

NTNU Neural Network 2021

Report2 : X-generation

Student id : 40747043S

Name : 劉紹楷

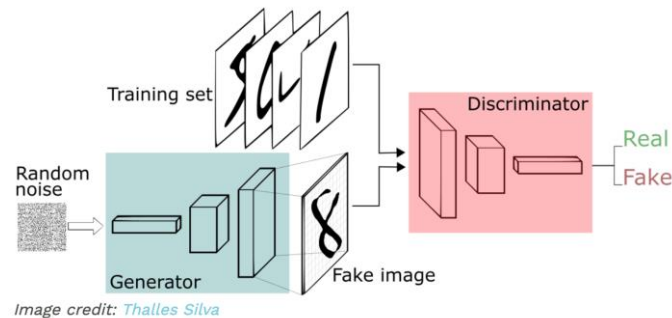
壹、 電腦設備

- 一、CPU : i7-9700F
- 二、GPU : GYGABYTE WINDFORCE 2070 SUPER
- 三、Operating System : Win 10 Enterprise Edition
- 四、Ram : 64GB
- 五、Language : Python 3.8.5
- 六、Framework : Pytorch 1.8.0+cu111、tensorflow

貳、 想法與分析

一、 關於 GAN :

GAN 的流程為，產生和 Training Set 相近的分布資料，利用 Generator 產生假的資料，再跟真的資料一起丟給 Discriminator，讓它判斷是真的還是假的。



- 二、 資料：cifar10 上有資料集，其中有 airplane、automobile、bird、cat、deer、dog、frog、horse、ship、truck 等分類。

參、 資料

一、 資料取得：

參考助教提供的 sample code，使用 cifar10 資料集

肆、 模型

一、 Generator :

```
# Define generator
G = Sequential()

# foundation for 4x4 image
G.add(Dense(256 * 4 * 4, input_dim=100))
G.add(LeakyReLU(alpha=0.2))
G.add(Reshape((4, 4, 256)))

# upsample to 8x8
G.add(Conv2DTranspose(128, (4,4), strides=(2,2), padding='same'))
G.add(LeakyReLU(alpha=0.2))

# upsample to 16x16
G.add(Conv2DTranspose(128, (4,4), strides=(2,2), padding='same'))
G.add(LeakyReLU(alpha=0.2))

# upsample to 32x32
G.add(Conv2DTranspose(128, (4,4), strides=(2,2), padding='same'))
G.add(LeakyReLU(alpha=0.2))

# output layer
G.add(Conv2D(3, (3,3), activation='tanh', padding='same'))

G.summary()
```

二、 Discriminator :

```
# Define discriminator
D = Sequential()

# normal
D.add(Conv2D(64, (3,3), padding='same', input_shape=in_shape))
D.add(LeakyReLU(alpha=0.2))

# downsample
D.add(Conv2D(128, (3,3), strides=(2,2), padding='same'))
D.add(LeakyReLU(alpha=0.2))

# downsample
D.add(Conv2D(128, (3,3), strides=(2,2), padding='same'))
D.add(LeakyReLU(alpha=0.2))

# downsample
D.add(Conv2D(256, (3,3), strides=(2,2), padding='same'))
D.add(LeakyReLU(alpha=0.2))

# classifier
D.add(Flatten())
D.add(Dropout(0.4))
D.add(Dense(1, activation='sigmoid'))
```

三、 其他參數：

1. Input_dim = 7 ([前日收盤價, 當日開盤價, 當日最高價, 當日最低價, 當日收盤價, 當日上引線, 當日下引線])
2. learn rate = 0.0002
3. batch_size = 32

伍、 結果

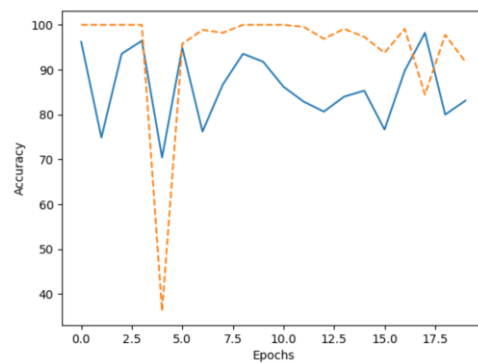
一、 Loss：

1. 參數 1：

Loss function：binary_crossentropy

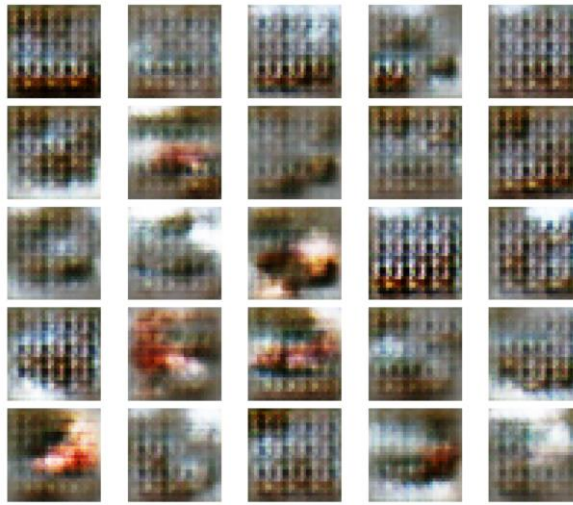
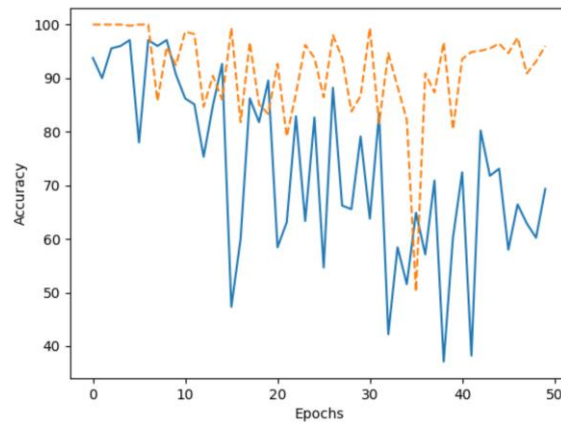
Optimizer：Adam

Epoch：20



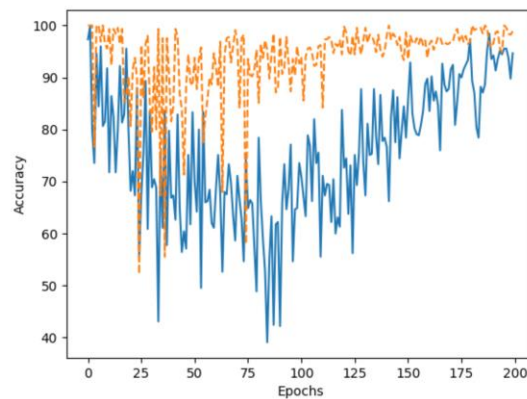
由於圖中看不出甚麼結果，所以試著增加 Epochs

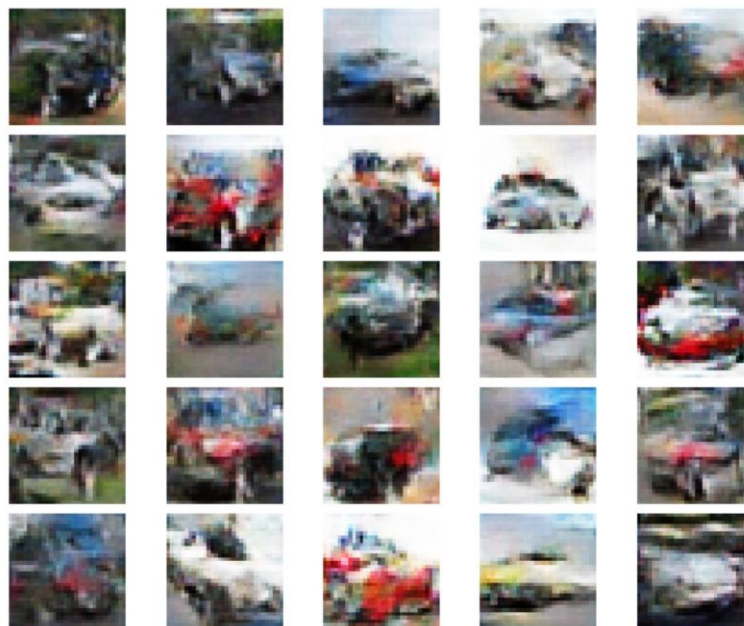
(1) 增加 Epochs 到 50



左下、中間等圖片似乎有東西，所以繼續增加 Epochs

(2) 增加 Epochs 到 200 :



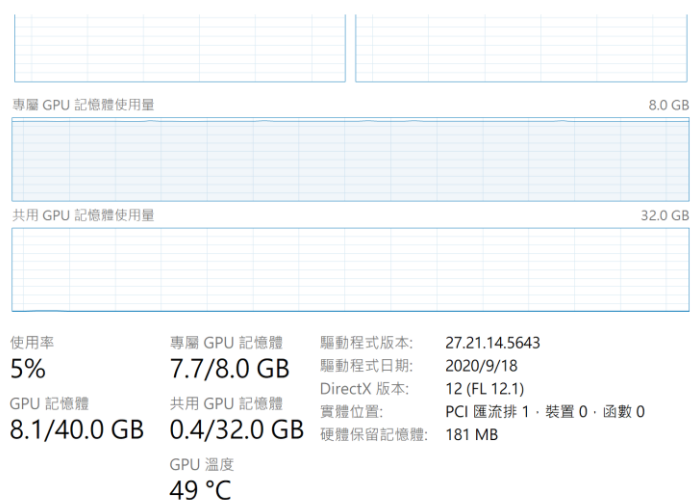


原本還想試試其他的 loss function，但訓練的時間比想像中久。

陸、問題討論與其他

一、GPU RAM：

第一次直接訓練時，GPU RAM 直接不夠用，因為一次讀取 5000 張資料。為了不要在訓練時 RAM 不足，只使用了 450 張。需要學習一下資料太多時如何分次訓練，不讓 RAM 爆炸的



柒、 參考連結

一、 Cifar10 介紹：

<https://www.cs.toronto.edu/~kriz/cifar.html>

二、 Pytorch：

<https://pytorch.org/tutorials/>

<https://pytorch.org/docs/stable/generated/torch.nn.GRU.html>

三、 Paper with code：

<https://paperswithcode.com/paper/conditional-image-synthesis-with-auxiliary>