Wasserstein Autoencoders

1. Latent Variable Models

Consider a latent variable model where we (1) sample $Z \sim P_Z$ from Z (2) map Z to X with 'random transformation' $P_G(X|Z)$,

$$p_G(X) := \int_Z p_G(x|z)p(z)dz$$

A deterministic transform is a function that maps a point in the latent space to a point in the feature space. A random transform maps the latent code to distribution in the feature space.

2. Optimal Transport: Introduction

Optimal transport gives a framework for comparing measures μ and η. One pays a cost for transporting one measure to another. Consider the first measure μ as a pile of sand and the second measure η as a hole we wish to fill up. We assume that both measures are probability measures on spaces X and Y respectively. Let $c : X \times Y \to [0, \infty]$ be a cost function where $c(x, y)$ measures the cost of transporting one unit of mass from $x \in X$ to $y \in Y$. The optimal transport problem is how to transport μ to η while minimizing the cost c. We say that $T : X \to Y$ transports $\mu \in P(X)$ to $\eta \in P(Y)$ and we call T a transport map if $\eta(B) = \mu(T^{-1}(B))$ for all η-measurable sets B. You can formulate transport problems and prove that there are minimizers.

3. Optimal Transport: Application

Idea: Move from f-divergences (ex. KL, JS) to optimal transport distances, in particular the Wasserstein distance.

Take the cost function/distance defined between two points in the space and leverage it to the level of distributions over the space.

$$W_c(P_X, P_G) = \inf_{\Gamma \in P(X \sim P_X, Y \sim P_G)} \mathbb{E}_{(X,Y)\sim\Gamma}\Big[c(X, Y)\Big]$$

The minimal expected cost between two input points jointly distributed according to Γ, a coupling (forms joint distribution from the marginals). The Γ is constrained to have particular marginals.

$$\Gamma(X, Y) = P_X(X) \times \Gamma(Y|X) = P_G(Y) \times \Gamma(X|Y)$$

We only tune the conditional component of the joint – the conditional does the 'coupling', linking two variables. If we let $c(x, y) = d^p(x, y)$ then we get a Wasserstein distance.

When P_G is a latent variable model, this can be rewritten where the constrained optimization with respect to joint distribution is replaced with a constrained optimization with respect to probabilistic encoders. The constraint is that we ask the marginal to match the prior. We encode and average the encodings of our samples, then we match this average to the prior.

This leads us to the main theorem of the paper. Assume $P_G(X|Z) = \delta_{G(Z)}$ for any $G : Z \to X$. Then,

$$W_c(P_X, P_G) = \inf_{Q: Q_Z = P_Z} \mathbb{E}_{P(X)}\mathbb{E}_{Q(Z|X)}\Big[c(X, G(Z))\Big]$$

where $Q_Z(Z) := \mathbb{E}_{P_X}[Q(Z|X)]$ is the aggregated posterior (the constrained marginal).

This constraint that $Q : Q_Z = P_Z$ is hard to satisfy so we relax it by penalizing the distance between distributions,

$$W_c = \inf_{P_G} \inf_{Q(Z|X)\in Q} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}\Big[c(X, G(Z))\Big] + \lambda * D_Z(Q_Z, P_Z)$$

where D_Z is a divergence measure.

Let's make a comparison between VAE and WAE objectives,

$$L_{VAE} = \inf_{P_G} \inf_{Q \in Q} \mathbb{E}_{P_X} \mathbb{E}_{Q(Z|X)}[-\log p_G(X|Z)] + \mathbb{E}_{P_X}[KL(Q(Z|X), P_Z)]$$

$$L_{WAE} = \inf_{P_G} \inf_{Q \in Q} \mathbb{E}_{P_X} \mathbb{E}_{Q(Z|X)}[c(X, G(Z))] + \lambda * D_Z(\mathbb{E}_{P_X}[Q(Z|X)], P_Z)$$

If you take VAEs with Gaussian encoders and Wasserstein encoders with squared pixel loss then the reconstruction terms correspond. This brings us to the core difference between the methods. The expectation moves from outside to inside the KL divergence.

The picture with red and green circles: The regularizer of VAE asks every encoded distribution to match the prior. Every red ball must match the prior. This causes them to intersect which is a problem.

The regularizer of WAE asks the average of every encoded distribution to match the prior. The mixture (green) must match the prior.

WAE can be used with (1) any encoders/decoders, (2) any prior distribution that lets you estimate penalty (3) any reconstruction lost function (since vae reconstruction is a density you need to normalize it so you might need the partition function).

4. WAEGAN

Choose $D_Z(Q_Z, P_Z) = D_{JS}(Q_Z, P_Z)$ and use adversarial training,

$$D_{JS}(Q_Z, P_Z) \approx \sup_D \mathbb{E}_{Z \sim P_Z}[\log D(Z)] + \mathbb{E}_{Z* \sim Q_Z}[\log(1 - D(Z*))]$$

But in this case the discriminator acts in the *latent space* and tries to seaparate points sampled from P_Z from those samples from Q_Z. This should be easier because the discriminator is matching distributions in a much lower dimensional space than in normal GANs.

5. WAEMMD

For any p.d. reproducing kernel $k : Z \times Z \to \mathbb{R}$ and its RKHS H_k,

$$MMD_k(P_Z, Q_Z) = || \int_Z k(z, \cdot) dP_Z(z) - \int_Z k(z, \cdot) dQ_Z(z) ||_{H_k}$$

For any characteristic kernel (gaussian, inverse-multiquadratic, Matern,...) there is a u-statistic estimator of this MMD distance that can be combined with stochastic gradient descent.