Disconnected Manifold Learning for GANs

Mahyar Khayatkhoei*, Ahmed Elgammal*, Maneesh Singh**
*Department of Computer Science, Rutgers-New Brunswick I **Verisk Analytics

Introduction

- Real world data may lie on a “Disconnected Manifold”: can we transform a bird picture into a cat picture without ever generating an image that is not neither bird nor cat?
- Assumption: real data distribution is supported on a union of disjoint globally connected manifolds (each denoted a submanifold)
- Can GANs generate distributions supported on disconnected manifolds?

Problem Statement

- Generative Adversarial Networks (GANs): two player game
  - Generator: \(G; \theta \); \(Z \rightarrow X \) with \( Z \sim N(0, I) \)
  - Discriminator/Critic: \(D(x); \) \(R \rightarrow \)
- WGAN variant objective: \( \min_{G} \max_{D} \mathbb{E}_{x \sim P_{data}}[\log D(x)] - \mathbb{E}_{z \sim P_{z}}[\log (1 - D(G(z)))] \)
- 1-Lipschitzness: Gradient Penalty
- Convergence Assumption: G, D have unlimited capacity
- Neural networks are used to model G and D
- In practice, regularizations and optimizers explicitly limit NNs to learning continuous functions in order to ensure stability and generalization
- G cannot generate a disconnected support: \(z\) is supported on a connected manifold
- \(G\) is a continuous function
- Thus: \(G(z)\) is also supported on a connected manifold

DMWGAN Model

- A set of generators \(\{G(z; \theta_i)\}_{i \geq 1}\) together can model a discontinuous function
- We encourage different \(\theta_i\) to learn different submanifolds of real data (due to the non-convexity of optimization)
- Otherwise same difficulties as single G may occur
- Maximize mutual information \(I[\theta_i \rightarrow \hat{\theta_i}]\) between generator id (c) and generated sample (x)
- \(I(c; x) \geq H(c) + \mathbb{E}_{p(c=x|y)P(x|c)} \log \mathbb{E}_{p(c=x|y)} \frac{P(c=x|y)}{P(c=x)}\)
- \(y\) is induced by neural net \(G(z; \theta_i)\)
- \(y\) is modeled by neural net \(Q(x; y): X \rightarrow \{0, 1\}\)
- \(p(c)\) is the prior over generators (assumed uniform)

Objectives:

\[
\begin{align*}
V_G &= \mathbb{E}_{p(c=x|y)P(x|c)} \log q(c|x) \\
V_D &= \mathbb{E}_{p(x|c)D(x|w)} - \mathbb{E}_{p(c=x|y)P(x|c)}[D(x|w)] \\
V_Q &= \mathbb{E}_{p(c=x|y)P(x|c)}[D(x|w) + \lambda V_0] \\
\end{align*}
\]

Consequences

- Sample Quality: \(G\) learns a cover of the real manifold
- Thus: \(G\) generates off real manifold samples
- Mode Dropping: a trade-off for \(G\) in learning the support
- Covering all submanifolds of the real manifold
- Minimizing the volume of off manifold space
- Thus: \(G\) may drop certain submanifolds completely or partially in favor of less off manifold space

Local Convergence: The training of GANs is locally convergent when generated and real data distributions are equal near the equilibrium point \([1]\).

References


Prior Learning

- The true number of submanifolds \(N_{\theta}\) and the prior over them are not known in practice
- If \(N_{\theta} < N_f\): one generator covers multiple submanifolds
- If \(N_{\theta} > N_f\): multiple generators share one submanifold

- We assume \(N_{\theta} = N_f\):
- Solving this case reduces the first case into a trade-off: the more generators in a model, the better the support
- An EM approach:
  - E step: \(p(c) = \mathbb{E}_{p(x|c)}[\delta(c|x)]\)
  - M step: \(p(c)\) to train WGAN
- We use \(v(c), \xi\) to Softmax\(\{v(c)\}\) to approximate \(p(c)\)
- \(r(c)\) trained by minimizing cross entropy, \(H(p(c), r(c))\)
- We add entropy penalty and decay its weight with time to gradually shift \(r(c)\) away from uniform
- \(L_{\text{prior}} = \mathbb{E}_{p(x|c)}[H(\xi(q(c|x), r(c))] = a^xH(r(c))\)

Experiments

- FaceBed dataset: LSUN bedrooms (20K) + CelebA (20K)
- MNIST dataset
- CelebaA
- JSD and FID scores
- Prior Learning: \(r(c; \xi)\) during training of DMWGAN-PL

References