Challenges of GANs

GAN 被認為極難訓練的根本原因是模型的架構會產生一些數學上難以克服的困難與矛盾。
我們知道，G 的 loss J_G 可以寫成：

$$J_G = \mathbb{E}_{x\sim p_g}[\log(1 - D(x)]$$

或：

$$J_G = -\mathbb{E}_{x\sim p_g}[\log(D(x))]$$

而兩種寫法都有各自的問題。
第一種寫法的問題在原始的 GAN 論文裡也有提及：D 越好，G 的梯度消失越嚴重，所以 GAN 的訓練有一個技巧：我們不能將 D 訓練的太好，不然 G 會學不下去。背後的原因是因為當 D 學得很好（幾乎完美），這時的 loss V 會是：

$$C(G) = V(D_G^*,\ G) = -\log 4 + 2 \cdot D_{\text{JS}}(p_{\text{data}}||p_g)$$

而若 $\text{supp}(p_g) \cap \text{supp}(p_{\text{data}}) = 0$，即兩個機率分佈完全不重疊的話，則根據定義有 $D_{\text{JS}}(p_{\text{data}}||p_g) = \log 2$，這導致 V 變成一個常數，對它取梯度是 0。要知道，在訓練初期隨機生成的 p_g 跟真實的 p_{data} 是有很大機會完全不相關的（甚至有更嚴謹的數學證明這幾乎一定發生），使得梯度消失成為一個不可忽視的問題。
第二種寫法的問題是：最小化這個 J_G，等價於最小化一個不合理的距離衡量，造成梯度不穩定與嚴重的 mode collapse。
首先，$D_{\text{KL}}(p_g||p_{\text{data}})$ 代入最優 D_G^*，是：

$$D_{\text{KL}}(p_g||p_{\text{data}}) = \mathbb{E}_{x\sim p_g}[\log(p_g/p_{\text{data}})] = \mathbb{E}_{x\sim p_g}[\log(1 - D_G^*(x)/D_G^*(x))]$$

$$= \mathbb{E}_{x\sim p_g}[\log(1 - D_G^*(x))] - \mathbb{E}_{x\sim p_g}[\log(D_G^*(x))]$$

因此，得到第二種寫法其實是：

$$J_G = -\mathbb{E}_{x\sim p_g}[\log(D_G^*(x))] = D_{\text{KL}}(p_g||p_{\text{data}}) - \mathbb{E}_{x\sim p_g}[\log(1 - D_G^*(x))]$$

$$= D_{\text{KL}}(p_g||p_{\text{data}}) - 2 \cdot D_{\text{JS}}(p_{\text{data}}||p_g) + \log 4 + \mathbb{E}_{x\sim p_{\text{data}}}[\log D_G^*(x)]$$

等價於最小化：

$$D_{\text{KL}}(p_g||p_{\text{data}}) - 2 \cdot D_{\text{JS}}(p_{\text{data}}||p_g)$$

這個最小化目標有兩個嚴重的問題。第一個問題是它要最小化生成分佈與真實分佈的 KL 散度，卻又要最大化兩者的 JS 散度，同時拉近跟推遠，這在直觀上是矛盾且荒謬的。第二個問題是 $D_{\text{KL}}(p_g||p_{\text{data}})$ 會給出兩種不平等的懲罰：

(1) $p_g(x) = 0,\ p_{\text{data}}(x) > 0$：$G$ 沒能涵蓋所有真實樣本，生成缺乏多樣性。

$$p_g(x) \to 0,\ p_{\text{data}}(x) \to 1,\ p_g(x) \cdot p_g(x)/p_{\text{data}}(x) \to 0,\ D_{\text{KL}}(p_g||p_{\text{data}}) \to 0$$

(2) $p_g(x) > 0,\ p_{\text{data}}(x) = 0$：$G$ 亂編出根本不存在的樣本，生成缺乏準確性。

$$p_g(x) \to 1,\ p_{\text{data}}(x) \to 0,\ p_g(x) \cdot p_g(x)/p_{\text{data}}(x) \to +\infty,\ D_{\text{KL}}(p_g||p_{\text{data}}) \to +\infty$$

KL 散度對「缺乏多樣性」的錯誤非常寬容，但對「缺乏準確性」的錯誤非常嚴厲，這將導致 G 變得保守，寧可多生成一些重複但安全的樣本，也不願意去生成有多樣性的樣本，造成 mode collapse。在實作中很容易發生的狀況是當 G 發現生成某種圖片能騙過 D 後，不管我們怎麼改變 noise，它都會一直生成那張圖片。

Wasserstein Distance

Wasserstein 距離又稱為 EM 距離 (earth-mover distance)，是 KL、JS 散度之外，一種描述兩個分佈之間的距離的方式，它的定義為：

$$W(p_{\text{data}},\ p_g) = \inf_{\gamma \in \Pi(p_{\text{data}},\ p_g)} \mathbb{E}_{(x,\ y) \sim \gamma}[||x - y||]$$

解釋如下：$\Pi(p_{\text{data}}, p_g)$ 是 p_{data} 和 p_g 組合起來的所有可能的聯合分布的集合，換句話說，$\Pi(p_{\text{data}}, p_g)$ 中每一個分布的邊緣分佈都是 p_{data} 和 p_g。對於每個可能的聯合分布 γ 而言，可以從中採樣 $(x, y) \sim \gamma$ 得到一個真實樣本 x 和一個生成樣本 y，並計算這對樣本的距離 $\|x-y\|$，因此可以計算該聯合分佈 γ 下樣本對距離的期望值 $\mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$。在所有可能的聯合分布中能夠對這個期望值取到的下界 $\inf_{\gamma\in\Pi(p_r,p_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$ 即為 Wasserstein 距離。直觀上，可以將 $\mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$ 理解為在 γ 這個「路徑規劃」下，將 p_r 這堆「沙土」搬運至 p_g 位置所需的「消耗」，而 $W(p_r, p_g)$ 則是「最優路徑規劃」下的「最小消耗」，因此被稱為 earth-mover 距離。Wasserstein 距離相比 KL 散度、JS 散度的優越性在於，即便兩個分佈沒有重疊，Wasserstein 距離仍然能夠反映它們的遠近，並提供有意義的梯度。

Wasserstein GANs

既然 Wasserstein 距離有這麼好的性質，我們能將它作為 J_G。在 WGAN 論文中，作者用了一個既有的定理將它變換為下式（證略）：

$$W(p_{\text{data}},\ p_g) = 1/K \cdot \sup_{\|f\|_L \le K} \mathbb{E}_{x\sim p_{\text{data}}}[f(x)] - \mathbb{E}_{x\sim p_g}[f(x)]$$

其中 Lipschitz 連續是一個對函數的條件，限制了函數改變的速度。滿足 Lipschitz 連續就是滿足：

$$|f(x_1) - f(x_2)| \leq K \cdot |x_1 - x_2|$$

Lipschitz 連續等價於 f 的導函數之絕對值不超過 K，限制了一個連續函數的最大局部變動幅度。K 稱為 Lipschitz 常數。因此，Wasserstein 距離的式子的意思就是在函數 f 的 Lipschitz 常數不超過 K 的條件下，對所有滿足條件的 f 取到 $\mathbb{E}_{x\sim p_{\text{data}}}[f(x)] - \mathbb{E}_{x\sim p_g}[f(x)]$ 的上界，再除以 K。f 可以用一個神經網路的輸出 f_θ 近似表達，那麼 Wasserstein 距離又能近似成：

$$K \cdot W(p_{\text{data}},\, p_g) \approx \max_{\theta:\|f_\theta\|_L \leq K} \mathbb{E}_{x\sim p_{\text{data}}}[f_\theta(x)] - \mathbb{E}_{x\sim p_g}[f_\theta(x)]$$

因為神經網路擬合任意函數的強大能力，我們有理由相信這能高度近似 $\sup_{\|f\|_L \leq K}$。為了簡單起見，在 WGAN 論文中作者假設了 $K = 1$，f 必須是 1-Lipschitz。但神經網路訓練時要如何保證這一定發生？一個不太嚴謹但有用的猜測是作 clipping，就是限制神經網路中每個參數不超過某個範圍（$w_i \in [-c,\, c]$），那麼 $\partial f_\theta / \partial x$ 也不會超過某個範圍，Lipschitz 條件得以被滿足。但是這個方法存在缺點，所以後來出現了許多改進版本，例如下一章要講的 WGAN-GP。

至此，我們能構造一個最後一層沒有 activation 的神經網路 f_θ，在 w_i 經過 clipping 的條件下使得：

$$L = \mathbb{E}_{x\sim p_{\text{data}}}[f_\theta(x)] - \mathbb{E}_{x\sim p_g}[f_\theta(x)]$$

盡可能取到最大，此時 $L = J_G$ 就會近似生成分佈與真實分佈的 Wasserstein 距離。而 G 的目標就是最小化這 Wasserstein 距離 L。

接下來就是 WGAN 最精彩的部分了，我們會發現：f_θ 試圖最大化 L，它要想辦法區分真實樣本與生成樣本並最大化這個差值；G 試圖最小化 L，它希望生成的樣本與真實樣本越接近越好，這一樣是一場 minimax game：

$$\min_G \max_{f_\theta \in K\text{-Lipschitz}} \mathbb{E}_{x\sim p_{\text{data}}}[f_\theta(x)] - \mathbb{E}_{x\sim p_g}[f_\theta(x)]$$

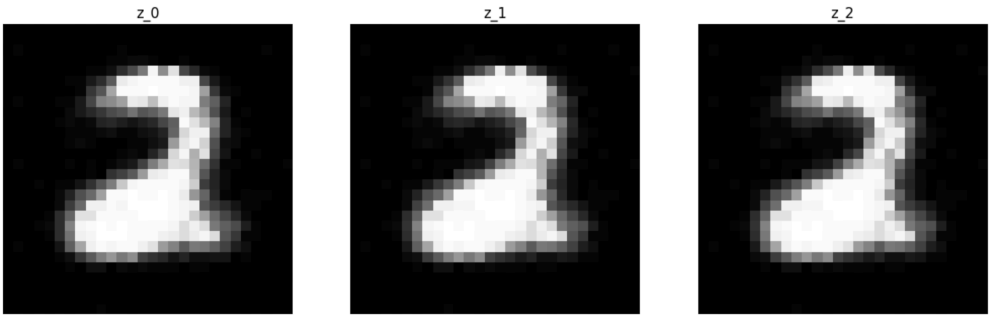
也就是說，我們可以讓衡量距離的迴歸網路 f_θ，取代原本的分類網路成為新的 D。這個 D 不再扮演一個 discriminator，而是扮演 critic 的角色。考慮 L 的第一項與 G 無關，就得到了 WGAN 的兩個 loss，J_D：

$$J_D = \mathbb{E}_{x\sim p_{\text{data}}}[f_\theta(x)] - \mathbb{E}_{x\sim p_g}[f_\theta(x)]$$

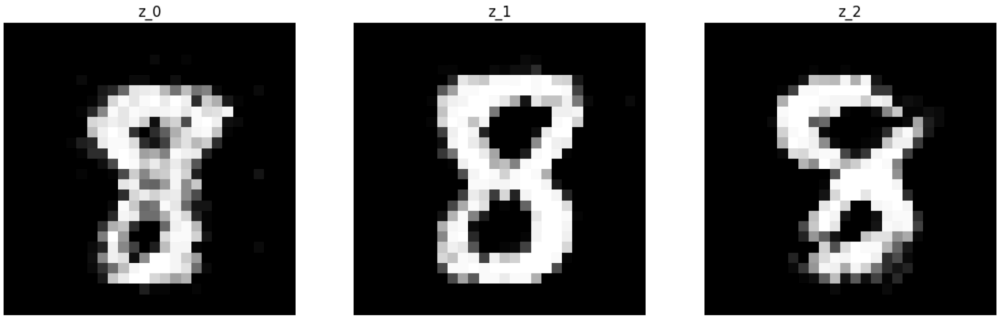
與 J_G：

$$J_G = -\mathbb{E}_{x\sim p_g}[f_\theta(x)]$$

我自己的實作經驗中，Wasserstein GAN 幾乎不曾出現 mode collpase。以下圖一、圖二是對於三筆不同的 noise，GAN 與 Wasserstein GAN 所生成的圖像。



圖一：vanilla GAN 所生成之圖像



圖二：Wasserstein GAN 所生成之圖像