

# Report

鄭適其  
土木工程學系  
國立陽明交通大學  
ezioatiar@gmail.com

August 16, 2021

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Implementation details</b>	<b>2</b>
2.1	Model and Loss Function . . . . .	2
2.2	Hyperparameters . . . . .	3
<b>3</b>	<b>Results and discussion</b>	<b>3</b>

# 1 Introduction

本次的 Lab 要做的是 conditional GAN，希望可以生成帶有使用者所指定 3D 物件的圖片，可以選擇顏色與形狀兩種屬性，可選擇的顏色有: brown、purple、gray、red、blue、green、cyan、yellow，而形狀有: sphere、cylinder、cube。

## 2 Implementation details

### 2.1 Model and Loss Function

我在本次 Lab 中採用的模型是 WACV2021 中的 multi-hinge GAN (MH-GAN)，基本上可以算是 SAGAN + ACGAN + projection discriminator 的結合體，但是 auxiliary classifier 是使用 hinge loss，並且在訓練 discriminator 時只對真實圖片的分類做訓練，不會訓練生成圖片的分類。以下是其 loss functions 設計：

$$\begin{aligned} L_D &= E_{(x,y) \sim p_d} [\max(0, 1 - D(x, y))] \\ &\quad + E_{z \sim p_z, y \sim p_d} [\max(0, 1 + D(G(z, y), y))] \\ L_G &= -E_{z \sim p_z, y \sim p_d} [D(G(z, y), y)] \end{aligned} \quad (1)$$

Auxiliary classifier 的 loss functions 為：

$$\begin{aligned} L_{\text{Daux}} &= E_{(x,y) \sim p_d} [\max(0, 1 - C_y(x) + C_{\neg y}(x))] \\ L_{\text{Gaux}} &= E_{z \sim p_z, y \sim p_d} [\max(0, 1 - C_y(G(z, y)) + C_{\neg y}(G(z, y)))] \end{aligned} \quad (2)$$

最終得到：

$$\begin{aligned} L_{\text{MH,D}} &= L_D + L_{\text{Daux}} \\ L_{\text{MH,G}} &= L_G + \lambda L_{\text{Gaux}} \end{aligned} \quad (3)$$

其中  $p_d$  為真實 data 分布、 $p_z \sim N(0, 1)$ 、 $C_y(x)$  為正確 label 對應到的 auxiliary classifier 輸出值、 $C_{\neg y}(x)$  為錯誤 labels 對應到的 auxiliary classifier 輸出值中的最大值、 $D$  為 discriminator、 $G$  為 generator。

但是為因應本次 Lab 是 multi-label 任務，因此我自行做出一些調整，使得 performance 進一步提升。首先是 auxiliary classifier 的 loss function 改成：

$$\begin{aligned} L_{\text{Daux}} &= E_{(x,y) \sim p_d} [\max(0, 1 - C(x) \cdot y)] \\ L_{\text{Gaux}} &= E_{z \sim p_z, y \sim p_d} [\max(0, 1 - C(G(z, y)) \cdot y)] \end{aligned} \quad (4)$$

其中  $y$  是 elements 全為-1 或 1 的 vector，當圖片中未出現某 class 時期對應的  $y$  值為-1，反之， $y$  值為 1。

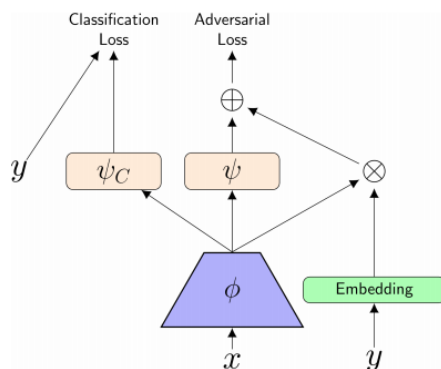


Figure 1: Discriminator 架構

## 2.2 Hyperparameters

- Noise length: 128
- G activation: nn.LeakyReLU(0.1)
- D activation: nn.LeakyReLU(0.1)
- G learning rate: 0.0001
- D learning rate: 0.0004
- Spectral normalization in G: True
- Adam betas: (0, 0.9)
- Epochs: 1000
- Batch size: 64
- $\lambda$  in (3): 1.0
- Max number of channels in G and D: 512
- Location of self-attention in G: At 32x32 feature maps
- Random seed: 87

## 3 Results and discussion

在本次 Lab 中發現，只使用 projection discriminator 似乎難以產生符合指定 condition 的圖片，因此我從 implement SAGAN 轉而去 implement MHGAN，在加入 MHGAN 所提出的 hinge loss auxiliary classifier 後，generator 的 performance 有了很大的提升。

另外在 conditional batch normalization(CBN) 中，原來將 label 的 one hot encoding 轉成 BN 權重的矩陣本來初始化為全 1 矩陣，在我的版本中改成初始化為正交矩陣，實驗證明可以大幅加快訓練速度。



Figure 2: 最終生成結果 (Accuracy = 0.83)