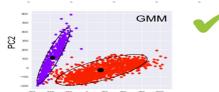
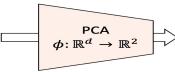
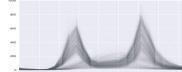
$$\hat{x} = w w^T x$$

Therefore, the reconstruction error:

$$\frac{1}{N} \sum_{i=1}^N \| \hat{x}^{(i)} - x^{(i)} \|^2 = \frac{1}{N} \sum_{i=1}^N \| w w^T x^{(i)} - x^{(i)} \|^2$$

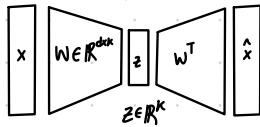
However, high-dimensional data often lies on non-linear manifolds that cannot be captured by linear models such as PCA ($w w^T x^{(i)}$ is linear)

Fremont Bridge Hourly Bicycle Counts – Seattle



PCA vs Auto-Encoder (AE)

PCA



PCA: Linear dimensionality reduction

- Forward transform: $z = Wx$

- Inverse transform: $\hat{x} = Wz$

$$\text{Training loss: } \min_w \mathbb{E}_x [\|x - \hat{x}\|^2] = \mathbb{E}_x [\|x - Ww^T x\|^2]$$

$$\text{s.t. } W^T W = I_{K \times K}$$

AE: Nonlinear dimensionality reduction

- Forward transform: $z = \phi(x)$

- Inverse transform: $\hat{x} = \psi(z) = \psi(\phi(x))$

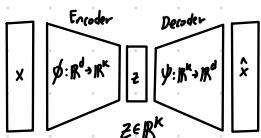
$$\text{Training loss: } \min_{\phi, \psi} \mathbb{E}_x [\|x - \hat{x}\|^2] = \mathbb{E}_x [\|x - \psi(\phi(x))\|^2]$$

Auto Encoder

- Autoencoding is a form of compression. Smaller latent space will force a longer training bottleneck.

- Hidden layer forces network to learn a compressed latent representation

- Reconstruction loss forces latent representation to capture as much information about the data as possible.



Need to estimate the latent distribution post-hoc.

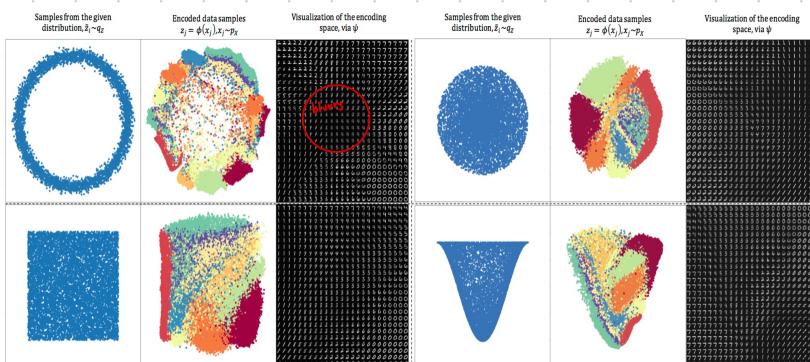
How can we avoid post-hoc Density Estimation?

① VAE

② Sliced Wasserstein Auto-Encoders

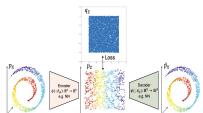
etc

What does it mean by avoiding post-hoc density estimation?



: By comparing p_x to q_x , we can model our model to produce distribution as we want, e.g. Gaussian.

Auto-Encoders with Priors on Latent Space



$$\text{Loss: } \min_{\theta_0, \theta_1} \mathbb{E}_{x \sim p_x} [\|x - \hat{x}\|^2] + \lambda D(p_z, q_z), \text{ where } D \text{ stands for dissimilarity measure.}$$

A very common dissimilarity measure

① KL-Divergence:

$$D_{KL}(P||Q) = \mathbb{E}_{x \sim P} [\log \frac{P(x)}{Q(x)}] = \int P(x) \log \frac{P(x)}{Q(x)} dx$$

Problem w/ KL

- Not symmetric

- Not defined when we have non-overlapping support, i.e., $Q(x)=0$

② Jensen-Shannon Divergence (JSD)

$$JS(P||Q) = \frac{1}{2} \left(D_{KL}(P || \frac{P+Q}{2}) + D_{KL}(Q || \frac{P+Q}{2}) \right)$$

③ L_p -metrics ($p \geq 1$)

$$= \left(\int_x |P(x) - Q(x)|^p dx \right)^{\frac{1}{p}}$$

$$\text{e.g. } L_1 \text{ metric: } \int_x |P(x) - Q(x)| dx$$

$$L_2 \text{ metric: } \left(\int_x |P(x) - Q(x)|^2 dx \right)^{\frac{1}{2}}$$

④ Symmetric chi-squared distances:

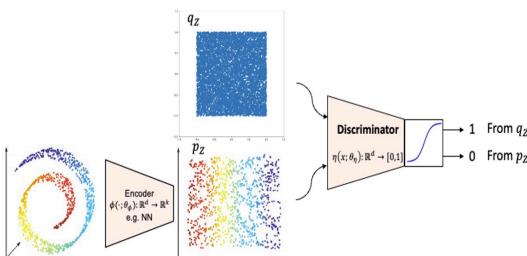
$$\int_x \frac{2(P(x) - Q(x))^2}{Q(x) + P(x)} dx$$

⑤ Hellinger distances:

$$\left(\frac{1}{2} \int_x (\sqrt{P(x)} - \sqrt{Q(x)})^2 dx \right)^{\frac{1}{2}}$$

⑥ Wasserstein distance

Adversarial Networks Allow us to measure Jensen-Shannon Divergence



We don't have access to p_z , but we only have samples from it.

Dissimilarity between p_z and q_z :

$$D(p_z, q_z) = \max_{\eta} \mathbb{E}_{z \sim q_z} [\log(\eta(z))] + \mathbb{E}_{z' \sim p_z} [\log(1 - \eta(z'))]$$

We want $\eta(z) = 1, \eta(z') = 0$, where $z \sim q_z, z' \sim p_z$, hence $\log(\eta(z)) = \log(1) / \log(1 - \eta(z')) = \log(1)$

$$\text{Loss: } \min_{\theta_0, \theta_1} \max_{\eta} \mathbb{E}_{x \sim p_x} [\|x - \psi(\phi(x))\|^2 + \log(1 - \eta(\phi(x)))] + \mathbb{E}_{z \sim q_z} [\log(\eta(z))]$$

Why Adversarial Networks allow us to measure JSO?

$$\max_{\eta} \mathbb{E}_{z \sim q_z} [\log(\eta(z))] + \mathbb{E}_{x \sim p_x} [\log(1 - \eta(\phi(x)))] = \mathbb{E}_{z \sim q_z} [\log(\eta(z))] + \mathbb{E}_{z' \sim p_z} [\log(1 - \eta(z'))]$$

$$= \int_z q_z(z) \log(\eta(z)) dz + \int_z p_z(z) \log(1 - \eta(z)) dz$$

Then, we can find η^* when $\frac{\partial J}{\partial \eta} = 0$

$$\frac{\partial J}{\partial \eta} = \int_{\mathcal{Z}} \frac{q_z(z)}{\eta(z)} dz - \int_{\mathcal{Z}} \frac{p_z(z)}{1-\eta(z)} dz$$

$$\int_{\mathcal{Z}} \frac{q_z(z)}{\eta(z)} dz - \int_{\mathcal{Z}} \frac{p_z(z)}{1-\eta(z)} dz = 0$$

$$\int_{\mathcal{Z}} \frac{q_z(z)}{\eta(z)} - \frac{p_z(z)}{1-\eta(z)} dz = 0$$

It must holds for all z , so

$$\frac{q}{\eta} - \frac{p}{1-\eta} = 0$$

$$\Rightarrow \frac{q}{\eta} = \frac{p}{1-\eta} \Rightarrow q(1-\eta) = \eta p$$

$$\Rightarrow q - q\eta = \eta p$$

$$\Rightarrow q = \eta p + \eta q$$

$$\Rightarrow q = \eta(p+q)$$

$$\Rightarrow \eta = \frac{q}{p+q}$$

$$\eta^* = \frac{q_z}{q_z + p_z}$$

Therefore,

$$\begin{aligned} \max_{\eta} \mathbb{E}_{z \sim q_z} [\log(\eta(z))] + \mathbb{E}_{x \sim p_x} [\log(1-\eta(x))] &= \int_{\mathcal{Z}} q_z(z) \log\left(\frac{q_z}{q_z + p_z}\right) dz + \int_{\mathcal{Z}} p_z(z) \log\left(1 - \frac{q_z}{q_z + p_z}\right) dz \\ &= \int_{\mathcal{Z}} q_z(z) \log\left(\frac{q_z}{q_z + p_z}\right) dz + \int_{\mathcal{Z}} p_z(z) \log\left(\frac{q_z}{q_z + p_z}\right) dz, \quad \text{since } \log(1-a) = \log(1) + \log(a) \text{ and } \log(1)=0 \\ &= \int_{\mathcal{Z}} q_z(z) \log\left(\frac{q_z}{\frac{q_z + p_z}{2}}\right) dz + \int_{\mathcal{Z}} p_z(z) \log\left(\frac{q_z}{\frac{q_z + p_z}{2}}\right) dz \\ &= \int_{\mathcal{Z}} q_z(z) \left(\log\left(\frac{1}{2}\right) - \log\left(\frac{q_z}{\frac{q_z + p_z}{2}}\right) \right) dz + \int_{\mathcal{Z}} p_z(z) \log\left(\frac{1}{2} - \log\left(\frac{q_z}{\frac{q_z + p_z}{2}}\right)\right) dz \\ &= \int_{\mathcal{Z}} q_z(z) \left(\log\left(\frac{1}{2}\right) + \log\left(\frac{q_z}{\frac{q_z + p_z}{2}}\right) \right) dz + \int_{\mathcal{Z}} p_z(z) \left(\log\left(\frac{1}{2}\right) + \log\left(\frac{q_z}{\frac{q_z + p_z}{2}}\right) \right) dz \\ &= \int_{\mathcal{Z}} q_z(z) \log\left(\frac{1}{2}\right) dz + \int_{\mathcal{Z}} q_z(z) \log\left(\frac{q_z}{\frac{q_z + p_z}{2}}\right) dz + \int_{\mathcal{Z}} p_z(z) \log\left(\frac{1}{2}\right) dz + \int_{\mathcal{Z}} p_z(z) \log\left(\frac{q_z}{\frac{q_z + p_z}{2}}\right) dz \end{aligned}$$

Since $q_z(z)$ is 1df, $\int_{\mathcal{Z}} q_z(z) dz$ and $\int_{\mathcal{Z}} p_z(z) dz = 1$.

Therefore, $\int_{\mathcal{Z}} q_z(z) \log\left(\frac{1}{2}\right) dz = -\log(2)$

$$\int_{\mathcal{Z}} p_z(z) \log\left(\frac{1}{2}\right) dz = -\log(2)$$

$$= \int_{\mathcal{Z}} q_z(z) \log\left(\frac{q_z}{\frac{q_z + p_z}{2}}\right) dz + \int_{\mathcal{Z}} p_z(z) \log\left(\frac{q_z}{\frac{q_z + p_z}{2}}\right) dz - 2\log(2) = 2JS(q_z, p_z) - 2\log(2)$$

For the optimal discriminator, $\eta^* = \arg\min_{\theta, \gamma_0} \mathbb{E}_{x \sim p_x} [||x - \hat{x}||^2] + \lambda \cdot JS(p_z, q_z)$

Then, what are the benefits of using prior distribution?

① Regularization: By imposing a prior distribution on the latent space, such as Gaussian distribution, it helps to prevent overfitting and encourages the model to learn a more meaningful representation of the input data.

② Improved Latent space: The prior distribution guides the learning process, encouraging AE to map input data into a latent space that follows the specified distribution. It can lead to more structured and interpretable latent space representation.

③ Enhance Generative Modelling

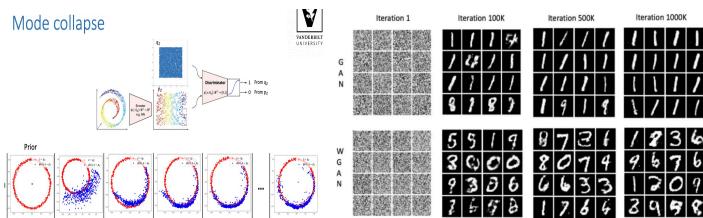
④ Interpretability and Controllability: w/ a prior distribution, it becomes easier to interpret and control the latent space. For example, in a Gaussian prior, each dimension of the latent space corresponds to a specific feature, allowing for more meaningful manipulation of generated samples.

What's the problem of using JS Divergence?

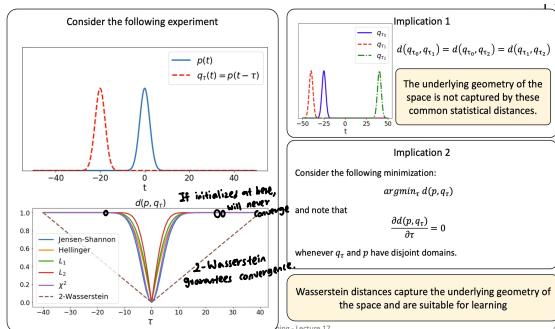
: Mode Collapse.

Mode Collapse happens when the generator focuses on producing a limited set of data patterns that deceive the discriminator. It becomes fixated on a few dominant modes in the training data and fails to capture the full diversity of the data distribution.

Mode collapse



How to avoid Mode Collapse? : Use Wasserstein distance



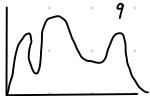
Wasserstein distance: Also known as Earth Mover's distance, is a measure of dissimilarity between two probability distributions. Given two distributions U and V on metric space (X, d) and (Y, d) , the Wasserstein distance $W_p(U, V)$ is defined as the minimum cost of transporting mass from V to U , where the cost is measured as the p th power of the distance function d .

$$W_p^p(U, V) \stackrel{p \geq 1}{=} \left(\min_{\pi} \int_{X \times Y} d^p(x, y) d\pi(x, y) \right)^{\frac{1}{p}}$$

$$\text{s.t. } \nu(A, Y) = \nu(A), \nu(X, B) = \nu(B), \forall A \subseteq X, \forall B \subseteq Y$$

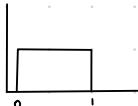
Before dive deep into Wasserstein distance, how to sample data from an unknown probability density function?

let's say there's q , unknown pdf

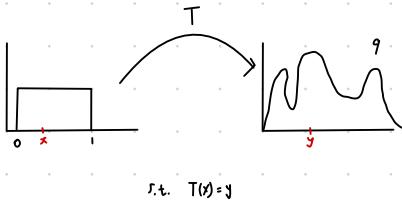


how to sample data from q ?

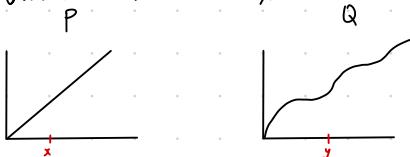
① First, generate a uniform distribution, p



② let's say there's T that map $x \rightarrow y$.



③ let's make CDF at p and q , then



We need mapping x to y s.t. $P(x) = Q(y)$

Since $T(x) = y$, $P(x) = Q(T(x))$

$$Q^{-1}(P(x)) = T(x)$$

Since p is uniform distribution, $P(x) = x$, hence

$$Q^{-1}(x) = T(x)$$

And T is optimal transport map.

Now, let there two unknown p, q



$$\begin{aligned} \text{Expected transport cost} &: \int \|T(x) - x\|^2 p(x) dx \\ &= \int \|Q^{-1}(P(x)) - x\|^2 p(x) dx \end{aligned}$$

Let $P(x) = u$, then $x = P^{-1}(u)$, and $p(x)dx = du$

$$= \int \|Q^{-1}(u) - P^{-1}(u)\|^2 du. \quad \text{It's the general idea of Wasserstein distance.}$$

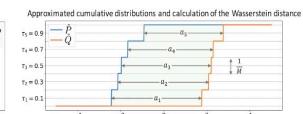
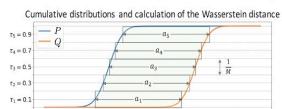
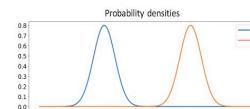
W 2-Wasserstein

$$\text{Loss: } \min_{\rho, \gamma_{\theta}} \mathbb{E}_{x \sim p_x} [\|x - \hat{x}\|^p] + \lambda W_2^p(p, q)$$

where $W_2^p(p, q) = \int_0^1 \|P^{-1}(u) - Q^{-1}(u)\|^p du$

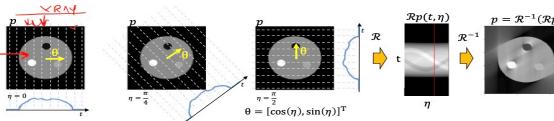
and $P(t) = \int_{-\infty}^t P(x) dx = F([- \infty, t])$ where now is pdf, and $P(t)$ is cdf

$$Q(t) = \int_{-\infty}^t q(u) du = F([- \infty, t])$$



Slicing high-dimensional distribution

$$R_p(t, \theta) = \int_X f(t-x, \theta) p(x) dx, \quad t \in \mathbb{R}, \quad \theta \in S^{d-1}$$



And, sliced Wasserstein distance:

$$\int W_p(u, v) = \int_{S^{d-1}} W_p^p(R_p(\cdot, \theta), R_q(\cdot, \theta)) d\theta$$

Monte-Carlo Integration:

$$\int W_{p, L}(u, v) = \left(\frac{1}{L} \sum_{\ell=1}^L W_p^p(R_p(\cdot, \theta_\ell), R_q(\cdot, \theta_\ell)) \right)^{\frac{1}{p}}$$

Sliced Wasserstein Auto-Encoders

Algorithm 1 Sliced-Wasserstein Auto-Encoder (SWAE)

Require: Regularization coefficient λ , and number of random projections, L .

Initialize the parameters of the encoder, ϕ , and decoder, ψ

while ϕ and ψ have not converged do

Sample $\{x_1, \dots, x_M\}$ from training set (i.e. p_X)

Sample $\{\tilde{z}_1, \dots, \tilde{z}_M\}$ from q_Z^{K-1}

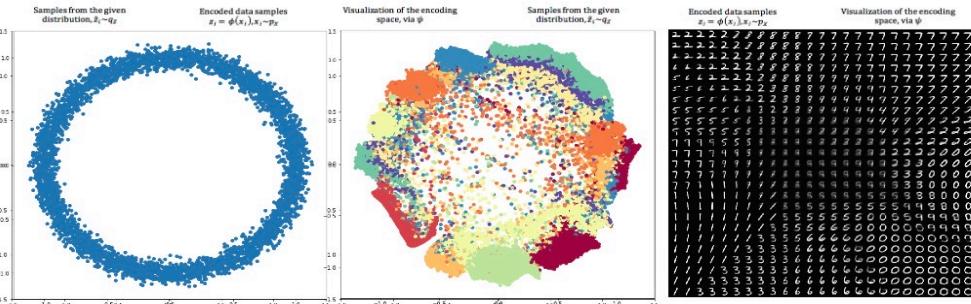
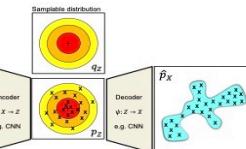
Sample $\{\theta_1, \dots, \theta_L\}$ from S^{d-1}

Sort θ_1, \tilde{z}_M such that $\theta_1 \cdot \tilde{z}_{i[m]} \leq \theta_1 \cdot \tilde{z}_{i[m+1]}$

Sort $\theta_1 \cdot \phi(x_m)$ such that $\theta_1 \cdot \phi(x_{j[m]}) \leq \theta_1 \cdot \phi(x_{j[m+1]})$

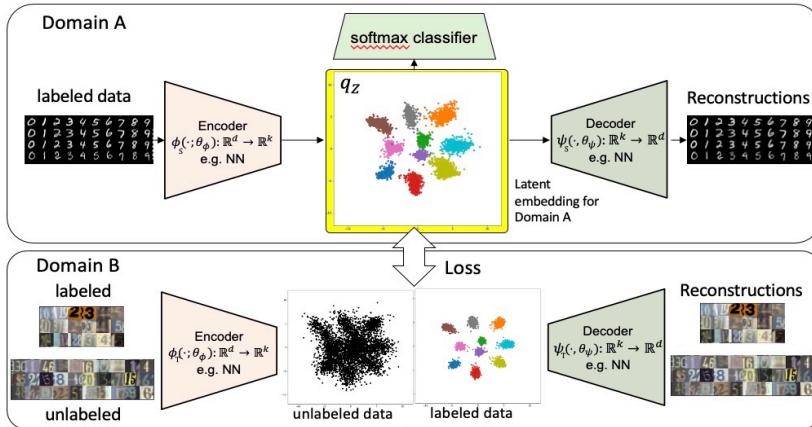
Update ϕ and ψ by descending: $\sum_{m=1}^M c(x_m, \psi(\phi(x_m))) + \lambda \sum_{l=1}^L \sum_{m=1}^M c(\theta_l \cdot \tilde{z}_{i[m]}, \theta_l \cdot \phi(x_{j[m]}))$

end while



SWAE for Domain Adaptation

Main idea: Set the prior distribution, q_z , for Domain B to be the latent distribution of Domain A.



3/22/24

42

SWAE for domain adaptation

