CS 598: Deep Generative and Dynamical Models

GAN3

Presented by Xiaoyang Bai

Wasserstein GAN

GAN Training as Distribution Matching

- Ground truth distribution $\mathbf{p}_r$ and generated distribution $\mathbf{p}_g$
- Vanilla GAN minimizes Jensen-Shannon divergence (JSD)
- f-GAN generalizes to the family of f-divergences
- Other distance functions:
  - Total variance (TV)
  - KL divergence
  - Earth-Mover distance (i.e. Wasserstein-1)

Earth-Mover Distance

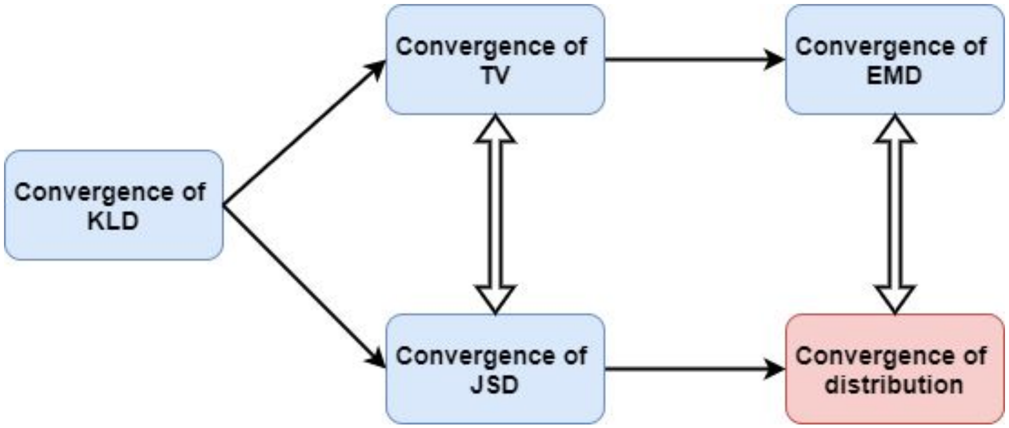
- Measures how much “mass” needs to be transported between distributions
- Formally:

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

- **γ** defines a matching between $\mathbf{p_r}$ and $\mathbf{p_g}$
- EMD defined as the minimal expected distance between matched points

Earth-Mover Distance

- Now consider optimizing for the true distribution $\mathbf{p}_r$
- EMD is the most sensible choice:



Earth-Mover Distance

- But we want EMD to be continuous and differentiable
- There are two conditions:
  - The generator **g** is continuous in $\boldsymbol{\theta}$
  - The generator **g** is locally Lipschitz
- A feed-forward NN as generator with finite prior distribution $\mathbf{p_z}$ suffices!

Wasserstein GAN (WGAN)

- Basically replacing discriminator criterion with EMD
- The original formulation is intractable for distribution dimension >1:

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

- We can apply Kantorovich-Rubinstein duality:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \le 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

- Take supremum over every 1-Lipschitz **f** that projects **x** down to a scalar

Wasserstein GAN (WGAN)

- Integrating that into the GAN pipeline:
  - Let the discriminator weight **ω** come from a compact space **W**
  - Then the discriminator $\mathbf{d}_{\boldsymbol{\omega}}$ is **K**-Lipschitz
  - Calculate EMD (batch-wise mean difference) as discriminator loss
- Formally:

$$\mathcal{L}_D = \frac{1}{m}\sum_{i=1}^{m} D_\omega(G_\theta(\mathbf{z}_i)) - D_\omega(\mathbf{x}_i)$$

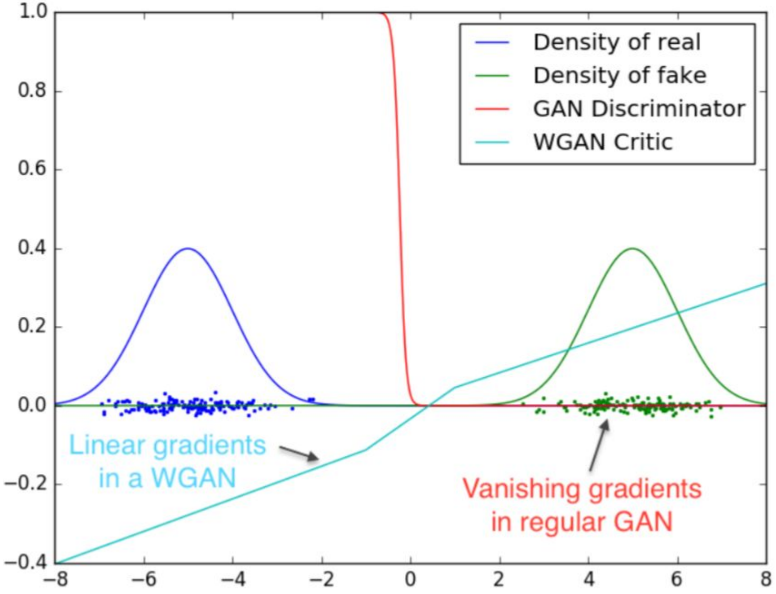
$$\mathcal{L}_G = -\frac{1}{m}\sum_{i=1}^{m} D_\omega(G_\theta(\mathbf{z}_i))$$

Wasserstein GAN (WGAN)

- Challenge: how to make **D** Lipschitz?
  - Weight clipping
  - Projecting weights to unit sphere
- We’ll see a much better approach in the next paper

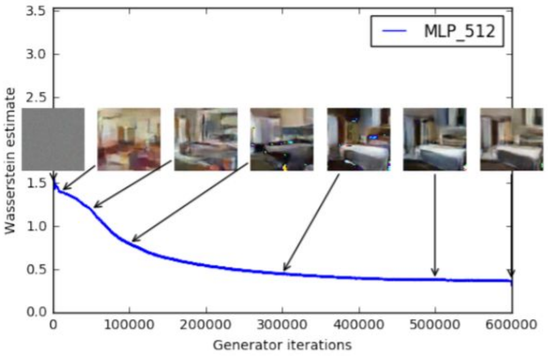
Advantage of WGAN

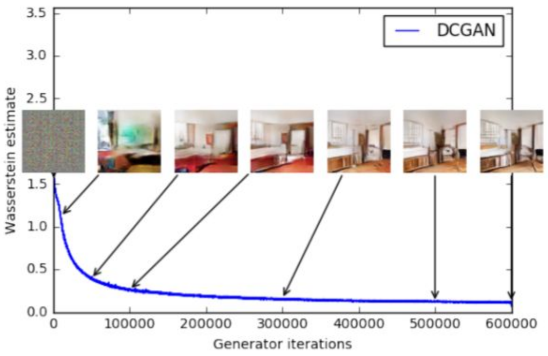
- No vanishing gradients with well-trained discriminator:



Experiments

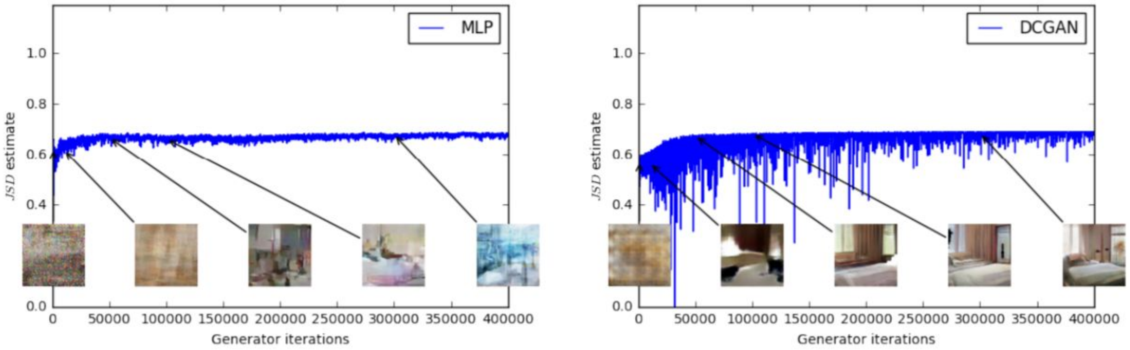
- Train on LSUN-bedroom with MLP/DCGAN as generator
    - Smooth decreasing loss as training progresses for WGAN





Experiments

- Non-smooth static loss for vanilla GAN:
    - And MLP doesn't work!



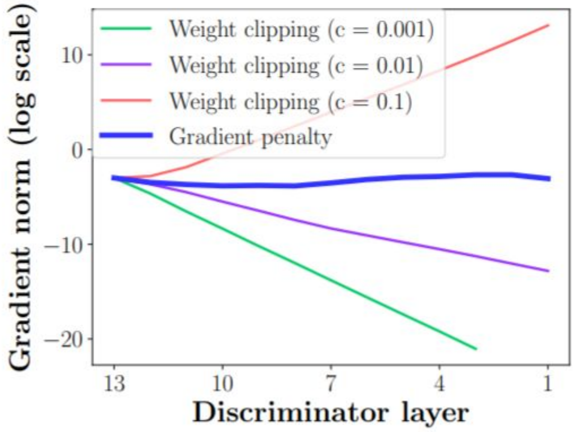
Improved Training of Wasserstein GANs

Recap on WGAN

- Weight clipping to enforce Lipschitz causes problem
  - Suboptimal generation results
  - End up learning simple functions
  - Gradient vanishing and explosion

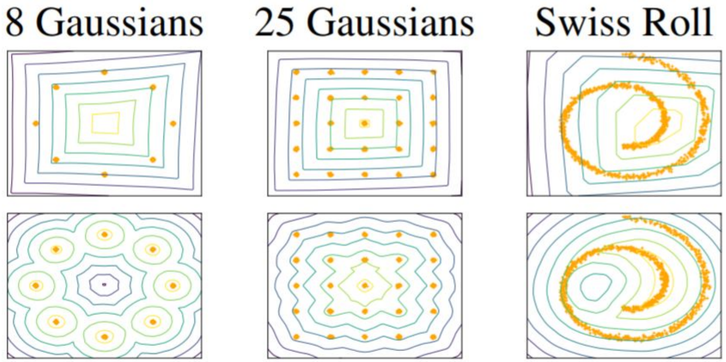
Recap on WGAN

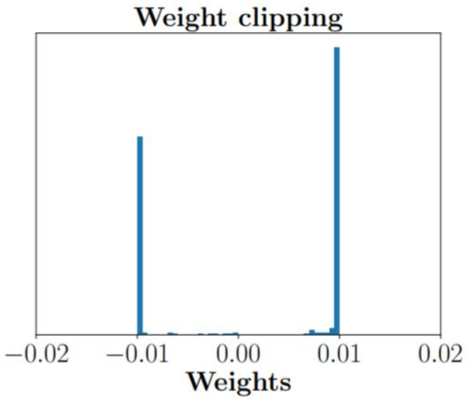
- The paper proves:
  - Optimal **f** for the KR-dual form has gradient norm 1 almost everywhere
  - Which is not the case for WGAN



Recap on WGAN

- WGAN learns very simple functions (left)
- And its weights are pushed to extremes (right)





Gradient Penalty

- Add a loss term to constraint the gradient norm of **D**
- Use the fact that for **f** in the KR-dual form,

$$P_{(x,y)\sim\pi}[\nabla f(x_t) = \frac{y - x_t}{\| y - x_t \|}] = 1, \text{ where } x_t = tx + (1-t)y$$

- So,
  - Sample **t** uniformly from [0, 1]
  - Interpolate between real and fake data points using **t** as weight
  - Calculate gradient of **D** w.r.t. interpolated data
  - Constraint norm to be 1

Gradient Penalty

- Formally:
  - **λ** is chosen to be 10 in experiments

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \left[D(\tilde{\boldsymbol{x}}) \right] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \left[D(\boldsymbol{x}) \right]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}} \left[(\| \nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}}) \|_2 - 1)^2 \right]}_{\text{Our gradient penalty}} .$$

- Two-side penalty instead of one-side
  - Authors claim that there is little difference empirically

Gradient Penalty

- Also removed batch normalization in **D**
    - Since it changes the target of **D** from single data points to whole batches
    - Which is inconsistent with the GP term

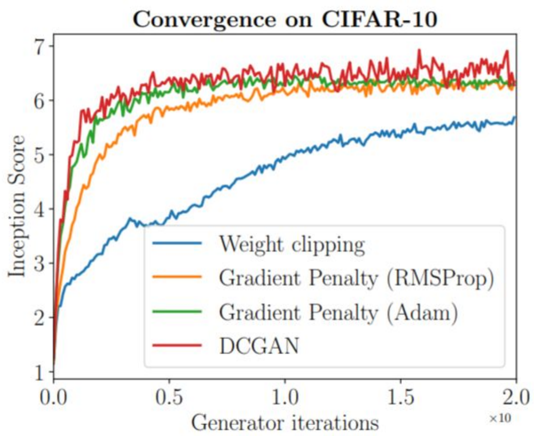
Experiments

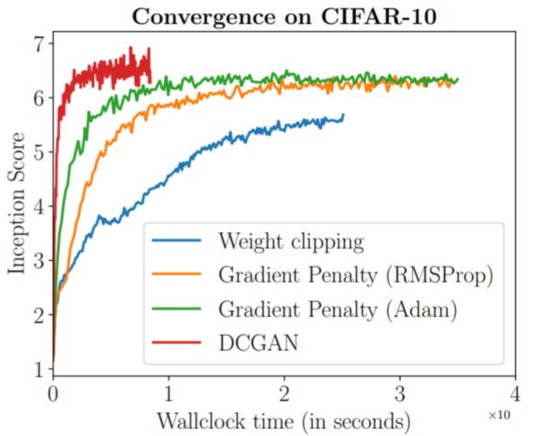
- Train vanilla GAN and WGAN-GP on 200 random architectures
  - WGAN-GP is more robust towards architecture shift

Min. score	Only GAN	Only WGAN-GP	Both succeeded	Both failed
1.0	0	8	192	0
3.0	1	88	110	1
5.0	0	147	42	11
7.0	1	104	5	90
9.0	0	0	0	200

Experiments

- Outperforms DCGAN and WGAN with weight clipping quantitatively
  - Inception score (IS) as metric





Experiments

- More quantitative results...

Unsupervised

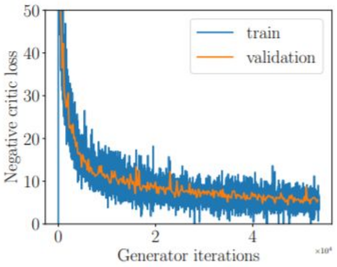
Method	Score
ALI [8] (in [27])	$5.34 \pm .05$
BEGAN [4]	5.62
DCGAN [22] (in [11])	$6.16 \pm .07$
Improved GAN (-L+HA) [23]	$6.86 \pm .06$
EGAN-Ent-VI [7]	$7.07 \pm .10$
DFM [27]	$7.72 \pm .13$
WGAN-GP ResNet (ours)	$7.86 \pm .07$

Supervised

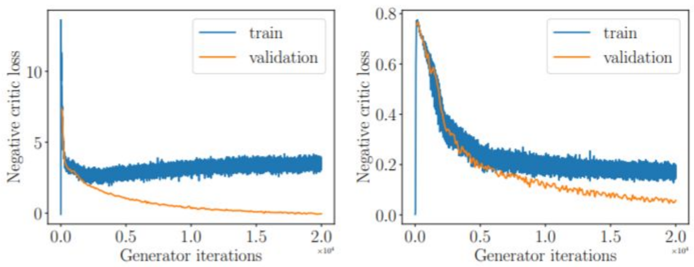
Method	Score
SteinGAN [26]	6.35
DCGAN (with labels, in [26])	6.58
Improved GAN [23]	$8.09 \pm .07$
AC-GAN [20]	$8.25 \pm .07$
SGAN-no-joint [11]	$8.37 \pm .08$
WGAN-GP ResNet (ours)	$8.42 \pm .10$
SGAN [11]	$8.59 \pm .12$

Experiments

- Interpreting loss curves
  - Negative critic loss converges for LSUN-bedroom (left)
  - Overfitting of **D** detected for both WGAN (right-right) and WGAN-GP (right-left)



(a)



(b)

Experiments

- Qualitative results (left: unconditional, right: conditional)