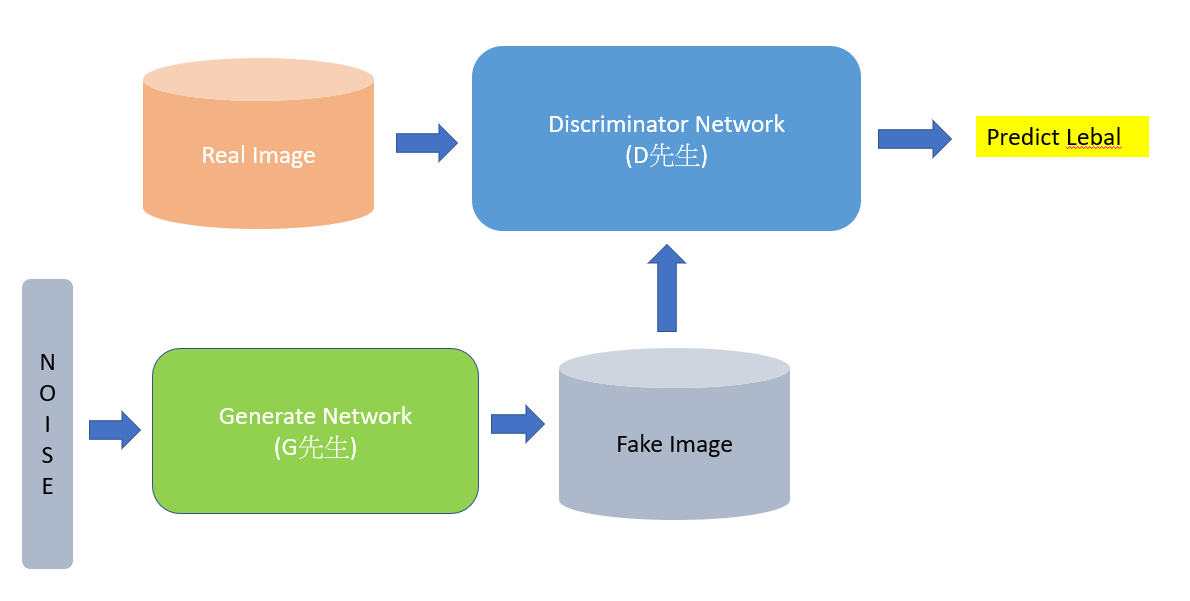
生成對抗網絡

生成對抗網絡(英語：Generative Adversarial Network，簡稱GAN)是[非監督式學習](https://zh.wikipedia.org/wiki/%E9%9D%9E%E7%9B%91%E7%9D%A3%E5%BC%8F%E5%AD%A6%E4%B9%A0)的一種方法，通過讓兩個[神經網路](https://zh.wikipedia.org/wiki/%E4%BA%BA%E5%B7%A5%E7%A5%9E%E7%BB%8F%E7%BD%91%E7%BB%9C)相互[博弈](https://zh.wikipedia.org/wiki/%E5%8D%9A%E5%BC%88%E8%AE%BA)的方式進行學習

原先目的為處理訓練樣本數不足問題,但現在已經被應用在各個領域中

像是無中生有的產生一些不重複的圖片或數據,又或者是優化和修改原本的數據,使其內容物或是風格發生一些轉變

最為原始的GAN如上所述是由兩個神經網路Discriminator Network(鑑別器網路)

和Generator Network (生成器網路)構成

如上圖所示

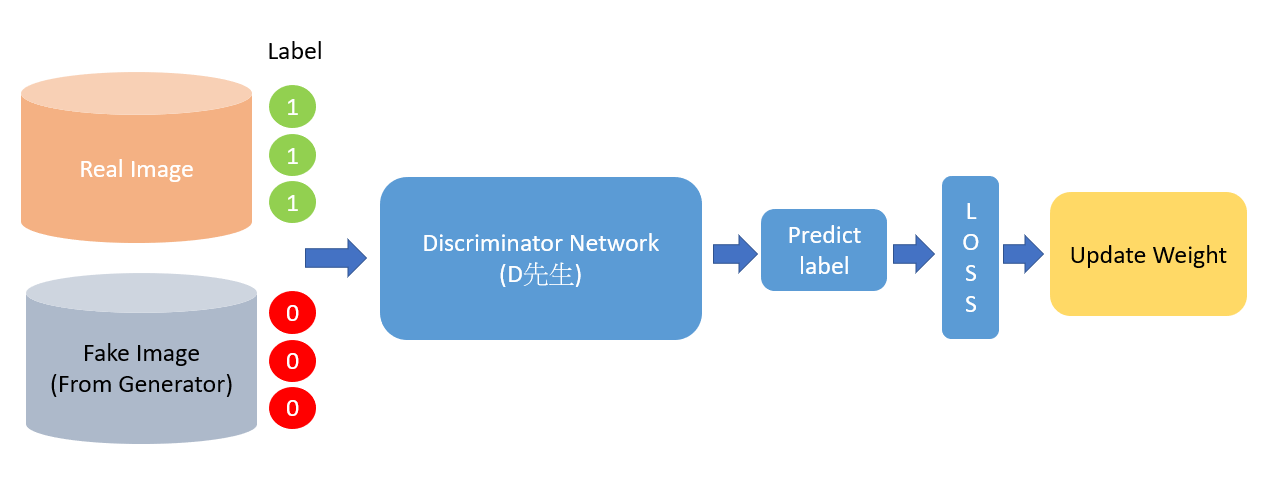
生成器網路是在輸入隨機亂數(noise)後產生虛假資料(fake image)

鑑別器網路是在輸入生成器所作的虛假資料(fake image)後判斷是不是真的圖片

並給予一個0-1之間的分數(predict Label)

至於真實圖片的部分,則是用來訓練鑑別器網路用的

*Discriminator Network(鑑別器網路)*

**

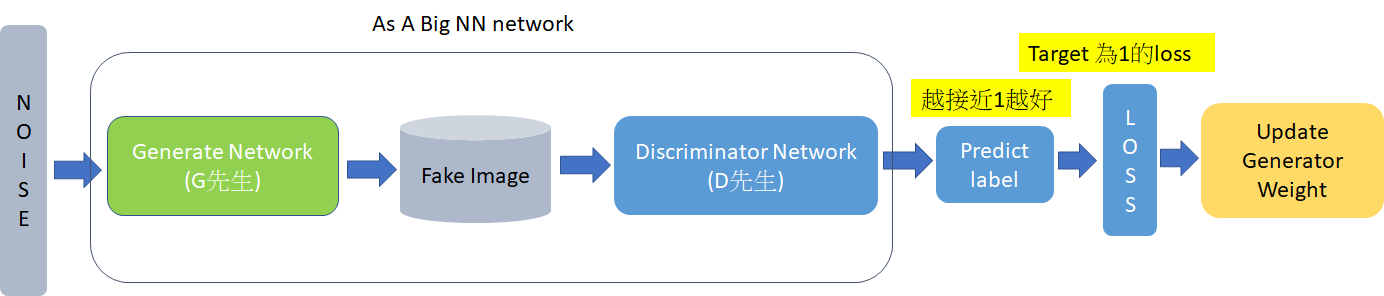
鑑別器網路的訓練

將真實資料放入鑑別器網路的分數

與(1.-虛假資料(藉由生成器網路生成)放入鑑別器網路的分數)

作為Loss值回傳到網路

*Generator Network (生成器網路)*



生成器網路的訓練

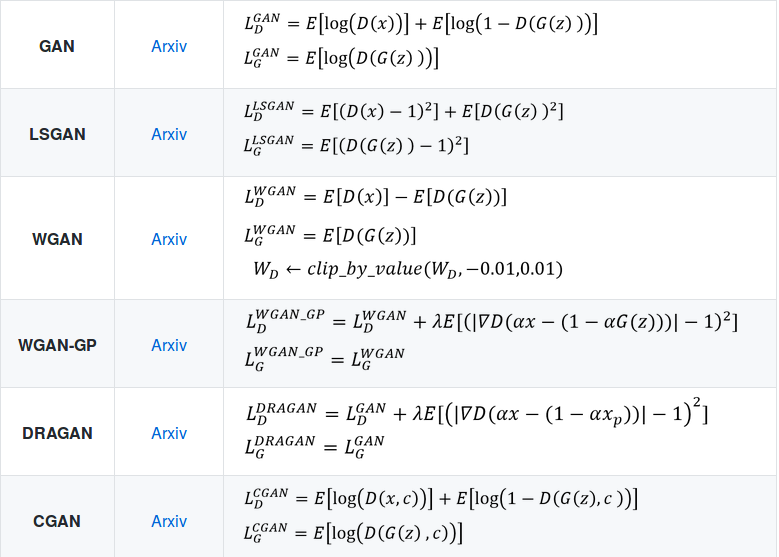
將隨機亂數的noise輸入後產生出的結果放入鑑別器網路得出的分數

作為Loss值回傳到網路

神經網路結構採用全dense層(全連接層)

通常訓練為一筆批次資料丟入生成器網路訓練和鑑別器網路各訓練一次

訓練整理(Loss計算):

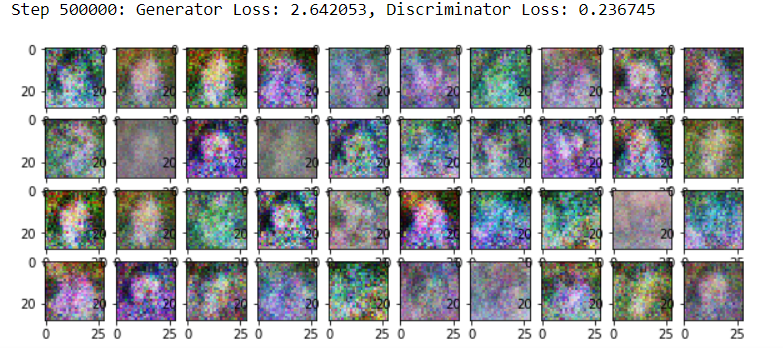


E:加總後取平均(訓練時為批次輸入資料)

D(x):x(真實數據)經過鑑別器網路的結果

G(z):z(隨機亂數)經過生成器網路的結果

執行結果:



在此之前,作為輸入資料的部分圖片



可以看到訓練效果真的不好

不說細節花紋,連輪廓以至於色塊都處理的不是很好

網路上有相當多種GAN,甚至有人做一個叫做GAN Zoo的匯總網站

而大部分的改良中

以兩種改良方式最為普及

其一為神經網路的結構改變

其二為Loss計算方法的改變

其餘參數調整比較少見

如:noise節點數/網路訓練次數/網路層數節點數/批次大小

這類參數幾乎都是需要自行調整

DCGAN

Deep Convolutional Generative Adversarial Network(深度卷積網路層)

簡述:

與GAN不同

G採用反捲積Conv2DTranspose來生成

D採用卷積Conv2D 來鑑別

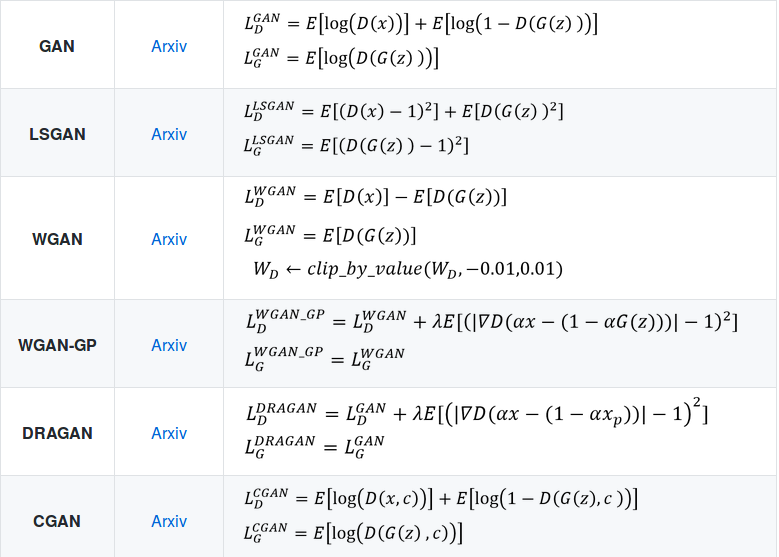
結構

G:全Conv2DTranspose層(反捲積)

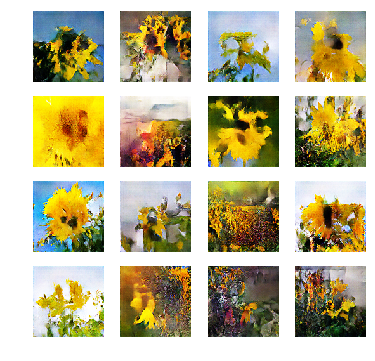
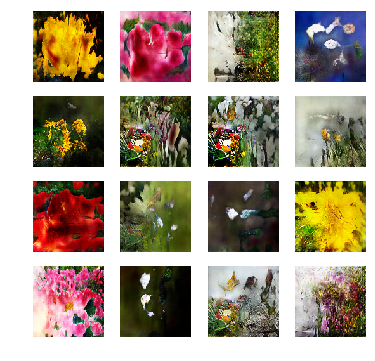
D:全Conv2D(卷積)

LOSS

同GAN



執行結果:



WGAN

Wasserstein GAN

簡述:

LOSS計算方式修改-不進行log計算

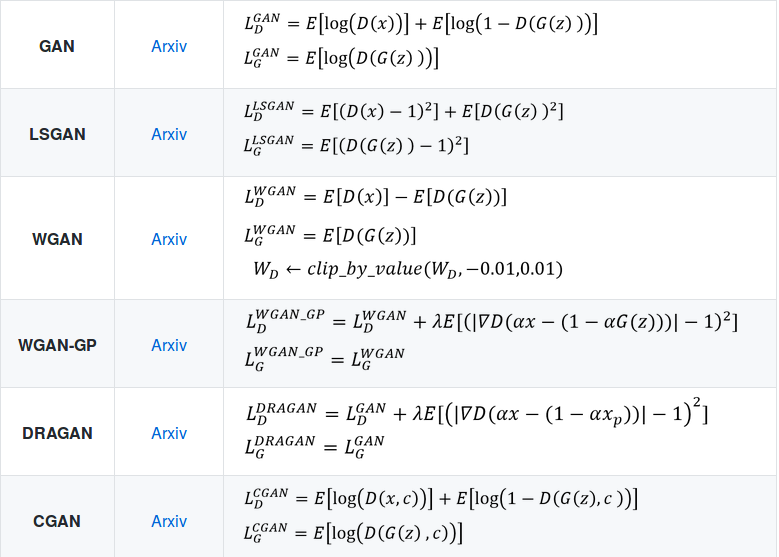
網路權重參數設定上限下限+-c

結構

任意結構 僅需額外加入上下限c值

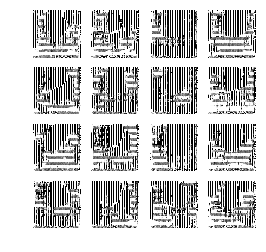
LOSS

不進行log計算



WD為權重的上下限

執行結果:



WGAN\_gp

簡介:

LOSS計算方式變更-加入懲罰梯度

移除WGAN的上限下限

結構

無 不需額外加入上下限c值

LOSS

加入懲罰變數

Ax-(1-aG(z))部分做的是將真實圖片與虛假圖片以隨機比例混合在一起

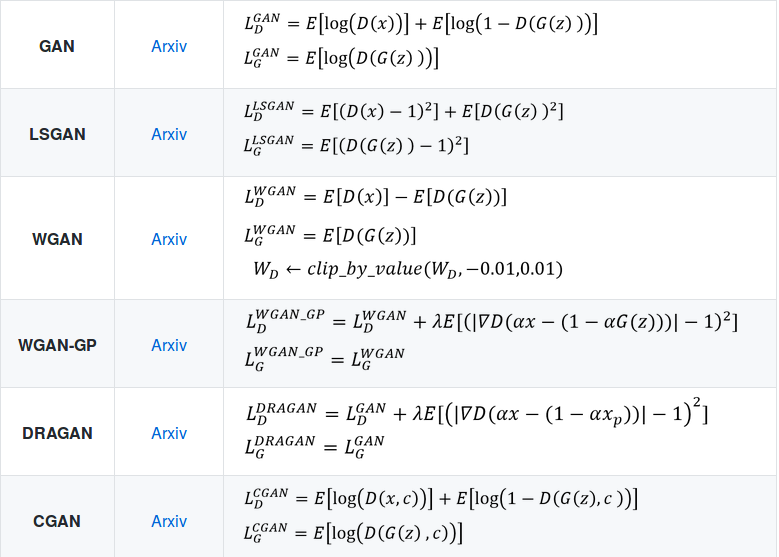
然後進行鑑別器鑑別

得到的分數與隨機比例混合圖進行梯度計算

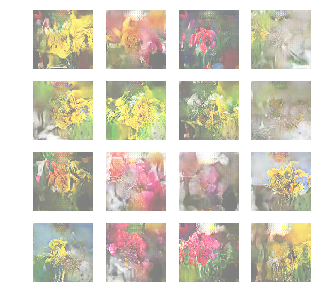
得到的圖片進行平方後加總(針對圖片內容)再平方根

得到的數值-1後平方再加總(針對批次數量)取平均

入 這個符號是指權重 可自行調整



執行結果:

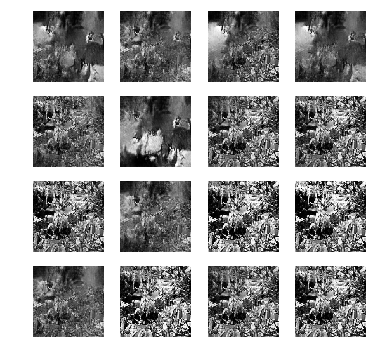


batch\_size

修改有相當差距

使用dcgan

批次大小為3時:



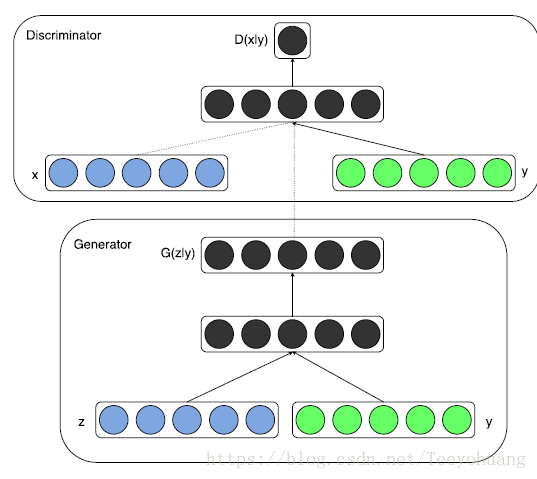
批次大小為64時:



Cgan

Conditional GAN

簡介:



在G的輸入內加入one\_hot後的正確答案

在D的輸入內加入one\_hot後的正確答案

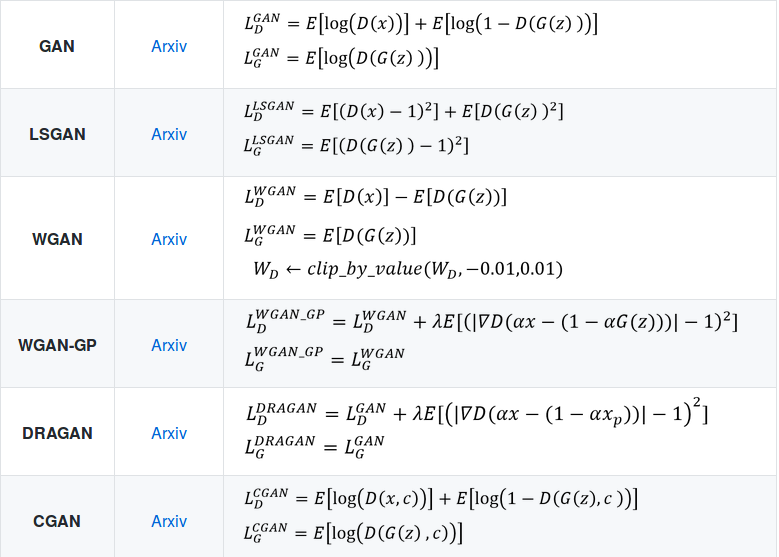
目的是做出可以訓練指定類別的圖片

結構

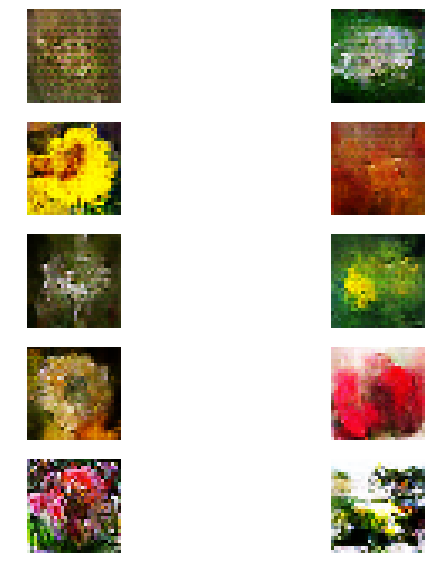
無 僅需在G和D輸入端並行輸入答案

LOSS

採用WGAN\_GP



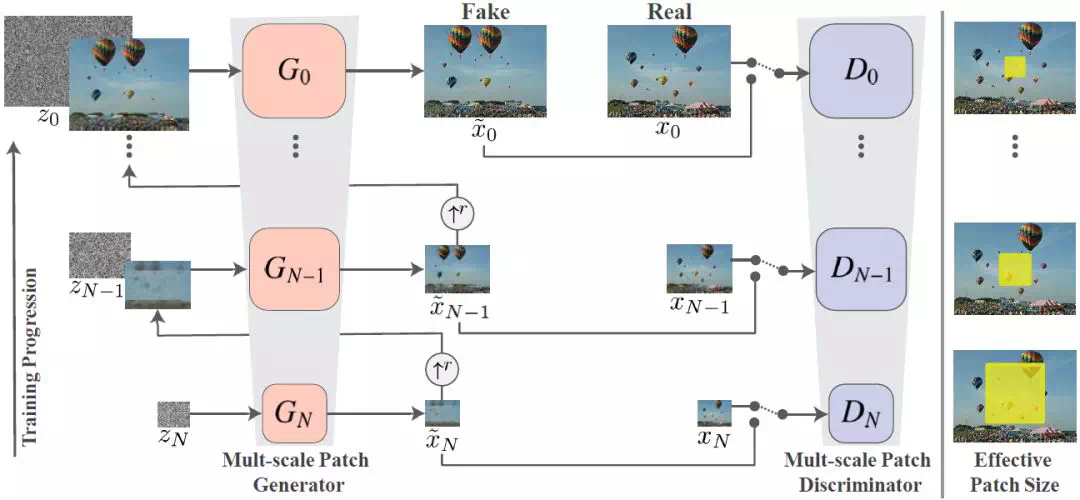
執行結果(cdcgan):



singan

singel gan

簡介:



將圖片分裝成不同大小

每種大小個訓練一個G和D

過程採用殘差學習

結構

G:第一層 輸入:noise 輸出:x大小的圖片

放大(自訂放大比)

第二層 輸入:x\*(自訂放大比)大小的noise圖片+第一層輸出\*(自訂放大比)

輸出:x\*(自訂放大比)大小的圖片

放大(自訂放大比)

第三層 輸入:x\*(自訂放大比)^2大小的noise圖片+第二層輸出\*(自訂放大比)

輸出:x\*(自訂放大比)^2大小的圖片

…

殘差學習:

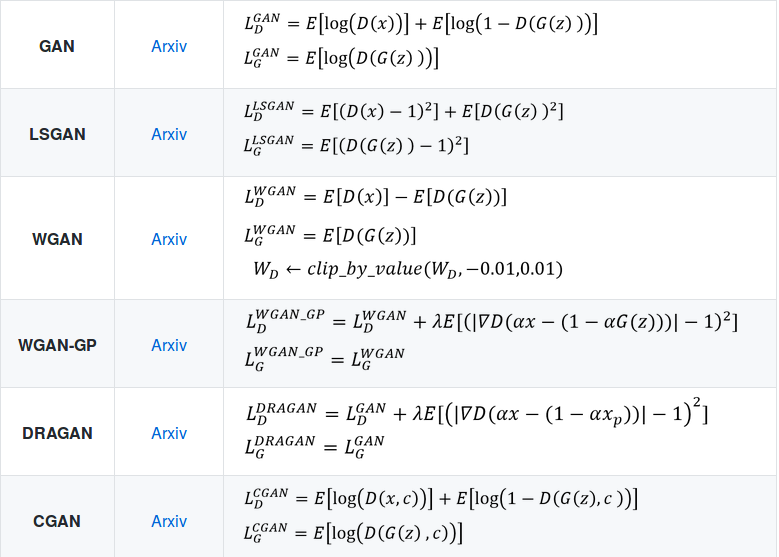
將輸入的兩個變數(大小相同)後加起來進行卷積

卷積後成與輸入圖片相同大小後加進原圖內(第x層輸出) 並輸出

D:就卷積而已

LOSS

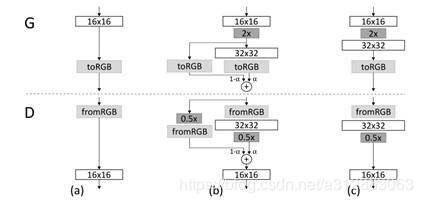
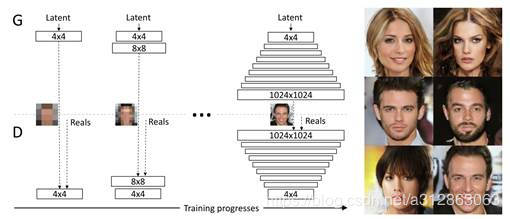
採用WGAN\_GP



執行結果:



PGGAN



與singan相同 同樣是對應不同解析度產生多個G和D

但不同在更高解析度的G和D會沿用上一個解析度的G和D的參數

結構

G:(一樣使用殘差)

第一次(4x4) 輸入:noise(512) 輸出:4x4image 內部:cov2d(name=1)

升維(強制放大2倍)(一個點當4個點用)

第二次(8x8) 輸入:noise(512) 輸出:8x8image 內部:cov2d(name=1)(沿用上層)+cov2d(name=2)

…

如同上述所說 cov2d(name=1)只是使用參數當作新gan的初始值

所以8x8的cov2d(name=1)依舊會被修改

但訓練途中加入結構會導致大變化

這裡採用兩條管路訓練

第一條使用比較少的卷積

第二條則比較多

訓練時先將第一條比重設定為1 第二條為0

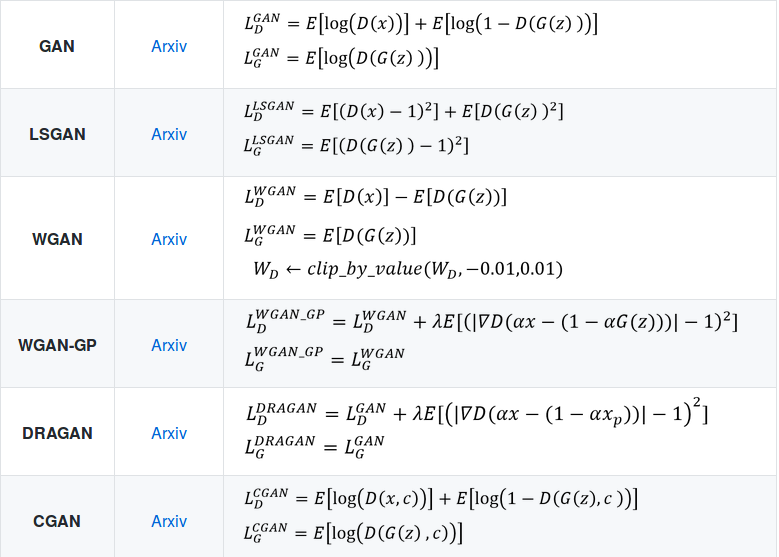
每次訓練時候 將比重慢慢變化

最後將設定第一條比重設定為0 第二條為1

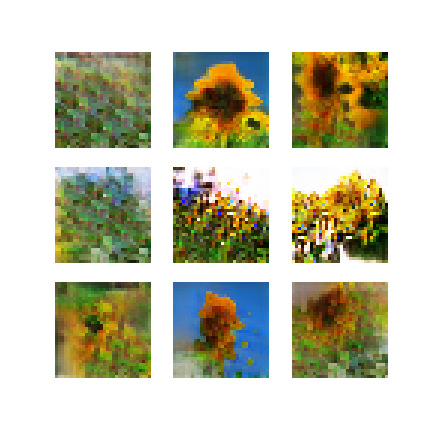
D:就卷積而已

LOSS

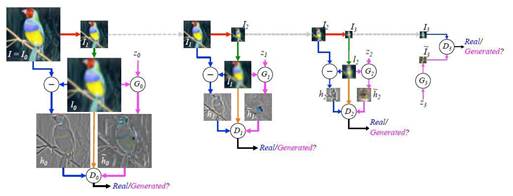
採用WGAN\_GP



執行結果:



Lapgan



與singan相同

採用數層G與D

結構

同singlegan 只是可以輸入多張圖片

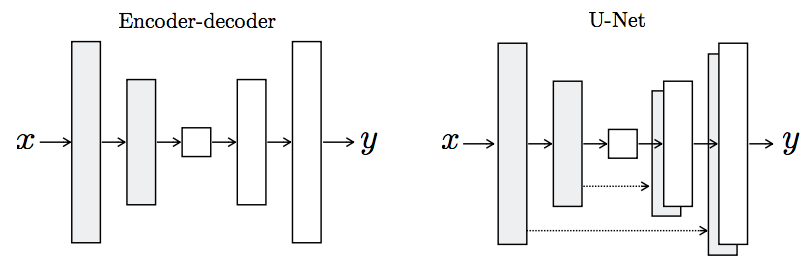
LOSS

pix2pix

採用兩種不同LOSS方法 組合

架構類似autoencoder的U-net

會將之前相同大小的encoder傳入decoder



結構

G:輸入部分採用cgan加入one\_hot後答案

架構採用U\_net結構(如上)

是在autoencoder的decoder上加入encoder的輸入

D:使用Patch\_GAN

類似加入Attention的卷積層分類

LOSS

cgan的loss

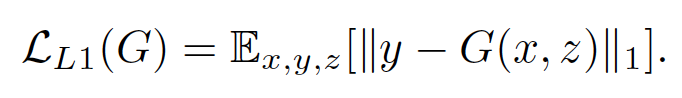
#### 

與GAN無差異,就只是輸入D和G時多幾個固定標籤

x為class標籤

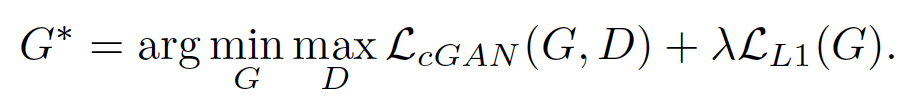
y為真實資料

l1的loss

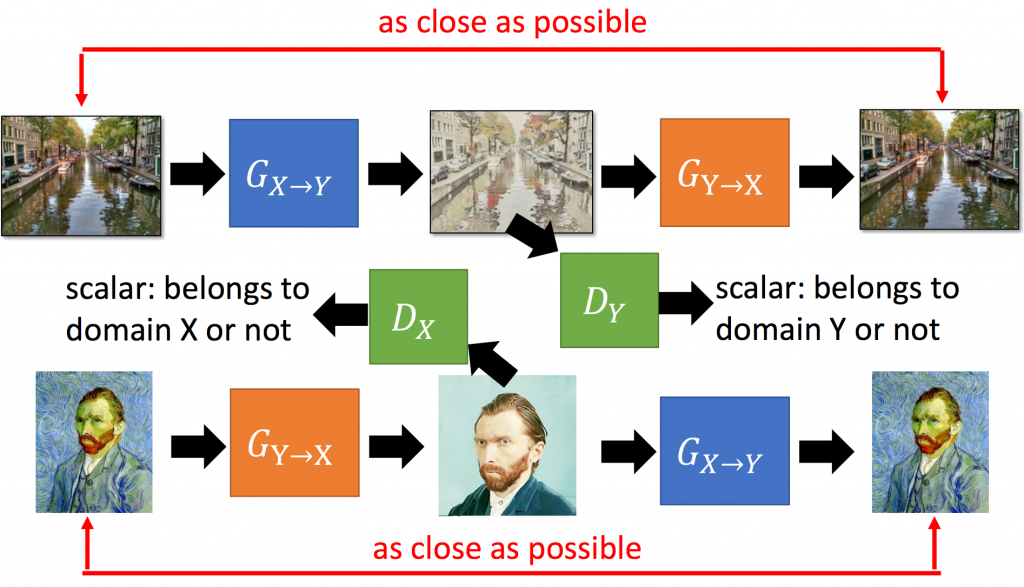


純粹計算假圖與真圖的差異

相加後



CycleGaN



同上類似autoencoder

先編碼encoder 讓原圖轉換為目標風格圖

在解碼decoder 讓轉換後圖轉換回原圖

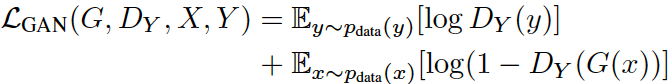
結構

製作兩個迴圈

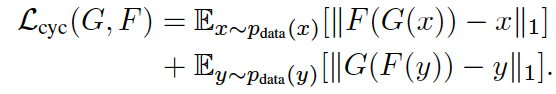
目標是讓一張圖片可以轉換成另一張圖片

也可以轉換回來

LOSS

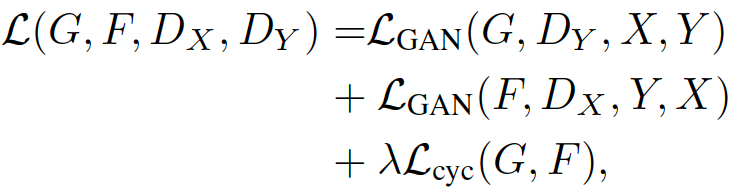


與GAN沒有差異



同pix2pix的L1loss

加總後



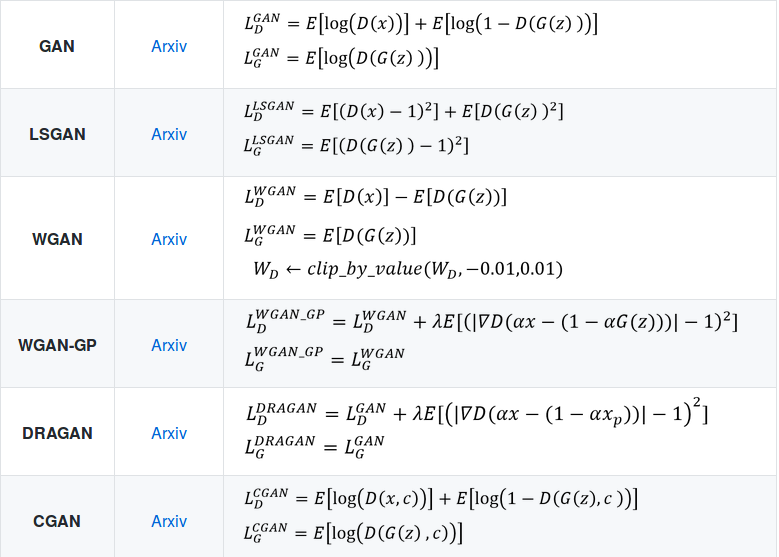
MDGAN/MGGAN

採取多個D或是多個G以求穩定的訓練

LSGAN

結構

LOSS



InfoGAN

### Infogan結構

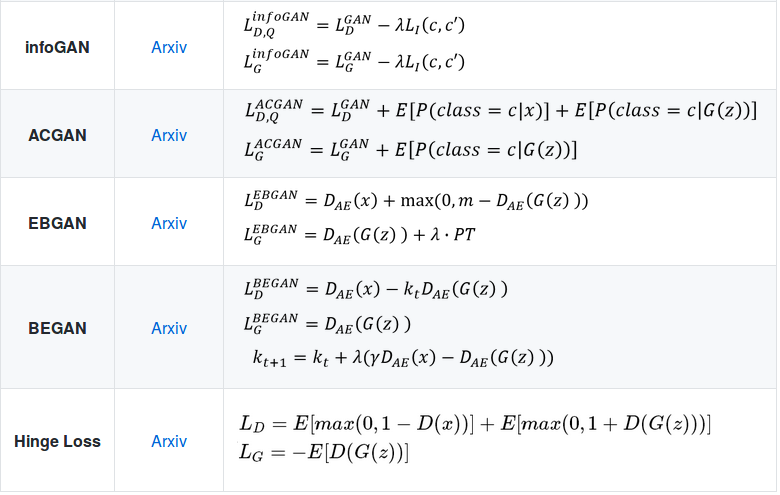
類似cgan

但onehot的答案變成Q(c|x)來尋找

LOSS

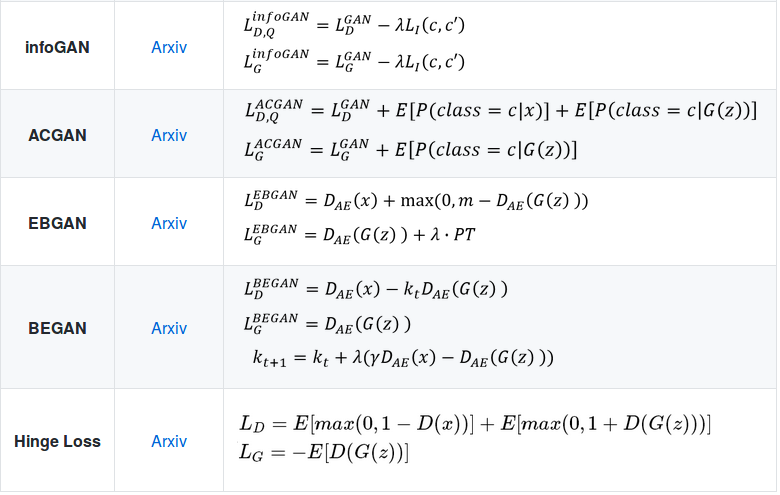
L1同pix2pix的L1

C是指class(標籤)

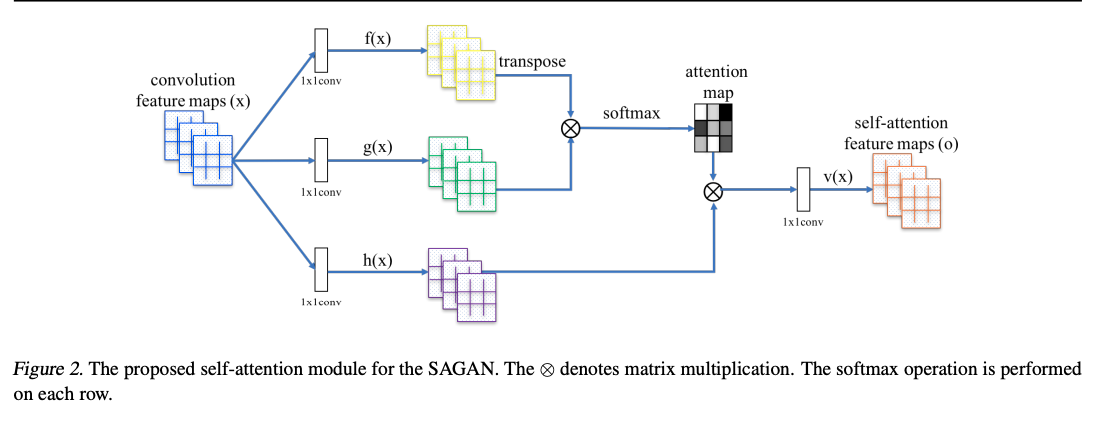


hinge loss

LOSS



SAGAN



f(x) 和 g(x) 所相成得出的結果作為Attention

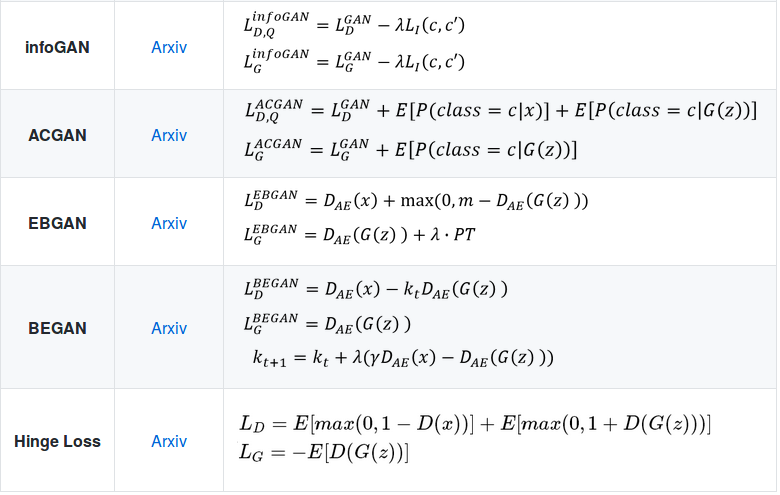
結構

將多張圖片降維後分成兩塊計算後進行softmax計算成Attention(權重)

並用這個權重加入卷積後結果

LOSS

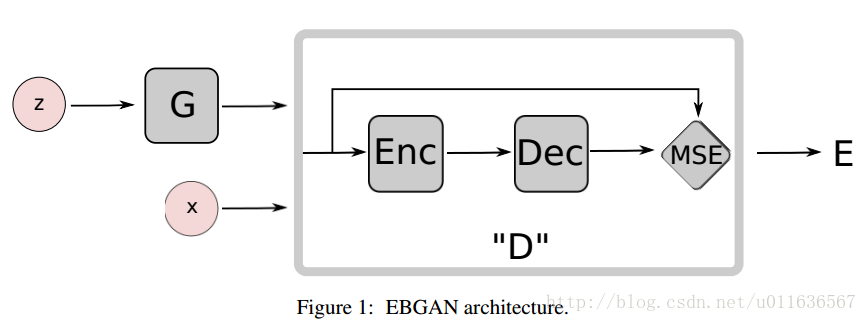
同hinge loss



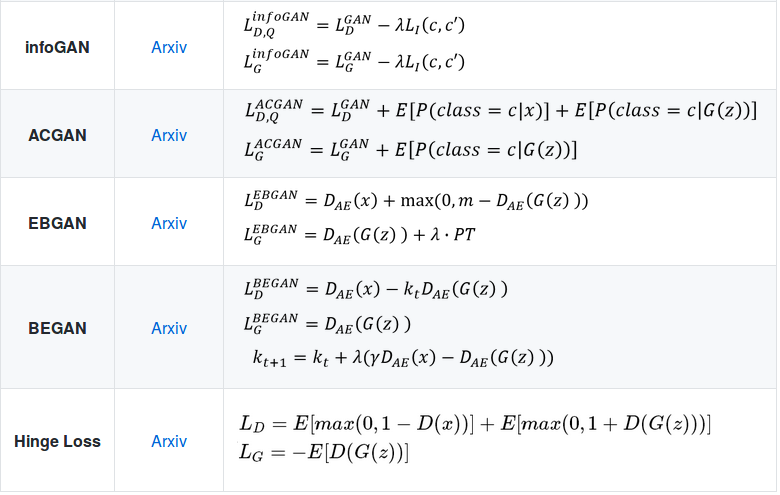
EBGAN

D採用一個Autoencoder

目的是可以防止圖片”紮根”(與原圖一模一樣)

結構

LOSS



M為一個標準 要鑑別出的圖片小於m才能做出貢獻

PT一個權重

BigGAN

採用SA-GAN

採用hinge loss

D,G訓練次數改動

G 的部分加入 class-conditional Batchnorm

D 的部分使用採用cgans Projection

結構

同SA-GAN

G:加入 class-conditional Batchnorm

D:採用cgans Projection(不用concat c 而用inner product)

LOSS

同hinge loss

