Improved Wasserstein-GAN

(April26, 2017)

Cheng Jianlin

Outline

- I. Introduction
- II. Optimal WGAN critic
- III. Improved Wasserstein GAN
- IV. Results
- V. Conclusion

• I. Introduction

GAN

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

$$\begin{aligned} C(G) =& \max_D V(G,D) \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\ =& \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] \end{aligned}$$

Min: $C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right)$

• I. Introduction

Wasserstein GAN

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

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

$$\min_G \max_{D\in\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}\big[D(\boldsymbol{x})\big] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}\big[D(\tilde{\boldsymbol{x}})\big]$$

$$\max_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$$

• I. Introduction

Wasserstein GAN

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

• II. Optimal WGAN critic

Lemma 1. *Let $\mathbb{P}_r$ and $\mathbb{P}_g$ be two distributions in $\mathcal{X}$, a compact metric space. Then, there is a 1-Lipschitz function f^* which is the optimal solution of*

$$\max_{\|f\|_L \leq 1} \mathop{\mathbb{E}}_{y \sim \mathbb{P}_r} [f(y)] - \mathop{\mathbb{E}}_{x \sim \mathbb{P}_g} [f(x)]$$

Let π be the optimal coupling between $\mathbb{P}_r$ and $\mathbb{P}_g$, defined as the minimizer of:

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

Where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ is the set of joint distributions $\pi(x, y)$ whose marginals are $\mathbb{P}_r$ and $\mathbb{P}_g$, respectively. Then, if f^* is differentiable [2] and $x_t = tx + (1-t)y$ with $0 \leq t \leq 1$,

$$\mathbb{P}_{(x,y) \sim \pi} \left[\nabla f^*(x_t) = \frac{y - x_t}{\|y - x_t\|} \right] = 1$$

• III. Improved Wasserstein GAN

Gradient penalty

As we discussed in the previous section, the use of weight clipping in WGAN can result in undesirable behaviors. We consider an alternative method to enforce the Lipschitz constraint on the training objective: a differentiable function is 1-Lipschitz if and only if it has gradients with norm less than or equal to 1 everywhere, so we would like to directly constrain the gradient norm of our critic function with respect to its input.

Exactly enforcing this constraint is not easily tractable, so instead we enforce a soft version: at certain points sampled from a distribution over the input space $\hat{x} \sim \mathbb{P}_{\hat{x}}$, we evaluate the gradient of the critic $\nabla_{\hat{x}} D(\hat{x})$ and penalize its squared distance from 1 in the critic loss function. Our new objective for the critic is:

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}\left[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}} \tag{3}$$

In the limit of high λ, the optimal critic under our formulation is still the optimal critic under the true Kantorovich-Rubinstein dual. Therefore, given enough capacity on the critic the cost function

• III. Improved Wasserstein GAN

Algorithm 1 WGAN with gradient penalty. We use default values of $\lambda = 10$, $n_{\text{critic}} = 5$, $\alpha = 0.0001$, $\beta_1 = 0.5$, $\beta_2 = 0.9$.

Require: The gradient penalty coefficient λ, the number of critic iterations per generator iteration n_{critic}, the batch size m, Adam hyperparameters α, β_1, β_2.
Require: initial critic parameters w_0, initial generator parameters θ_0.

while θ has not converged **do**
  for $t = 1, ..., n_{\text{critic}}$ **do**
    for $i = 1, ..., m$ **do**
      Sample real data $\boldsymbol{x} \sim \mathbb{P}_r$, latent variable $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random number $\epsilon \sim U[0, 1]$.
      $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z})$
      $\hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1 - \epsilon)\tilde{\boldsymbol{x}}$
      $L^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + \lambda(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
    end for
    $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha, \beta_1, \beta_2)$
  end for
  Sample a batch of latent variables $\{\boldsymbol{z}^{(i)}\}_{i=1}^{m} \sim p(\boldsymbol{z})$.
  $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -D_w(G_\theta(\boldsymbol{z})), \theta, \alpha, \beta_1, \beta_2)$
end while

• III. Improved Wasserstein GAN

Sampling along straight lines The gradient term $||\nabla_{\hat{x}} D(\hat{x})||_2$ is with respect to the points $\hat{x}$, not the parameters of D. We implicitly define the distribution of points $\mathbb{P}_{\hat{x}}$ at which to penalize the gradient by taking straight lines between points in the data distribution $\mathbb{P}_r$ and the generator distribution $\mathbb{P}_g$:

$$\epsilon \sim U[0,1], x \sim \mathbb{P}_r, \tilde{x} \sim \mathbb{P}_g \tag{4}$$

$$\hat{x} = \epsilon x + (1-\epsilon)\tilde{x} \tag{5}$$

Our motivation for penalizing the gradient over this distribution comes from the fact that the graph of the optimal critic consists of straight lines connecting points from $\mathbb{P}_r$ and $\mathbb{P}_g$ (see subsection 2.3). Given that enforcing the Lipschitz constraint everywhere is intractable, enforcing it only along these straight lines seems sufficient and experimentally results in good performance.

Hyperparameters Our proposed penalty term introduces one hyperparameter, λ, which controls the trade-off between optimizing the penalty term and the original objective. All experiments in this paper use $\lambda = 10$, which we found to work well across a variety of architectures and datasets ranging from toy tasks to large ImageNet CNNs.

No critic batch normalization Most prior GAN implementations (Radford et al., 2015; Salimans et al., 2016; Arjovsky et al., 2017) make use of batch normalization in both the generator and the discriminator to help stabilize training. However using batch normalization in the discriminator changes the form of the problem: instead of specifying a function mapping a single input to a single output, a discriminator with batch normalization specifies a function mapping from an entire batch of inputs to a batch of outputs (Salimans et al., 2016).

Our penalized training objective is no longer valid in this setting, since we penalize the norm of the critic's gradient with respect to each input independently, and not the entire batch. To resolve

• IV. Results

CIFAR-10

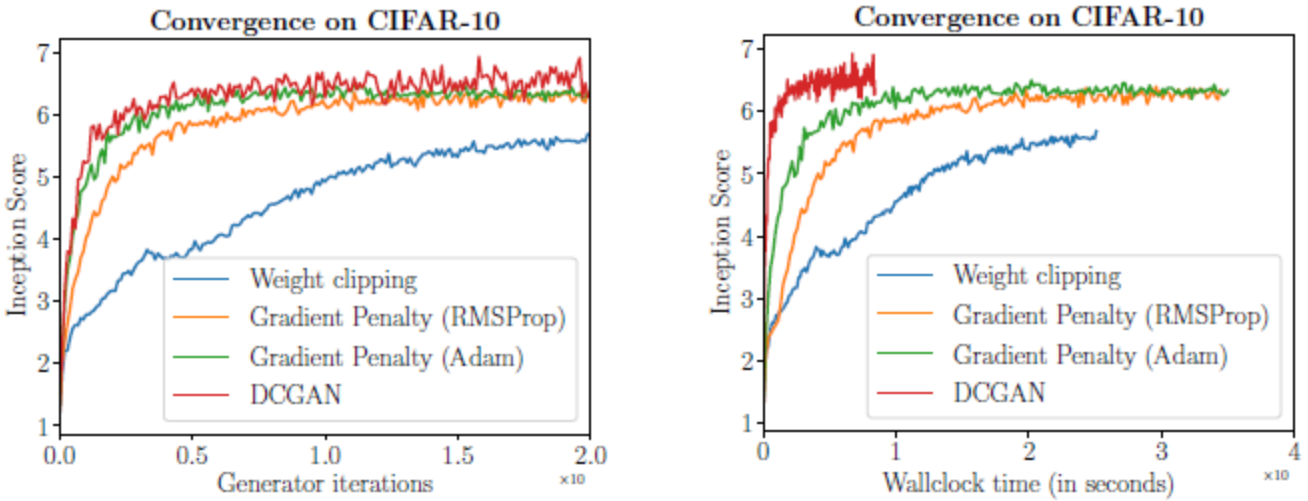

Figure 3: Plots of CIFAR-10 Inception score over generator iterations (left) or wall-clock time (right) for four models: WGAN with weight clipping, WGAN with gradient penalty and RMSProp (to control for the optimizer), WGAN with gradient penalty and Adam, and DCGAN. Even with the same learning rate, gradient penalty significantly outperforms weight clipping. DCGAN converges faster, but WGAN with gradient penalty achieves similar scores with improved stability.

• IV. Results

LSUN(Large-scale Scene Understanding) bedrooms dataset

• V. Conclusion

Improve performance and successfully train difficult GAN architectures

we think our work opens the path for strong modeling performance on large-scale image datasets and language

it might stabilize training by encouraging the discriminator to learn smoother decision boundaries

Bibliography

- [1] Wasserstein GAN
- [2] Improved Training of Wasserstein GANs

Thanks !