Introduction to GANs

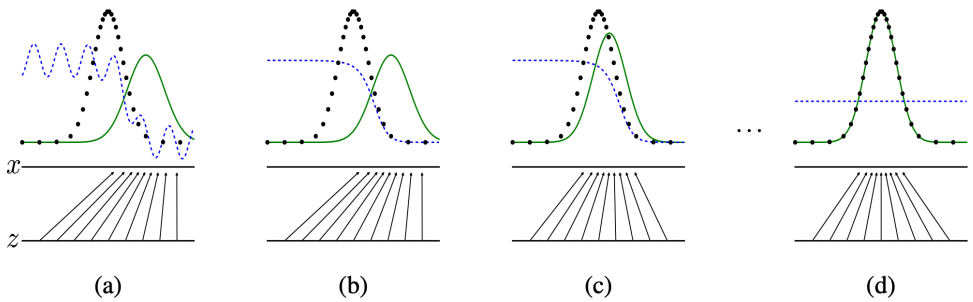
生成對抗網路 (generative adversarial network, GAN) 是非監督式學習的一種方法，通過兩個神經網路相互博弈的方式進行學習。它由一個生成網路 G 與一個判別網路 D 組成，生成網路從潛在空間 (latent space) 中隨機取樣作為輸入，其輸出結果需要盡量模仿訓練集中的真實樣本，而判別網路的輸入則為真實樣本或生成網路的輸出，其目的是將後者從前者中盡可能分辨出來。而生成網路則要盡可能地欺騙判別網路。兩個網路相互對抗、不斷調整參數，最終目的是使判別網路無法判斷生成網路的輸出結果是否真實。

Generative Adversarial Networks

為了學習生成模型對於樣本 x 的機率分佈 $p_g(x)$，我們會引入一個先驗分佈 $p_z(z)$，並且定義一個映射函數 $G(z;\, \theta_g) :\ z \mapsto x$，$G$ 是一個參數為 θ_g 的 MLP。怎麼理解呢？如果 x 是一張高解析度圖片（例如維度為 $1920 \times 1080 = 2073600$ 的向量），我們並不嘗試明確建構其真實的資料分佈 $p(x)$，而是假設這些複雜資料的生成，其實來自某個低維的潛在因素 z（例如 100 維），這些潛變量捕捉了生成邏輯的核心結構。透過訓練 G，我們希望學到一個能從潛變量 z 產生出仿真樣本 x 的模型。自始至終我們是不會理解這些 z 的意義，但理論上強大的 MLP 能擬合任意的函數，所以可以說 GAN 是強行地將 z 映射到我們想要的那些 x 去。
而 $D(x;\, \theta_d)$ 是一個二分類的 MLP，輸出一個值域 $[0,\, 1]$ 的標量，負責判別 x 是來自真實數據還是來自 G（1 代表判別正確，0 代表錯誤）。
論文中提到這是一場雙方對抗的遊戲 (minimax game)，直接給出了 loss function V (value function)：

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

站在 D 的立場，我們總是想要正確地判別，即優化 D 就是試圖最大化之。站在 G 的立場，我們總是想要讓 D 的判別出錯，即優化 G 就是試圖最小化 $\mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$。這等價於在優化 D 和 G 時，必須同時控制前者使 V 盡量大及後者使 V 盡量小。



圖一：GAN 訓練過程

接下來，論文則是提出了 GAN 優化算法：

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial networks.

for number of training iterations **do**

for k steps **do**

Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_z(z)$.

Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.

Update the discriminator by ascending its stochastic gradient:

$$\theta_d \coloneqq \theta_d + \frac{1}{m}\nabla_{\theta_d}\sum_{i=1}^{m}[\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))]$$

end for

Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_z(z)$.

Update the generator by descending its stochastic gradient:

$$\theta_g \coloneqq \theta_g - \frac{1}{m}\nabla_{\theta_g}\sum_{i=1}^{m}\log(1 - D(G(z^{(i)})))$$

end for

優化 D 和 G 時控制前者使 V 盡量大及後者使 V 盡量小，對應到更新 θ_d 和 θ_g 時分別使用 gradient ascent 和 gradient descent。我們注意到 k 可能會讓兩者的訓練「不平等」，但原論文中做的實驗是設 $k = 1$，所以這不是個問題。也可以感覺到，GAN 的訓練難度相對地高，事實上在實際訓練中 GAN 的收斂是非常不穩定的，之後有許多的工作對它進行改進。

接下來是一些理論上的結果。

固定任意的生成器 G（不論好壞），讓 $V(D, G)$ 最大化的最佳判別器 $D^*_G(x)$ 為：

$$D^*_G(x) = p_{\text{data}}(x) \,/\, p_{\text{data}}(x) + p_g(x)$$

【證明】

$$\begin{aligned} V(D, G) &= \int_x p_{\text{data}}(x)\log D(x)dx + \int_z p_z(z)\log(1 - D(G(z)))dz \\ &= \int_x p_{\text{data}}(x)\log D(x) + p_g(x)\log(1 - D(x))dx \end{aligned}$$

其中 $p_{\text{data}}(x)\log D(x) + p_g(x)\log(1 - D(x))$ 是 $y \to a\log y + b\log(1 - y)$ 的形式。通過基礎微積分可以確認，此式在 $a/a + b$，即 $p_{\text{data}(x)} \,/\, p_{\text{data}}(x) + p_g(x)$ 時有最大值。

代入這個約束條件，minimax game 能被寫成：

$$C(G) = V(D_G^*,\ G) = \mathbb{E}_{x\sim p_{\text{data}}}[\log D_G^*(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D_G^*(G(z)))]$$

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

$$= \mathbb{E}_{x\sim p_{\text{data}}}[\log p_{\text{data}}(x) \ / \ (p_{\text{data}}(x) + p_g(x))] + \mathbb{E}_{x\sim p_g}[\log p_g(x) \ / \ (p_{\text{data}}(x) + p_g(x))]$$

當 $p_g = p_{\text{data}}$，也就是 $D_G^*(x) = 1/2$（我們的最終目標）時，V 直接是：

$$\mathbb{E}_{x\sim p_{\text{data}}}[\log 1/2] + \mathbb{E}_{x\sim p_g}[\log 1/2] = -\log 4$$

這是整個損失的全局最小值。

【證明】

$$C(G) = -\log 4 + D_{\text{KL}}(p_{\text{data}}||(p_{\text{data}} + p_g)/2) + D_{\text{KL}}(p_g||(p_{\text{data}} + p_g)/2)$$

$$= -\log 4 + 2 \cdot D_{\text{JS}}(p_{\text{data}}||p_g)$$

由於 $D_{\text{KL}} \geq 0$，$C(G)$ 要取到最小值必須有 $p_g = p_{\text{data}}$。

論文中的這一段其實就是為了證明下面的事情：當 G 完美地學習到數據分佈時，D 將無法判別真偽，此時 loss function V 有最小值 $-\log 4$。

GAN 的原理大致如上。我們注意到它與大部分的 generative models（例如 VAE）不同，它是利用 MLP 的能力強行地將 latent variables z 映射到 x，而不去確實地學習數據分佈 $p(x)$ 的步驟，成功規避了很多複雜數學計算的同時，也增加了訓練的難度。GAN 能應用在圖像的生成、風格轉換、影像修復，甚至是 deepfake 技術，在距離發表十年後的今天，依然是強大的機器學習模型。

圖二：GAN 之數字圖像

圖三：GAN 之藝術作品

圖四：GAN 之影像生成

圖五：GAN 技術的進步