Wasserstein GAN

Courant Institute of Mathematical Sciences
Facebook AI Research

9 Mar 2017

April 17, 2017

Чем интересна данная статья?

- Новый алгоритм обучения GAN, который хорошо отрабатывает на популярных для генеративных моделей датасетах.
- Статья содержит математическое обоснование, почему метод работает и работает хорошо.
- Избавляемся от проблемы балансировки дискриминатора и генератора.
- Функция потерь значимо коррелирует с качеством генерации

Введение. Генеративные модели.

Пусть данные пришли из исходного распределения P_r. Мы хотим найти P_θ, которое аппроксимирует исходное. Два способа сделать это.

- Напрямую искать P_θ : $P_\theta \geq 0$ и $\int_x P_\theta(x)dx = 1$, максимизируя правдоподобие
- Искать P_θ, как преобразование $z \in \mathbb{Z}$. Пусть g_θ — дифференциируемая, z из известного распределения, а $g_\theta(z) = P_\theta$

MLE для P_θ

$$max_{\theta \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} logP_\theta(x^{(i)})$$

В пределе эквивалентно минимизации KL-divergence

Расстояния между распределениями

- Total Variation (TV) distance
  $\sigma(P_r, P_g) = sup_A|P_r(A) - P_g(A)|$
- Kullback-Leibler divergence
  $KL(P_r||P_g) = \int_x log(\frac{P_r(x)}{P_g(x)})P_r(x)dx$
- Jensen-Shannon divergence
  $JS(P_r, P_g) = KL(P_r||P_m) + KL(P_g||P_m)$,
  где $P_m = (P_r + P_g)/2$
- Earth-Mover distance or Wasserstein-1
  $W(P_r, P_g) = inf_{\gamma \in \prod(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||]$,
  где $\prod(P_r, P_g)$ — множество всех совместных распределений.

Пример, на котором Wasserstein лучше других метрик

- Total Variation (TV) distance
  $\sigma(P_r, P_g) = sup_A|P_r(A) - P_g(A)|$
- Kullback-Leibler divergence
  $KL(P_r||P_g) = \int_x log(\frac{P_r(x)}{P_g(x)})P_r(x)dx$
- Jensen-Shannon divergence
  $JS(P_r, P_g) = KL(P_r||P_m) + KL(P_g||P_m)$,
  где $P_m = (P_r + P_g)/2$
- Earth-Mover distance or Wasserstein-1
  $W(P_r, P_g) = inf_{\gamma \in \prod(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||]$,
  где $\prod(P_r, P_g)$ — множество всех совместных распределений.

Существуют последовательности распределений, которые не сходятся на JS, KL, reverse KL, TV, но сходятся на W. А так же, для этих функций градиент по θ всегда равен 0.

Теоремы

- Теорема 1
  Пусть P_r — фиксированное распределение на X.
  Пусть Z случайная величина из известного распределения (напр. Гауссовское или равномерное) на $\mathbb{Z}$
  Пусть $g : \mathbb{Z} \times \mathbb{R}^d \to X$ будет обозначена, как $g_\theta(z)$ с Z первой координатой и θ — второй. Пусть $P_\theta = g_\theta(Z)$.
  Тогда
  1. Если g непрерына в θ, то и $W(P_r, P_\theta)$ непрерына в ней.
  2. Если g локально Липшицева и удовлетворяет условиям, то $W(P_r, P_\theta)$ непрерывна везде и диф-ма почти всюду.
  3. пункты 1 и 2 не выполняются для JS и KL.
- Следствие из Теоремы 1
  Пусть g_θ — feedforward nn, параметризованная θ и p_z — распределение на $\mathbb{Z}$: $\mathbb{E}_{z \sim p(z)} ||z|| \leq \infty$ (например, Gaus, Unif)
  Тогда g — удолетворяет условию, а значит, $W(P_r, P_\theta)$ непрерывна везде и диф-ма почти всюду.

Теоремы

Теорема 2

Let P be a distribution, and $(P_n)_{n\in\mathbb{N}}$ be a sequence of distributions. Then, the following are true about the limit.

1. *The following statements are equivalent.*
   - $\delta(P_n, P) \to 0$ *with* δ *the total variation distance.*
   - $JS(P_n, P) \to 0$ *with* JS *the Jensen-Shannon divergence.*
2. *The following statements are equivalent.*
   - $W(P_n, P) \to 0$.
   - $P_n \to P$, *where* $\to$ *represents convergence in distribution for random variables.*
3. $KL(P_n \| P) \to 0$ *or* $KL(P \| P_n) \to 0$ *imply the statements in (1).*
4. *The statements in (1) imply the statements in (2).*

Wasserstein GAN

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

По Канторовичу-Рубинштейну W эквивалентно

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

Если заменить sup 1-L на K-L функции, то мупремум будет равен $KW(P_r, P_\theta)$

Wasserstein GAN

Пусть $\{f_w\}, w \in W$ — параметризированное семейство функций, где w – веса.

$$\max_{w \in \mathcal{W}} \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{x \sim P_\theta}[f_w(x)] \leq \sup_{\|f\|_L \leq K} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_\theta}[f(x)] = K \cdot W(P_r, P_\theta)$$

Вернемся к генеративным моделям. Хочется с помощью $P_\theta = g_\theta(Z)$ аппроксимировать P_r

Для фиксированного g_θ мы можем найти оптимальную f_w для Wasserstein distance. А затем с помощью backprop находим градиент $W(P_r, g_\theta(Z))$ по θ

$$\nabla_\theta W(P_r, P_\theta) = \nabla_\theta(\mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{z \sim Z}[f_w(g_\theta(z))]) = -\mathbb{E}_{z \sim Z}[\nabla_\theta f_w(g_\theta(z))]$$

Wasserstein GAN

Процесс обучения разбивается на 3 этапа.
1. Для фиксированного θ найдем оптимальную аппроксимацию $W(P_r, P_\theta)$, тренируя f_w
2. Когда нашли оптимальную f_w, высчитываем градиент по θ.
3. Обновляем θ и повторяем процесс.

Заметим, что все это выполняется только для K-Липшецевых функций, поэтому добавим clipping для весов, после их обновления $w \in [-c, c]$

Wasserstein GAN

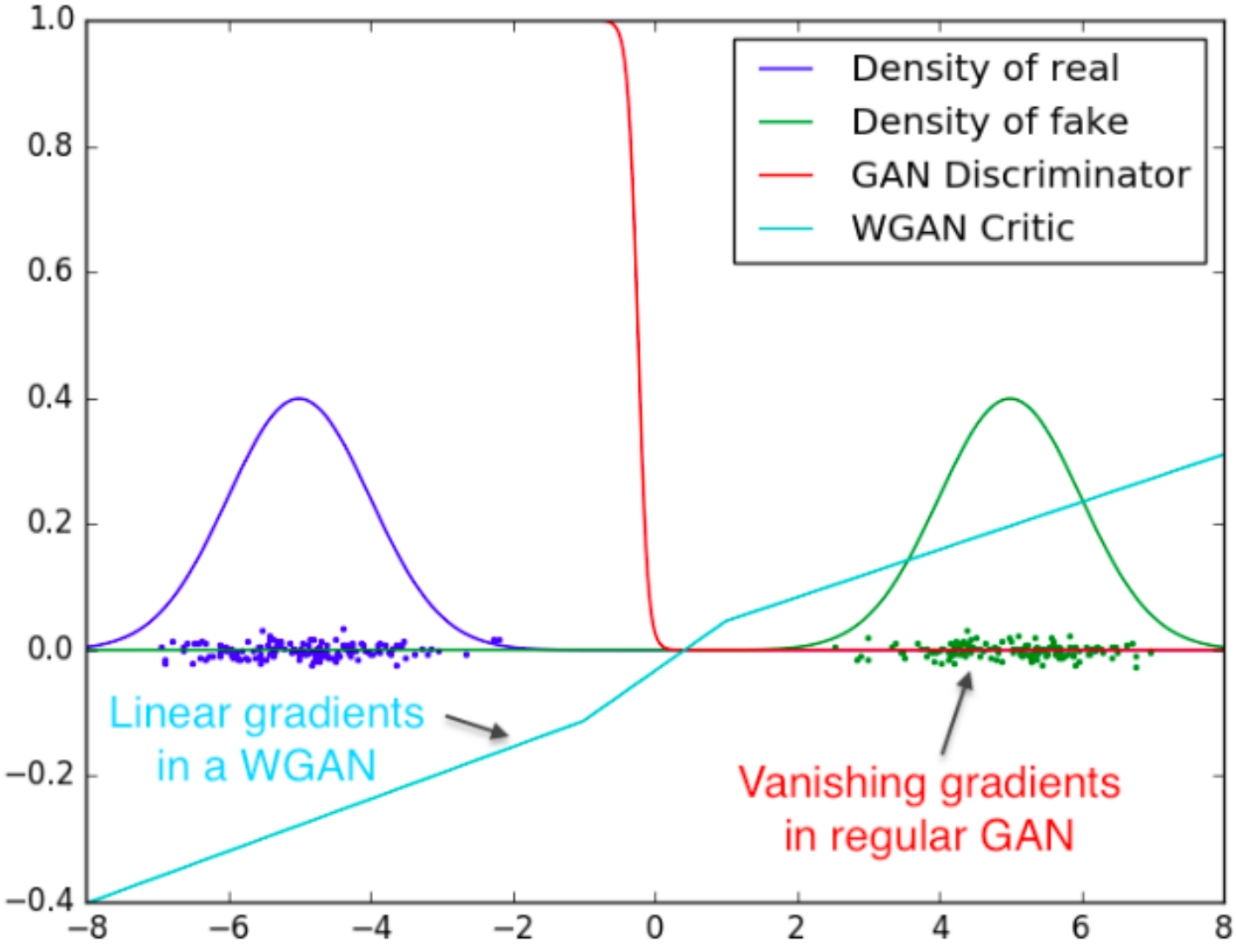
Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

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.

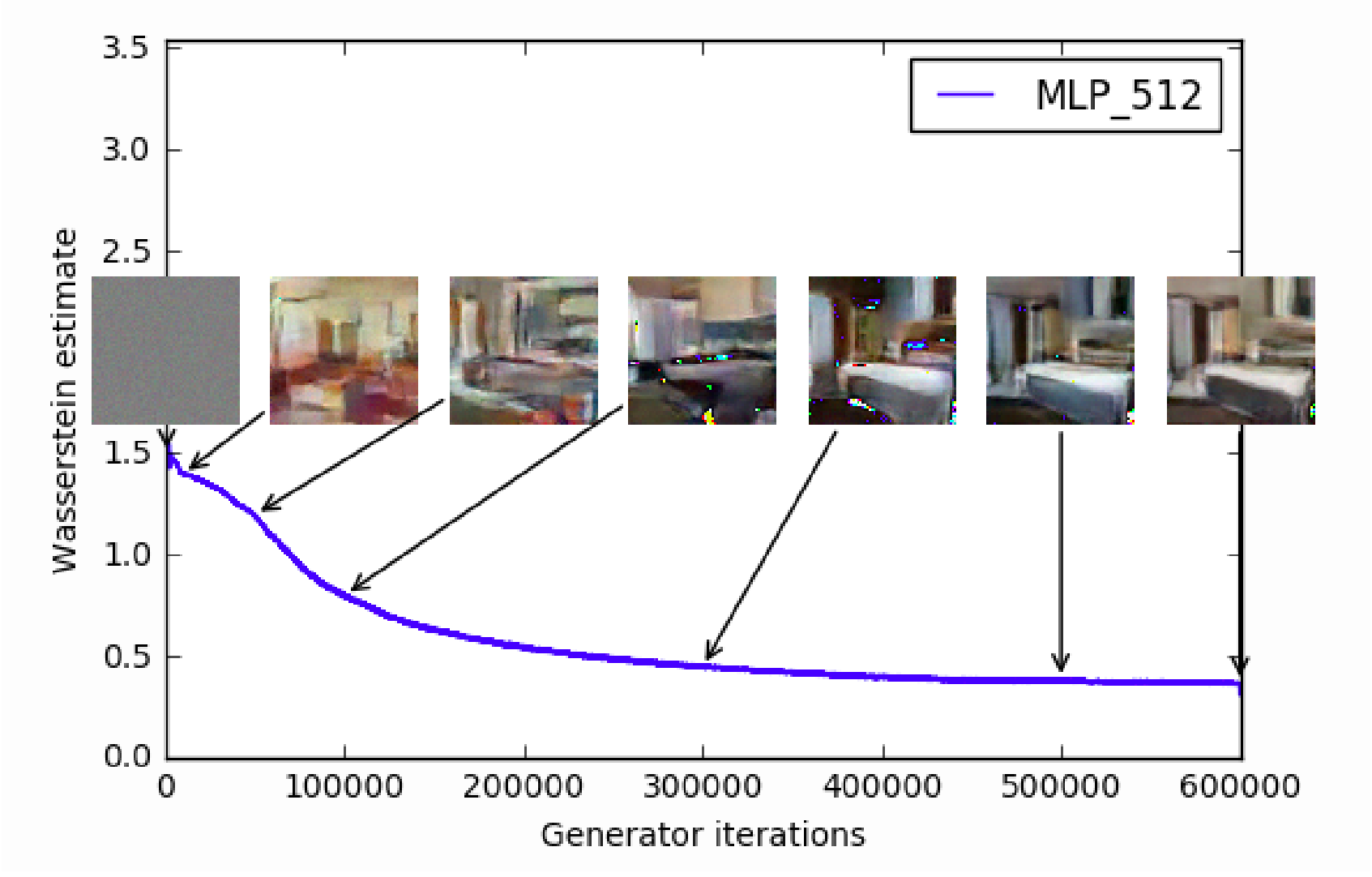
Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

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

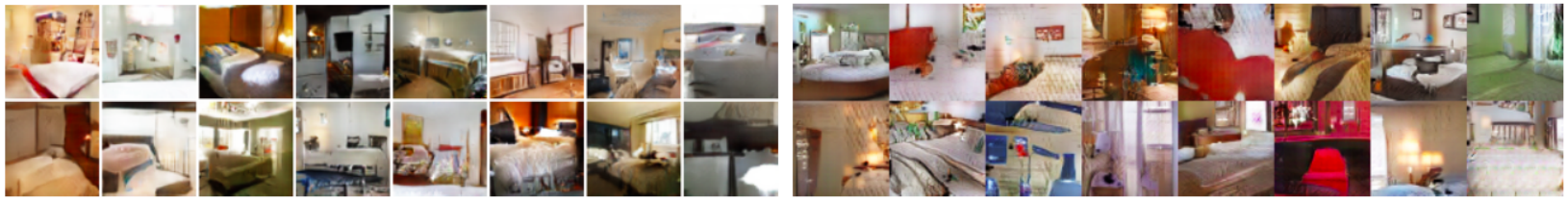
Wasserstein GAN



Wasserstein GAN



Wasserstein GAN



igure 5: Algorithms trained with a DCGAN generator. Left: WGAN algorithm. Right: andard GAN formulation. Both algorithms produce high quality samples.

Wasserstein GAN

Figure 6: Algorithms trained with a generator without batch normalization and constant number of filters at every layer (as opposed to duplicating them every time as in [18]). Aside from taking out batch normalization, the number of parameters is therefore reduced by a bit more than an order of magnitude. Left: WGAN algorithm. Right: standard GAN formulation. As we can see the standard GAN failed to learn while the WGAN still was able to produce samples.

Wasserstein GAN

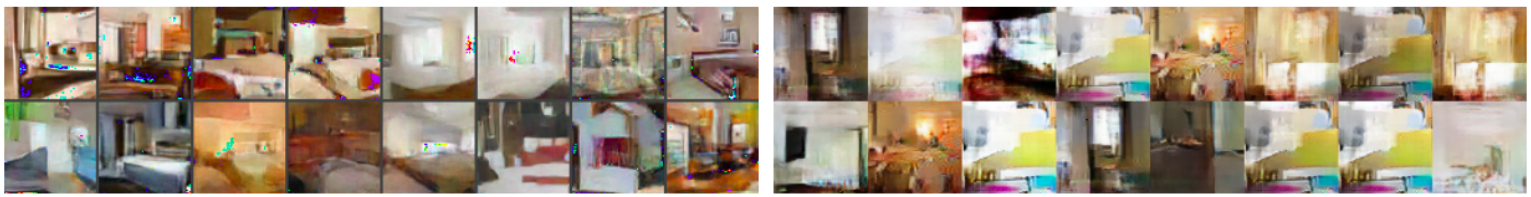


Figure 7: Algorithms trained with an MLP generator with 4 layers and 512 units with ReLU nonlinearities. The number of parameters is similar to that of a DCGAN, but it lacks a strong inductive bias for image generation. Left: WGAN algorithm. Right: standard GAN formulation. The WGAN method still was able to produce samples, lower quality than the DCGAN, and of higher quality than the MLP of the standard GAN. Note the significant degree of mode collapse in the GAN MLP.