DL Lab4

309833023 簡廷羽

1.Introduction of GAN model

1-1: InfoGAN

InfoGAN 使用 information-theoretic extension 加入一般對抗網路中。透過量化一些背景變數如筆劃粗細、光照方位等等，預先學習這些背景變數造成的影響，來增加判讀 test set 資料的準確性。

作者認為生成對抗網路為 Generator（G）與 Discriminator（D）的 MinMax Game

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

利用資訊理論的 Mutual information 的，如果兩個隨機變數 X 跟 Y 是有關連的（非各自獨立）， X 與 Y 的 Mutual information I(X;Y) 就可以理解為：在已經觀察到 Y 值的情況下，X 的不確定性下降的量。

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

代入 GAN 的 MinMax 公式

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I(c;G(z,c))$$

利用 lower bound 做近似

$$\begin{aligned} I(c;G(z,c)) &= H(c) - H(c|G(z,c)) \\ &= \mathbb{E}_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log P(c'|x)]] + H(c) \\ &= \mathbb{E}_{x\sim G(z,c)}[\underbrace{D_{\mathrm{KL}}(P(\cdot|x)\parallel Q(\cdot|x))}_{\geq 0} + \mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \\ &\geq \mathbb{E}_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \end{aligned}$$

$$\begin{aligned} L_I(G,Q) &= E_{c\sim P(c),x\sim G(z,c)}[\log Q(c|x)] + H(c) \\ &= E_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \\ &\leq I(c;G(z,c)) \end{aligned}$$

最後代回原始式子便是其核心演算法

$$\min_{G,Q} \max_D V_{\text{InfoGAN}}(D,G,Q) = V(D,G) - \lambda L_I(G,Q)$$

實驗結果如下，僅將 batch 降低為 50 而其他遵循 default，LOSS 判斷生成好壞

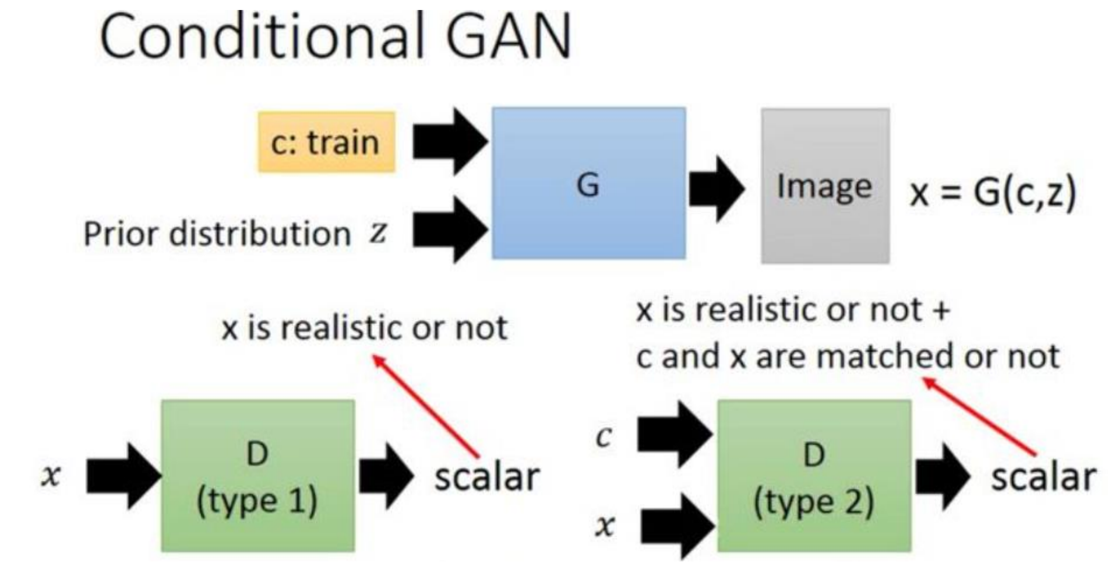
```
[info loss: 1.474881]
[Epoch 49/50] [Batch 933/938] [D loss: 0.244185] [G loss: 0.251602]
[info loss: 1.471616]
[Epoch 49/50] [Batch 934/938] [D loss: 0.062377] [G loss: 0.346832]
[info loss: 1.470984]
[Epoch 49/50] [Batch 935/938] [D loss: 0.206463] [G loss: 0.291382]
[info loss: 1.473173]
[Epoch 49/50] [Batch 936/938] [D loss: 0.089411] [G loss: 0.372573]
[info loss: 1.475475]
[Epoch 49/50] [Batch 937/938] [D loss: 0.324125] [G loss: 0.126295]
[info loss: 1.500670]
```

1.2:cGAN

cGAN 是最早期 GAN 的派生之一，將原先 GAN 中機率的部分全部改為條件機率

$$\max_{D}\{\mathbb{E}_{x\sim P_{data}} \log D(x|y) + \mathbb{E}_{x\sim P_{G}} \log(1-(D(x|y))\}$$

而條件可以為圖片或者是註記，在訓練時只要將條件併入原始數據即可



訓練結果如下，一樣僅將訓練 epoch 降低成 50 次，LOSS 判斷生成好壞

```
[Epoch 49/50] [Batch 912/938] [D loss: 0.224491] [G loss: 0.853703]
[Epoch 49/50] [Batch 913/938] [D loss: 0.252174] [G loss: 0.150555]
[Epoch 49/50] [Batch 914/938] [D loss: 0.236463] [G loss: 0.190448]
[Epoch 49/50] [Batch 915/938] [D loss: 0.160881] [G loss: 0.568119]
[Epoch 49/50] [Batch 916/938] [D loss: 0.173496] [G loss: 0.714114]
[Epoch 49/50] [Batch 917/938] [D loss: 0.167142] [G loss: 0.302079]
[Epoch 49/50] [Batch 918/938] [D loss: 0.162587] [G loss: 0.378507]
[Epoch 49/50] [Batch 919/938] [D loss: 0.161849] [G loss: 0.605132]
[Epoch 49/50] [Batch 920/938] [D loss: 0.150666] [G loss: 0.511965]
[Epoch 49/50] [Batch 921/938] [D loss: 0.163961] [G loss: 0.552911]
[Epoch 49/50] [Batch 922/938] [D loss: 0.150501] [G loss: 0.596812]
[Epoch 49/50] [Batch 923/938] [D loss: 0.179250] [G loss: 0.392891]
[Epoch 49/50] [Batch 924/938] [D loss: 0.158389] [G loss: 0.747249]
[Epoch 49/50] [Batch 925/938] [D loss: 0.146387] [G loss: 0.371635]
[Epoch 49/50] [Batch 926/938] [D loss: 0.159841] [G loss: 0.626175]
[Epoch 49/50] [Batch 927/938] [D loss: 0.169549] [G loss: 0.399431]
[Epoch 49/50] [Batch 928/938] [D loss: 0.185407] [G loss: 0.726126]
[Epoch 49/50] [Batch 929/938] [D loss: 0.204391] [G loss: 0.220829]
[Epoch 49/50] [Batch 930/938] [D loss: 0.124166] [G loss: 0.605306]
[Epoch 49/50] [Batch 931/938] [D loss: 0.175738] [G loss: 0.562132]
[Epoch 49/50] [Batch 932/938] [D loss: 0.158460] [G loss: 0.402236]
[Epoch 49/50] [Batch 933/938] [D loss: 0.172239] [G loss: 0.419690]
[Epoch 49/50] [Batch 934/938] [D loss: 0.163934] [G loss: 0.613591]
[Epoch 49/50] [Batch 935/938] [D loss: 0.172920] [G loss: 0.356517]
[Epoch 49/50] [Batch 936/938] [D loss: 0.161982] [G loss: 0.745131]
[Epoch 49/50] [Batch 937/938] [D loss: 0.187537] [G loss: 0.196418]
```

1-3.ACGAN

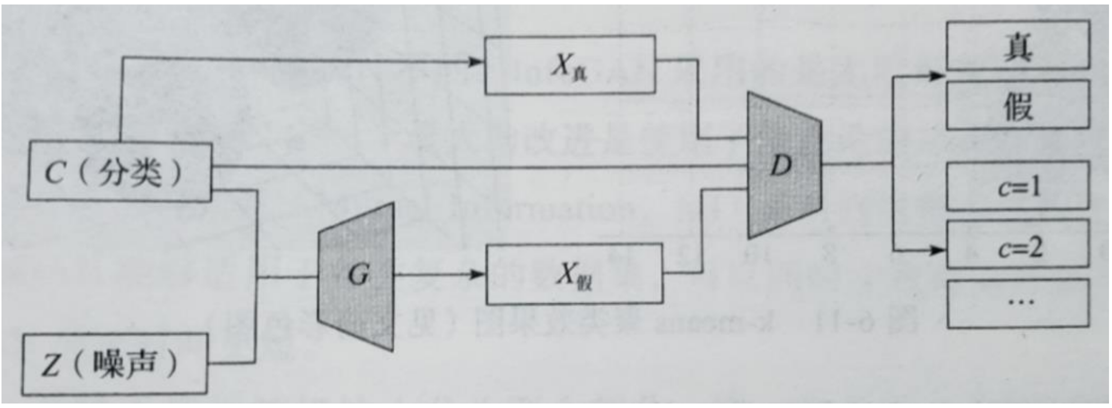
ACGAN 為兩種思路的合併，分別是使用了輔助的標籤信息來增強原始 GAN，對生成器和判別器都使用標籤數據進行訓練，從而實現模型具備產生特定條件數據的能力的 cGAN，以及利用少量標籤，使用判別器或者分類器的尾端重建標籤結構的 SGAN。

Ls 是判斷數據真實與否的代價函數，Lc 則是數據分類準確性的代價函數。

$$L_S = E[\log P(S = real \mid X_{real})] + E[\log P(S = fake \mid X_{fake})]$$

$$L_C = E[\log P(C = c \mid X_{real})] + E[\log P(C = c \mid X_{fake})]$$

詳細結構如下



而訓練結果如下，一樣採用 50 次 epoch，LOSS 判斷生成好壞

```
[Epoch 49/50] [Batch 928/938] [D loss: 1.043941, acc: 98%] [G loss: 1.108768]
[Epoch 49/50] [Batch 929/938] [D loss: 1.089898, acc: 99%] [G loss: 1.082304]
[Epoch 49/50] [Batch 930/938] [D loss: 1.067771, acc: 97%] [G loss: 1.066355]
[Epoch 49/50] [Batch 931/938] [D loss: 1.109168, acc: 97%] [G loss: 1.065504]
[Epoch 49/50] [Batch 932/938] [D loss: 1.102538, acc: 94%] [G loss: 1.107227]
[Epoch 49/50] [Batch 933/938] [D loss: 1.132328, acc: 97%] [G loss: 1.126357]
[Epoch 49/50] [Batch 934/938] [D loss: 1.107282, acc: 97%] [G loss: 1.114182]
[Epoch 49/50] [Batch 935/938] [D loss: 1.116903, acc: 98%] [G loss: 1.080557]
[Epoch 49/50] [Batch 936/938] [D loss: 1.068924, acc: 98%] [G loss: 1.127810]
[Epoch 49/50] [Batch 937/938] [D loss: 1.077462, acc: 100%] [G loss: 1.076680]
```

2.Improvement of GAN

I studied with guide on the Internet, the website had write several options. One is changing log loss to MSE loss, but in sample code it was originally build with MSE loss. Another is giving gradient penalty to avoid mode collapse and improve the stability of GAN. The formula above can inhibit the gradient explosion or disappear.

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

I tried this method but failed to complete the code, error message said my outputs for receiving gradient was inconsistence with the input size.