23/11/23

Deep learning

- Recap
- GAN model training
- Does a GAN give us what we want?
- Recap:
  - Generator G is a model that accepts $z \sim p_z(z)$ and generates fake samples $G(z)$
  - Discriminator D is a classifier trained to tell fake samples apart from real

$$D(x) = \begin{cases} 1, & \text{if } x \text{ is real i.e. } x \sim p_{data}(x) \\ 0, & \text{if } x \text{ is fake i.e. } x \sim p_g(x) \text{ where } x = G(z), \; z \sim p_z(z) \end{cases}$$

  - D & G play a game such that G tries to fool D into making an error while D tries to tell real apart from fake as accurately as possible.
  - $V(G, D) = \mathbb{E}_{x \sim p_{data}} \log [D(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - \underline{D(G(z))})]$

- Optimization: Find θ_g and θ_D i.e., the generator and the discriminator parameters such that the following condition is satisfied

$$\min_G \max_D V(G, D) \quad \text{i.e.,}$$

$$\theta_g^*, \theta_D^* = \arg \min_{\theta_g} \max_{\theta_D} V(G, D)$$

- How do we find θ_g & θ_D?
- for # training iterations do

  for k steps do

  (Fix Generator G)

  - Sample m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from $p_z(z)$
  - Draw m samples $\{x^{(1)}, \ldots, x^{(m)}\}$ from $p_{data}(x)$
  - Update θ_d using $\nabla_{\theta_d} \left\{ \frac{1}{m} \sum_{i=1}^{m} \left[\log (D(x^{(i)})) + \log (1 - D(G(z^{(i)}))) \right] \right\}$

  Discriminator update

  $\left(\text{i.e. } \theta_d^{(r)} = \theta_d^{(r-1)} + \eta \nabla_{\theta_d} \ldots \right)$

  $\hookrightarrow$ gradient ascent.

  End for

(o Fix discriminator D.)

Generator updates {
- o Sample m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from $p_z(z)$
- o Update θ_g using $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left[\log\left[1 - D(G(z^{(i)}))\right] \right]$

End for.

- o Does a GAN give us what we want? In other words does p_g converge to p_{data}?

- o Answer: Let's fix our generator $\underline{G}$.

$$V(D) = \mathbb{E}_{x \sim p_{data}} \log D(x) + \mathbb{E}_{z \sim p_z} \log(1 - D(G(z)))$$

$$= \mathbb{E}_{x \sim p_{data}} \log D(x) + \mathbb{E}_{x \sim p_g} \log(1 - D(x))$$

$$= \int p_{data}(x) \cdot \log D(x) + p_g(x) \cdot \log(1 - D(x)) \, dx$$

Std result:

If $f(y) = a \cdot \log(y) + b \cdot \log(1-y)$ for $a, b \in \mathbb{R}^2 \setminus \{0, 0\}$, $y \in [0,1]$, then $f(y)$ attains its maximum when $y = \frac{a}{a+b}$.

$$\Rightarrow D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \quad - (1)$$

- o Let's now work with $D^*(x)$ from (1) and minimize $V(G)$. i.e.

$$G^* = \underset{G}{\arg\min}\ V(G)$$

$$= \underset{G}{\arg\min}\ \mathbb{E}_{x \sim p_{data}} \left[\log\left(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right) \right] + \mathbb{E}_{x \sim p_g} \left[\log\left(\frac{p_g(x)}{p_{data}(x) + p_g(x)} \right) \right]$$

$$= \underset{G}{\arg\min}\ -\log(4) + 2\, JS(p_{data} \| p_g)$$

$\Rightarrow V(G^*) = -\log 4$ when p_{data} and p_g are identical. i.e. the optimal G gives us $p_g = p_{data}$ (in distribution).