Generative Adversarial Networks, Wasserstein Distance, and Adversarial Loss

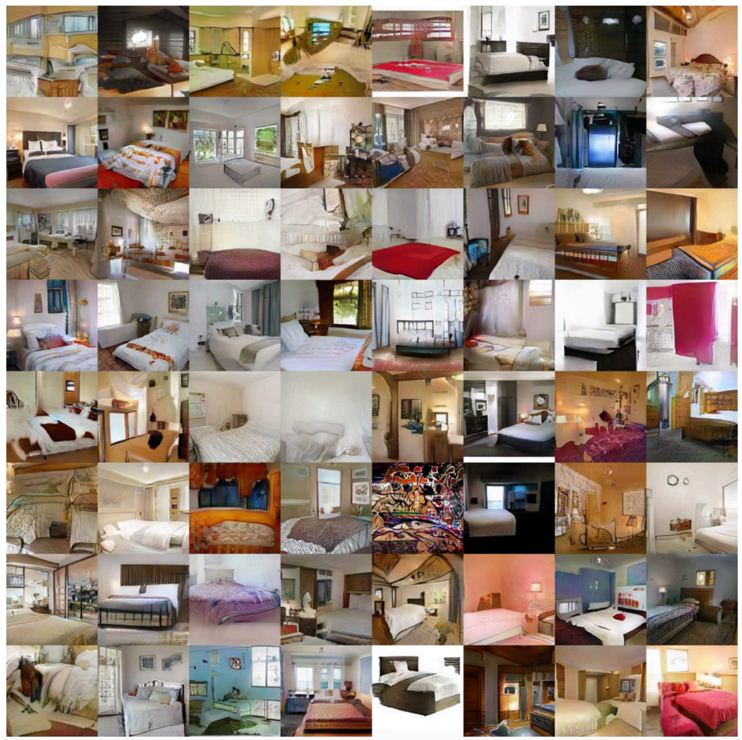
Zhiyu Min
Alibaba AliMe X-Lab

Outline

- GAN
  - Definition and formulation
  - Saddle point optimization
  - Vanishing gradient
  - Alternative objective for Generator
- Wasserstein Distance
  - Definition
  - Wasserstein GAN
  - Wasserstein Auto-Encoder
- Adversarial Loss
  - Different designs

Warm Up

- Generated room pictures by WGAN-GP



- [Face-off by CycleGAN]()

Generative Adversarial Networks

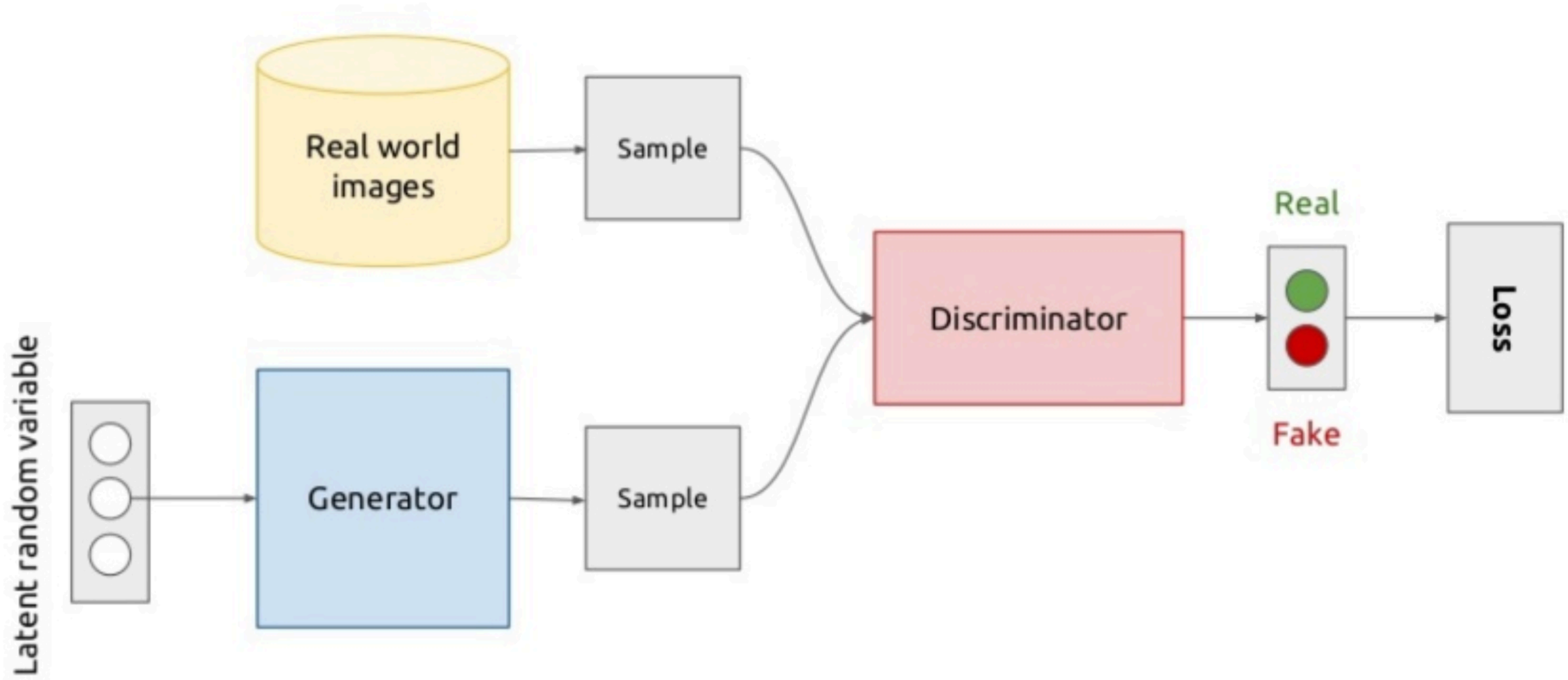
- Aim to generate fake data that *looks like* real data.
- Generator and Discriminator play an adversarial game
  - Generator tries to generate data that can *fool* the Discriminator, while Discriminator tries to distinguish between real data and generated data.

- Turing test
  - Test whether a machine can perform indistinguishably from a human.
- Nash Equilibrium
  - Every player reaches the best strategy as long as other players' decisions remain unchanged.

Generative Adversarial Networks

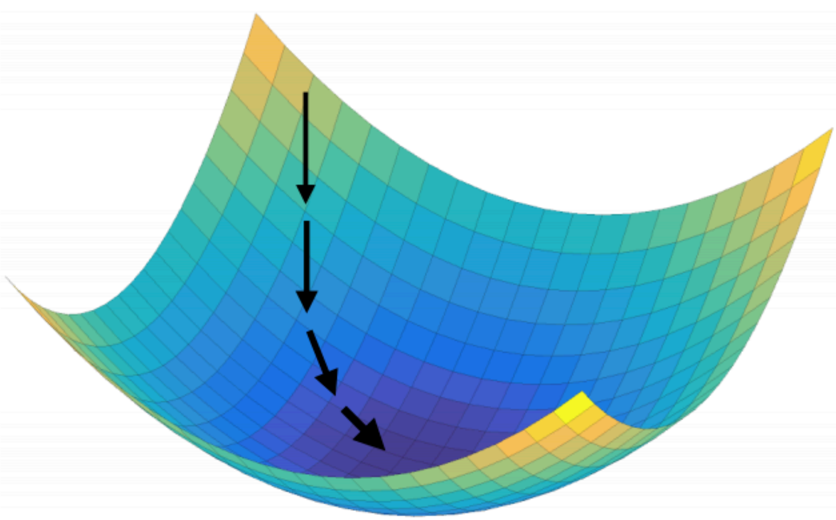
- Original formulation

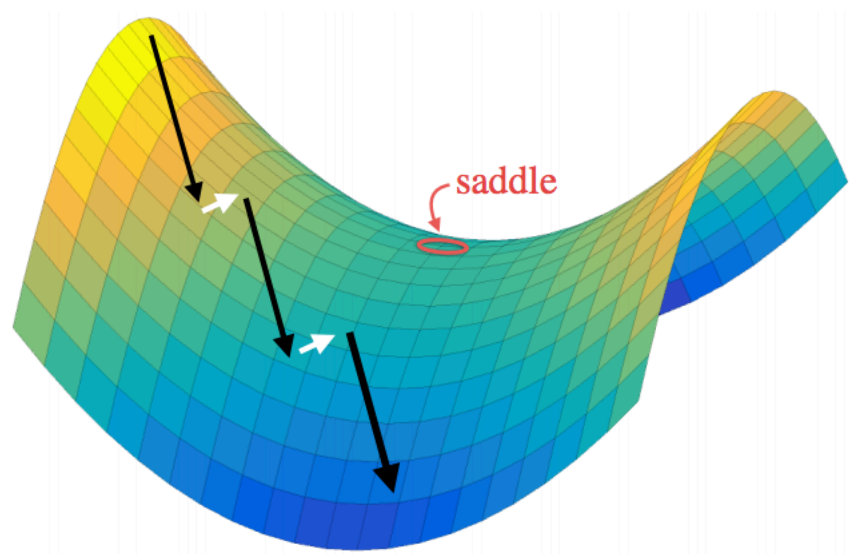
$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



Saddle Point Optimization

- Convex optimization v.s. saddle point optimization
  - Convex: descending along the gradient with reasonable learning rate guarantees global optimum
  - Saddle: the optimal point is fragile and hard to reach

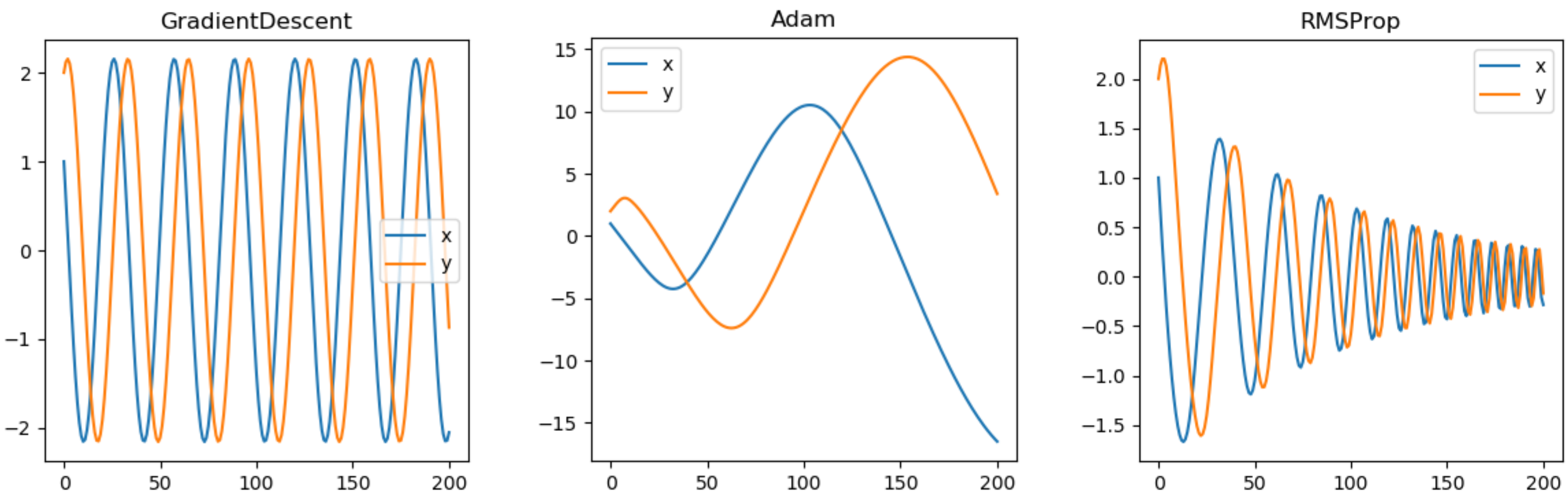




Saddle Point Optimization

- Hard to converge with gradient descent.

$$\min_x \max_y \; xy$$



– Initialize $x = 1$, $y = 2$. Same learning rate with Gradient Descent, Adam and RMSProp. Only RMSProp converges.

Vanishing Gradient

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- When real, fake distributions hardly overlaps, it is easy to distinguish them. When *D* is optimal, the gradient of *G* vanishes.
- Denote the optimal Discriminator with D^*.
  when $\|D - D^*\| < \epsilon$, the gradient of *G*

$$\|\nabla_\theta \mathbb{E}_{z \sim p(z)}[\log(1 - D(g_\theta(z)))]\|_2 < M \frac{\epsilon}{1 - \epsilon}$$

  - In the beginning of training, generated samples are easy to distinguish.
  - Discriminator: good one or bad one?

Alternative objective for Generator

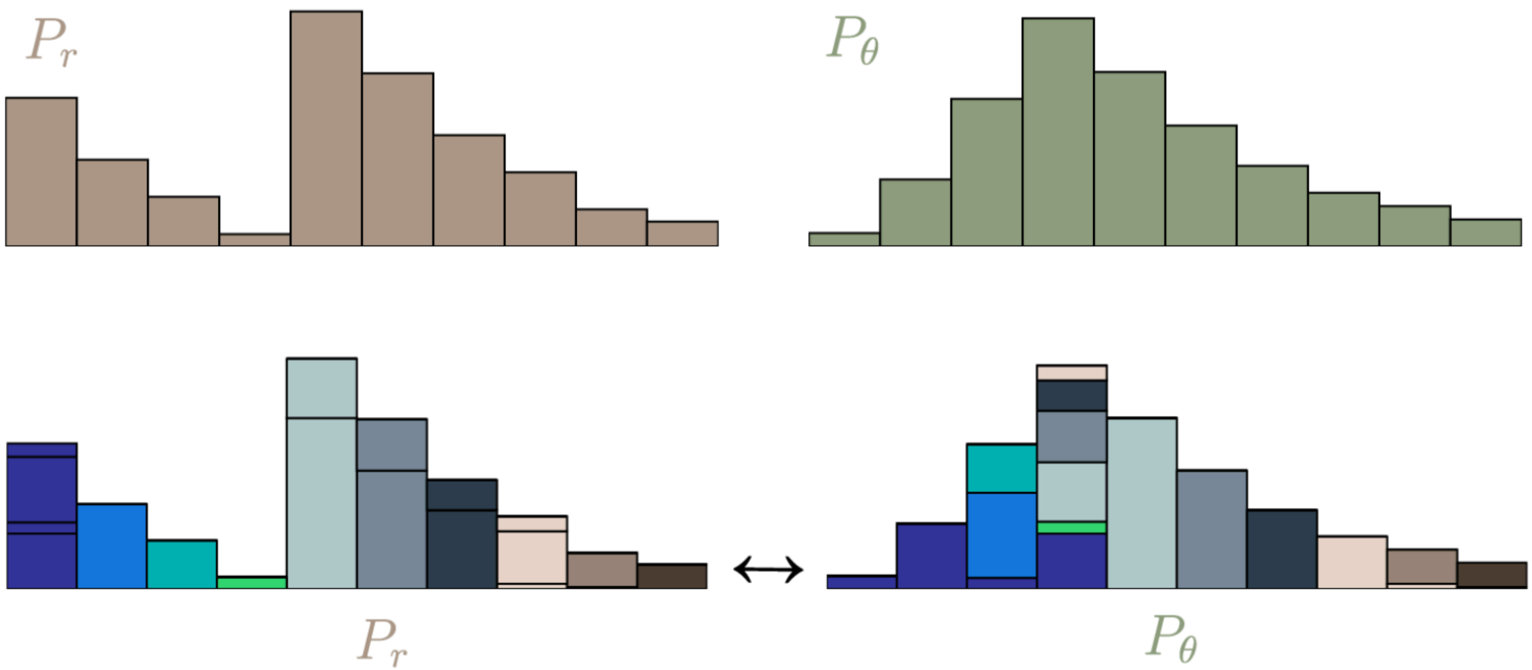
- Original $\mathbb{E}_{z\sim p(z)}[\log(1 - D(g_\theta(z)))]$
- Alternative $\mathbb{E}_{z\sim p(z)}\left[-\log D(g_\theta(z))\right]$
  - Alleviates the problem of gradient vanishing, but brings out new problems.
  - Equivalent to $[KL(\mathbb{P}_{g_\theta}\|\mathbb{P}_r) - 2JSD(\mathbb{P}_{g_\theta}\|\mathbb{P}_r)]$
- Problems
  - *KL* – 2*JSD* ?
  - Mode collapse: due to the asymmetric nature of KL-Divergence, the generation results of different latent codes are almost identical.

$$\nabla_{w_g} \int_x \log\left(\frac{P_g(x)}{P_r(x)}\right) P_g(x)dx = \int_x \Big(\nabla_{w_g} P_g(x)\Big) \log\left(\frac{P_g(x)}{P_r(x)}\right) dx$$

  - Instability of gradients: gradient is a centered Cauchy distribution with infinite expectation and variance

Wasserstein Distance

- Minimum *cost* of tuning a distribution to another



Wasserstein Distance

- Definition

$$W_p(\mu,\nu) := \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{M \times M} d(x,y)^p \, \mathrm{d}\gamma(x,y) \right)^{1/p}$$

  - $d(x, y)$: *distance* from x to y
  - $\mathrm{d}\gamma(x, y)$: *mass* moved from x to y

- Measures the distance between two distributions. p=1 leads to Earth Mover's Distance (Optimal Transport).

Distance Metrics for Distribution

- Total Variation distance

$$\delta(P_r, P_g) = \sup_{A} |P_r(A) - P_g(A)|$$

- Kullback–Leibler divergence

$$KL(P_r \| P_g) = \int_x \log\left(\frac{P_r(x)}{P_g(x)}\right) P_r(x)\, dx$$

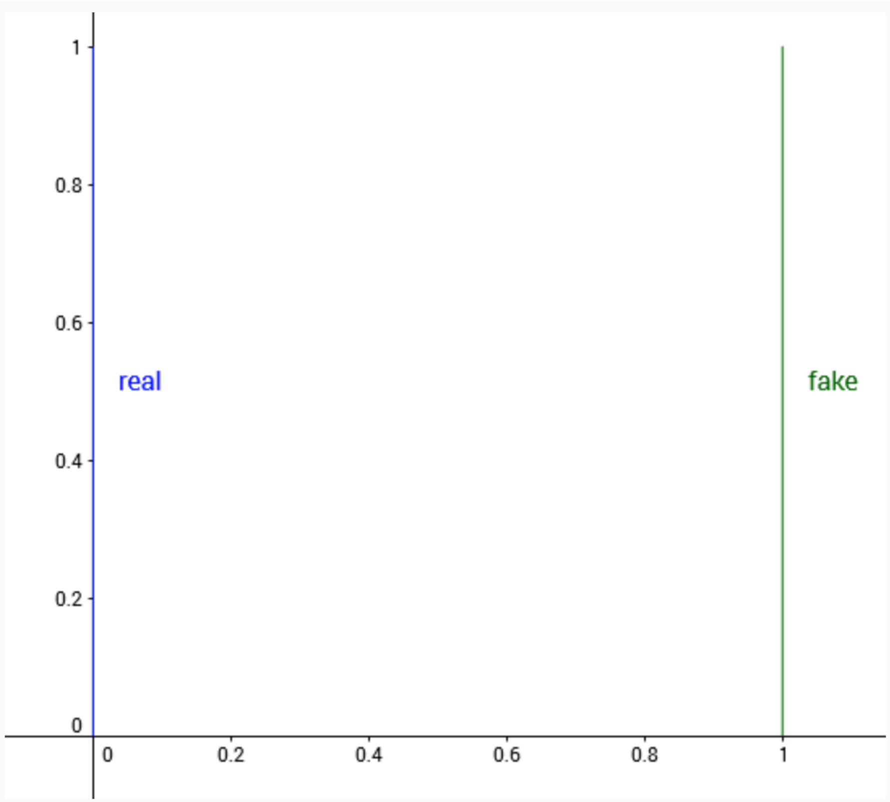
- Jensen–Shannon divergence

$$JS(P_r, P_g) = \frac{1}{2} KL(P_r \| P_m) + \frac{1}{2} KL(P_g \| P_m)$$

- Wasserstein distance

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

Problem with Non-overlap Distributions



Real and fake distribution when $\theta = 1$

Consider two distributions: with z sampled from uniform distribution U[0, 1], one distribution is $(0, z)$, and the other is (θ, z).

Use a distance metric to measure the distance

$$W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|,$$

$$JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0 \ , \\ 0 & \text{if } \theta = 0 \ , \end{cases}$$

$$KL(\mathbb{P}_\theta \| \mathbb{P}_0) = KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = \begin{cases} +\infty & \text{if } \theta \neq 0 \ , \\ 0 & \text{if } \theta = 0 \ , \end{cases}$$

$$\text{and } \delta(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} 1 & \text{if } \theta \neq 0 \ , \\ 0 & \text{if } \theta = 0 \ . \end{cases}$$

*Recall the Vanishing Gradient problem.

Wasserstein Distance

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big]$$

- Intractable: hard to exhaust all joint distributions.
  - Many approximations (papers).
- Kantorovich-Rubinstein Duality

$$W(P_r, P_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_\theta}[f(x)]$$

  - f : all functions satisfying 1-Lipschitz continuity.
  - Equivalent to deal with K-Lipschitz restriction.
    - derivatives are bounded

Wasserstein GAN

① Approximate Wasserstein distance with neural networks

$$\max_{w \in \mathcal{W}} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z \sim p(z)}[f_w(g_\theta(z)]$$

– Weight clipping to enforce Lipschitz continuity (bound derivatives of x)

② Minimize the approximated distance

$$\min_{\theta} \underbrace{\mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)]}_{\text{ignored}} - \mathbb{E}_{z \sim p(z)}[f_w(g_\theta(z)]$$

Wasserstein GAN

$$\min_{\theta} \max_{w} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z \sim p(z)}[f_w(g_\theta(z)]$$

- Samples are mapped to a scalar, 1-D latent space.
- "Discriminator" is instead called "Critic"
  - No longer used to classify, but provides distance feedback
- Code changes compared to GAN:
  - Remove the last classification layer
  - Weight clipping
- Problem: terrible way to enforce Lipschitz Continuity with gradient clipping
  - Refer to WGAN-GP (Gradient Penalty) for more details

Wasserstein Auto-Encoder

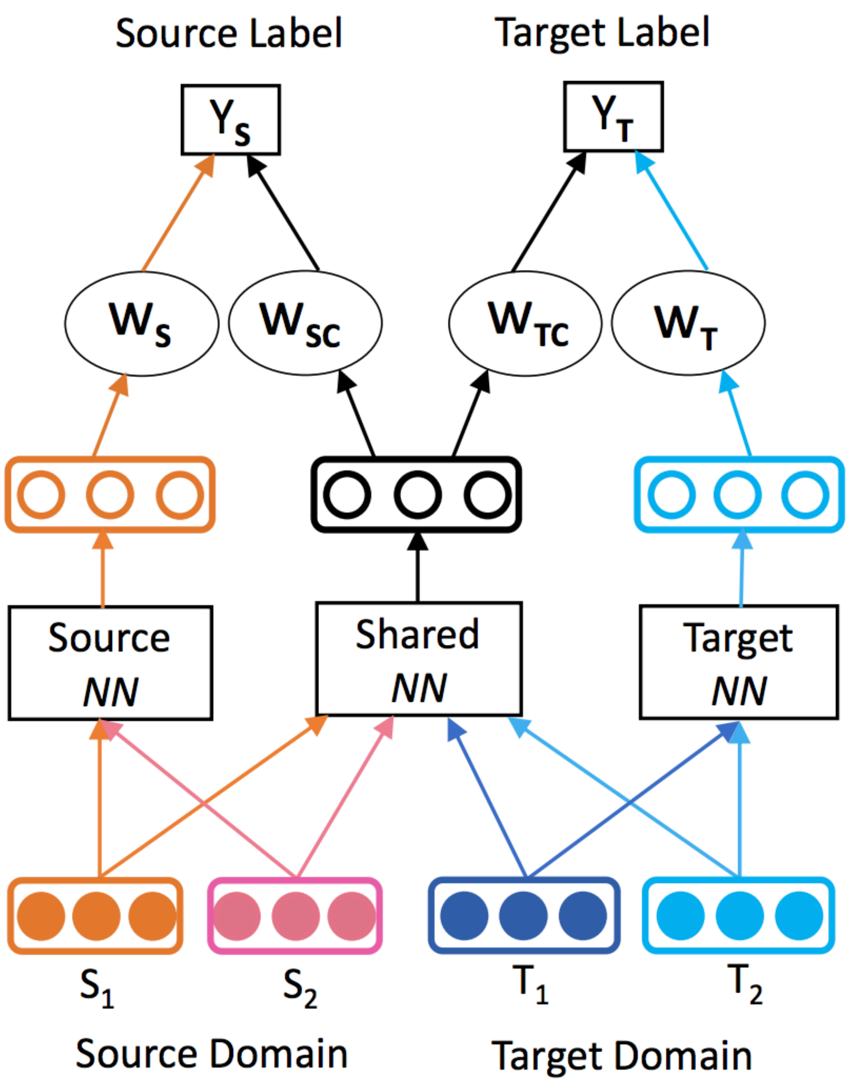
- WGAN: distribution distance is measured in the sample level.
- Move the distribution distance measuring to the latent code level ➔ WAE

$$\inf_{\Gamma \in \mathcal{P}(X \sim P_X, Y \sim P_G)} \mathbb{E}_{(X,Y)\sim\Gamma} \big[c\big(X, Y\big)\big] = \inf_{Q:\, Q_Z = P_Z} \mathbb{E}_{P_X} \mathbb{E}_{Q(Z|X)} \big[c\big(X, G(Z)\big)\big]$$

$$\inf_{Q(Z|X) \in \mathcal{Q}} \mathbb{E}_{P_X} \mathbb{E}_{Q(Z|X)} \big[c\big(X, G(Z)\big)\big] + \lambda \cdot \mathcal{D}_Z(Q_Z, P_Z)$$

- Refer to [WASSERSTEIN AUTO-ENCODERS] for more details.

Adversarial Loss



- A popular module in transfer learning tasks to learn *shared* representation between source domain and target domain.

Adversarial Loss Design 1

- Add the following negative entropy term to the objective and jointly optimize

$$\min_{\theta,w} \sum_{x\in\{S\cup T\}} \sum_{d\in\{s,t\}} p_\theta(d|g_w(x)) \log\Big(p_\theta(d|g_w(x))\Big)$$

- Many problems. List some:
  - $p = 0.5$ for both s and t can achieve optimal loss
    - A poor Discriminator, such as $\theta = \mathbf{0}$
    - A poor shared representation, such as $w = \mathbf{0}$
  - both can lead to optimal loss, but no prevention in the designed objective.

Adversarial Loss Design 2

- Add the cross entropy term as a min-max game

$$\min_{\theta} \max_{w} \; -\sum_{x} 1\{x \in S\} \log \Big(p_\theta(d = s | g_w(x)) \Big) + 1\{x \in T\} \log \Big(p_\theta(d = t | g_w(x)) \Big)$$

- Balance sample numbers in *S*, *T* and reformulate

$$\min_{w} \max_{\theta} \; \mathbb{E}_{x \in S}[\log(D_\theta(g_w(x)))] + \mathbb{E}_{x \in T}[\log(1 - D_\theta(g_w(x)))]$$

  – *D*, *g* share same status
    - *D*: for *x* in *S*, *D*(*g*(*x*)) ➔ 1; for *x* in *T*, *D*(*g*(*x*)) ➔ 0
    - *g* : for *x* in *S*, *D*(*g*(*x*)) ➔ 0; for *x* in *T*, *D*(*g*(*x*)) ➔ 1

- Ideal equilibrium: *x* from *S* and *T* are indistinguishable, *D*(*g*(*x*)) ➔ 0.5
  – Can this objective achieve this equilibrium?

Adversarial Loss Design 2

$$\min_w \max_\theta \ \mathbb{E}_{x\in S}[\log(D_\theta(g_w(x)))] + \mathbb{E}_{x\in T}[\log(1 - D_\theta(g_w(x)))]$$

- Apply chain rule and see what happens to gradient
  - D_θ $\quad \nabla_\theta = \int_x \nabla_\theta D \, \frac{p_s - (p_s + p_t)D}{D(1-D)} \, dx$
  - g_w $\quad \nabla_w = \int_x \nabla_g D \nabla_w g \, \frac{p_s - (p_s + p_t)D}{D(1-D)} \, dx$

- $D(g(x)) = p_s(x) / (p_s(x) + p_t(x))$ converges for both θ, w
  - When $D(g(x))$ outputs correct domain label, both D, g converge.

Adversarial Loss Design 3

- Hybrid solution: entropy & cross entropy

$$\min_g \sum_x \sum_d p_\theta(d|g(x)) \log(p_\theta(d|g(x)))$$

$$\min_\theta \; -\sum_x 1\{x \in S\} \log(p_\theta(d = s|g(x))) + 1\{x \in T\} \log(p_\theta(d = t|g(x)))$$

– D_θ $\quad \nabla_\theta = \int_x \nabla_\theta D \, \frac{p_s - (p_s + p_t)D}{D(1-D)} \, dx$

– g_w $\quad \nabla_w = \sum_{x \in \{S \cup T\}} \nabla_g D \nabla_w g \log \frac{D}{1-D}$

Adversarial Loss Design 4

- Apply Discriminator on both *shared* and *specific* representation

$$\min_g \sum_x \sum_d p_\theta(d|g(x)) \log(p_\theta(d|g(x)))$$

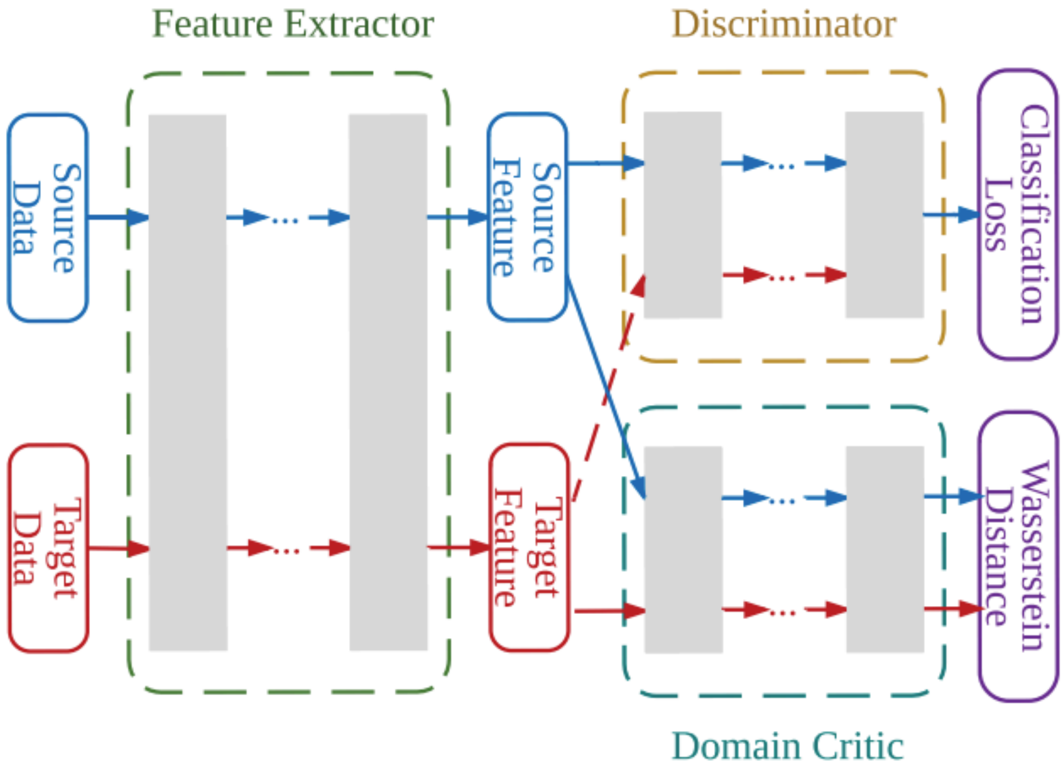
$$\min_\theta \; -\sum_x 1\{x \in S\} \log(p_\theta(d = s|f_s(x))) + 1\{x \in T\} \log(p_\theta(d = t|f_t(x))) + \lambda\Big(1\{x \in S\} \log(p_\theta(d = s|g(x))) + 1\{x \in T\} \log(p_\theta(d = t|g(x)))\Big)$$

  - f_s, f_t: *specific* network in source, target domain
  - g: *shared* network in both domains

- Possibly better than the previous design, but requires *specific* representation

Adversarial Loss Design 5

- Shared representation should be both indistinguishable and meaningful
  - Use Wasserstein distance to pull close shared representations
  - Add a task on the shared representations to enrich content



References

1. Goodfellow, Ian, et al. "Generative adversarial nets." NIPS. 2014.
2. Salimans, Tim, et al. "Improved techniques for training gans." NIPS. 2016.
3. Arjovsky, Martin, et al. “Towards principled methods for training generative adversarial networks.” ICLR. 2017.
4. Arjovsky, Martin, et al. “Wasserstein gan.” ICML. 2017.
5. Gulrajani, Ishaan, et al. “Improved training of wasserstein gans.” NIPS. 2017.
6. Shen, Jian, et al. “Wasserstein Distance Guided Representation Learning for Domain Adaptation.” AAAI. 2018.
7. Yadav, Abhay, et al. “Stabilizing Adversarial Nets With Prediction Methods.” ICLR. 2018.
8. Jun-Yan Zhu, et al. “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks.” ICCV. 2017.
9. Tolstikhin, Ilya, et al. “Wasserstein Auto-Encoders.” ICLR. 2018.
10. Yu, Jianfei, et al. “Modelling Domain Relationships for Transfer Learning on Retrieval-based Question Answering Systems in E-commerce.” WSDM. 2018.
11. Qiu, Minghui, et al. “Transfer Learning for Context-Aware Question Matching in Information-seeking Conversations in E-commerce.” ACL. 2018.
12. Ganin, Yaroslav, et al. “Unsupervised Domain Adaptation by Backpropagation.” ICML. 2015.