程式語言-期末專題報告（第 12 組）

1. 成員:

組長：資工 2B 108502571 楊佳峻

組員：資工 2B 108502570 吳榕憲

組員：資工 2B 108502572 吳秉鴻

2. 題目：GAN 手寫數字生成

3. 實作方法：

使用 Python3.8、TensorFlow 套件

網站：https://bit.ly/3lAsvUe

參考論文方法說明:

GAN 可以看成是一個框架，它包含一個 generator 和 discriminator。Generator 負責透過模型輸入，生成能夠媲美真實數據的結果；discriminator 則負責防偽，也就是能辨識 generator 的輸出與真實數據。在兩種模型的對抗下，generator 的輸出可以越來越逼近真實數據，而 discriminator 則越來越精明，能辨識細微的真偽。

4. GAN 生成模型與辨識模型架構

Discriminator: conv2d 作為卷積層取得圖像的特徵值，batch normalization 防止梯度消失並加快訓練速度，使用 leaky relu 作為激勵函數 max pooling 將節點數縮小，重複兩次將圖片從 28*28 變成 7*7 使用 flatten 將資料變為一維，最後縮小變成 1*11 的扁平矩陣以表達圖片答案為各數字的可能性，所使用的批次為 64。

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 64)	640
batch_normalization_2 (Batch	(None, 28, 28, 64)	256
leaky_re_lu_2 (LeakyReLU)	(None, 28, 28, 64)	0
max_pooling2d (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_1 (Conv2D)	(None, 14, 14, 64)	36928
batch_normalization_3 (Batch	(None, 14, 14, 64)	256
leaky_re_lu_3 (LeakyReLU)	(None, 14, 14, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 7, 7, 64)	0
flatten (Flatten)	(None, 3136)	0
dense_1 (Dense)	(None, 128)	401536
batch_normalization_4 (Batch	(None, 128)	512
leaky_re_lu_4 (LeakyReLU)	(None, 128)	0
dense_2 (Dense)	(None, 11)	1419

Generator: input 為 1*10 的 one hot 矩陣及 1*100 的雜訊，一樣使用 batch nomalization 防止梯度消失並使用 leaky relu 作為激勵函數，透過 reshape 將矩陣變為 7*7 的圖形，透過 up sampling 擴大圖片，使用 conv2d transpose(反卷積)取樣，重複兩次可得 28*28 的圖片。

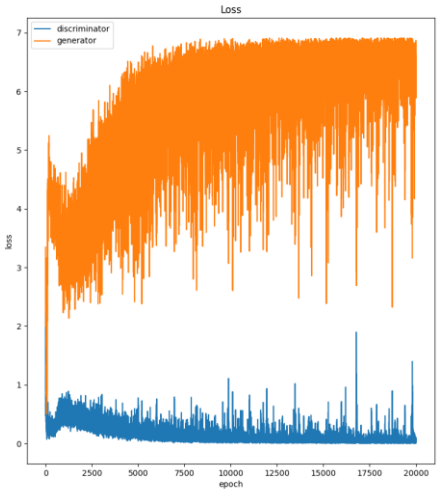
Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None,)]	0
input_1 (InputLayer)	[(None, 100)]	0
tf.one_hot (TFOpLambda)	(None, 10)	0
tf.concat (TFOpLambda)	(None, 110)	0
dense (Dense)	(None, 3136)	348096
batch_normalization (BatchNo	(None, 3136)	12544
leaky_re_lu (LeakyReLU)	(None, 3136)	0
reshape (Reshape)	(None, 7, 7, 64)	0
up_sampling2d (UpSampling2D)	(None, 14, 14, 64)	0
conv2d_transpose (Conv2DTran	(None, 14, 14, 64)	36928
batch_normalization_1 (Batch	(None, 14, 14, 64)	256
leaky_re_lu_1 (LeakyReLU)	(None, 14, 14, 64)	0
up_sampling2d_1 (UpSampling2	(None, 28, 28, 64)	0
conv2d_transpose_1 (Conv2DTr	(None, 28, 28, 1)	577

5. Loss function

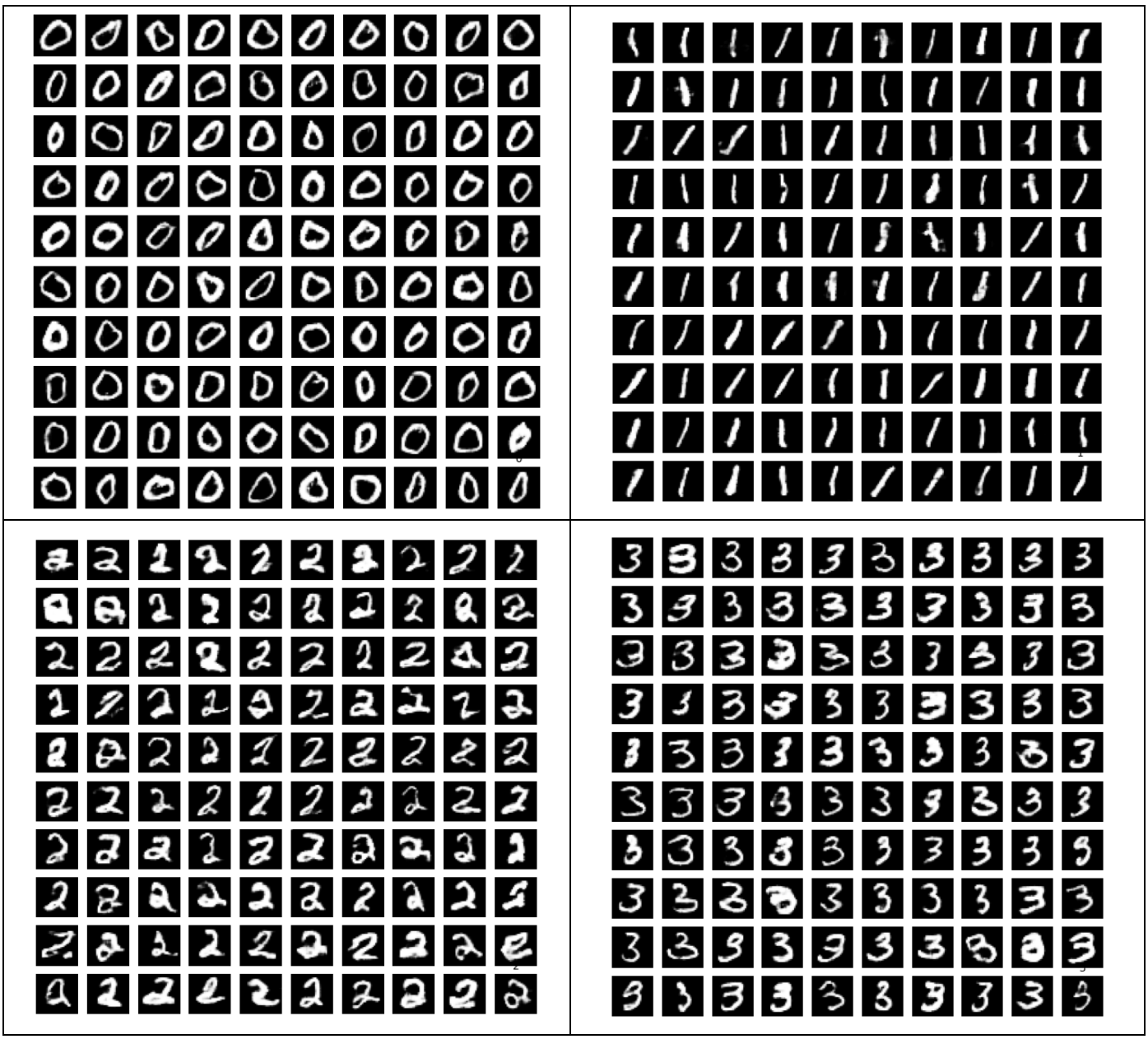
Discriminator: $$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

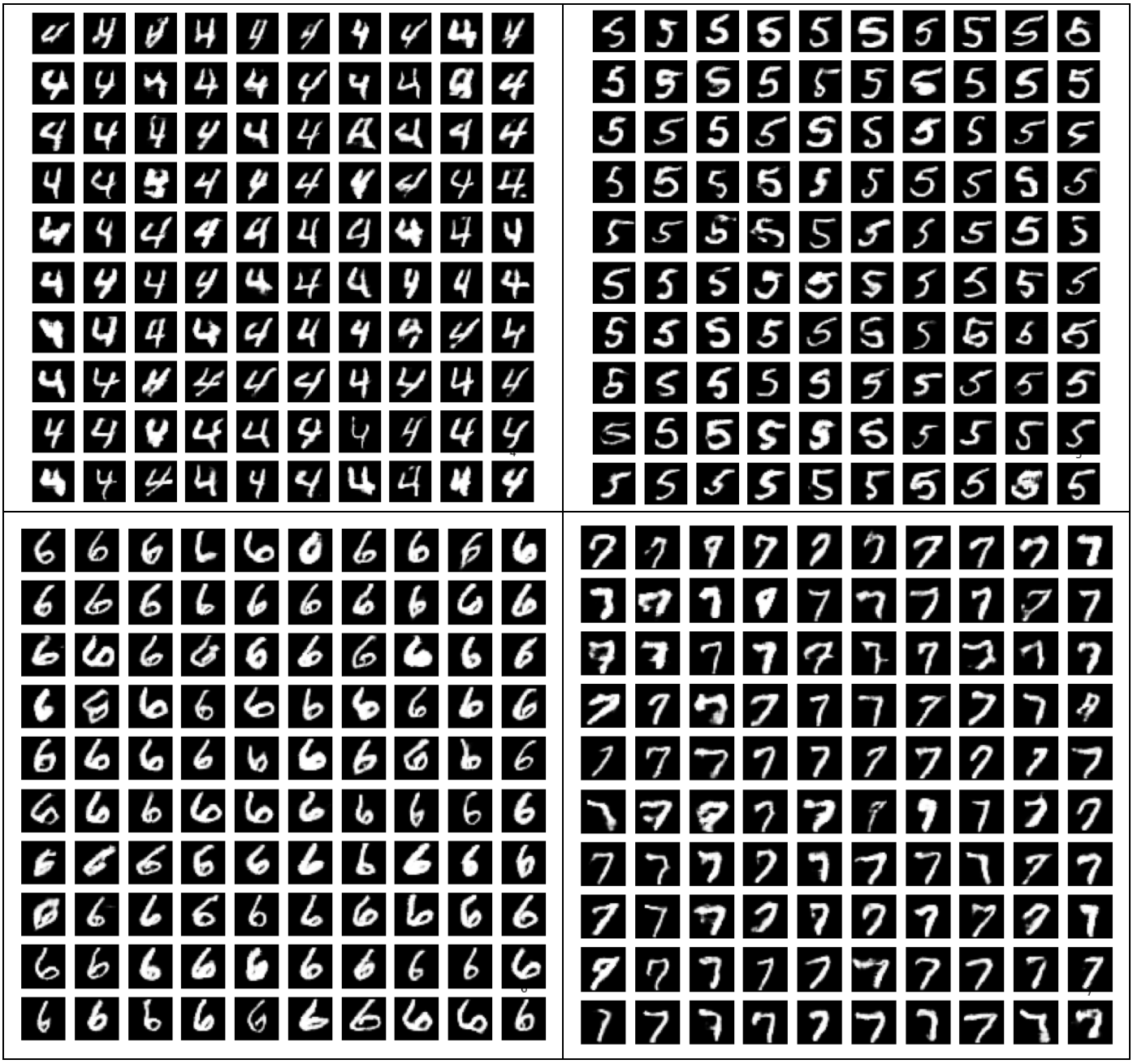
Generator: $$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

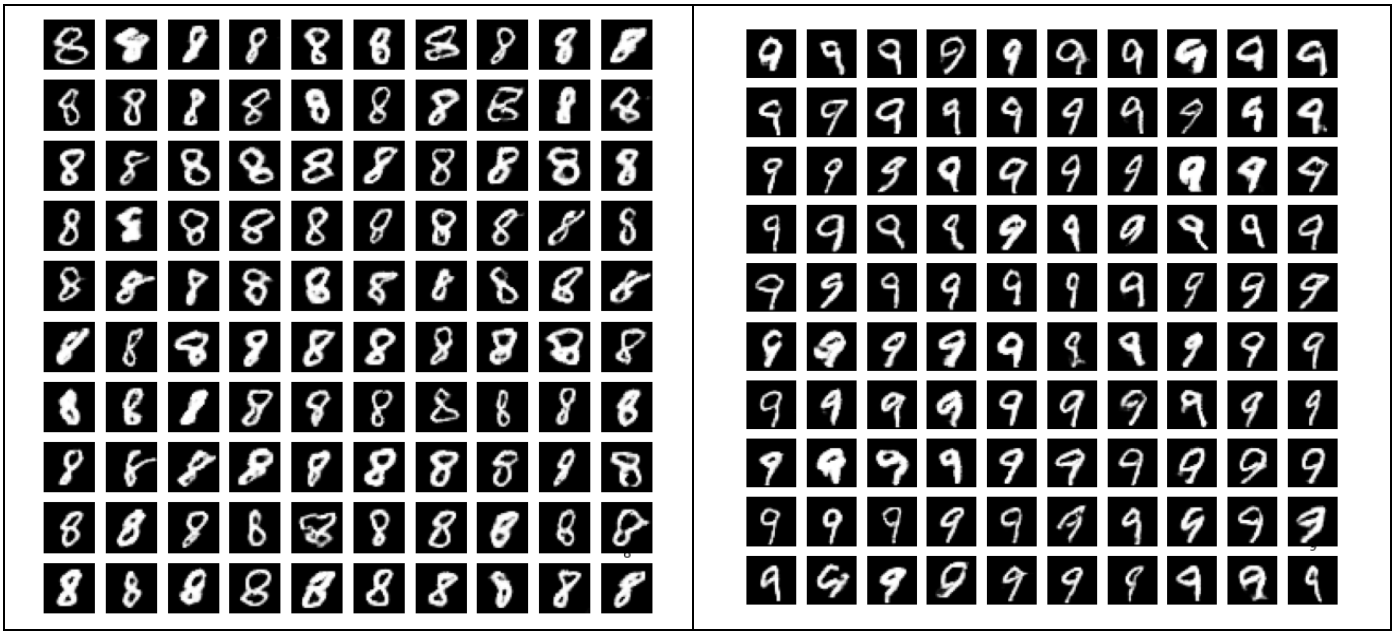
- X: 批次為 i 的 mnist 資料
- G(z) : generator 產生的批次為 i 的資料
- D() : discriminator 給出正確標籤的機率
- discriminator：當輸入資料為 mnist 時使輸出結果為正確標籤，而輸入資料為 generator 產生的資料時，使輸出結果為錯誤標籤
- generator：使 discriminator 判斷其產生的資料時輸出結果為正確標籤
- loss function 曲線圖



6. 生成手寫數字(訓練 10000 次)







7. 個人心得

楊佳竣: Gan 生成對抗網路真的很有趣，理論上，透過兩個神經網路可以互相強化對方，但在實作上，卻要花不少心力在連接他們，不過，成果是好的，的確值得花時間常識，而老師教的 Matlab 畫圖法也與 Python 語法十分雷同，很好上手，謝謝老師這學期的教導，讓我從零開始學會一種程式語言。

吳榕憲: 這次的專題讓我學習如何自學，自己找網路資源學習如何實作生成對抗網路，根據教學打造網路並除錯。也讓我找到 google colab 加速結果生成。此後若要再構造別的深度學習模型必能活用此次的結果。

吳秉鴻: 這次的 GAN 手寫數字生成讓我學到了生成對抗網路的結構，在訓練模型的時候也學到了如何使用 google colab 的 GPU 加速系統，在 demo 的時候，了解到 loss function 在人工智慧模型中佔有非常重要的地位，雖然我們生成的數字還有待加強，但是我覺得這一學期學到的非常多，經過這次的經驗，以後可以避免一些錯誤的方法，使結果更好看一些。