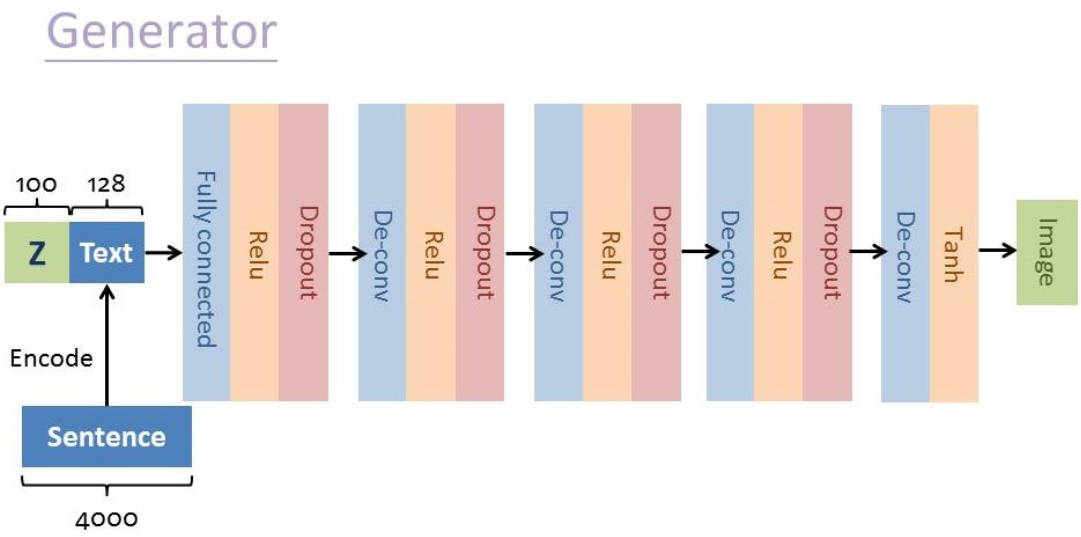
HW3 Report

R04922108, R04922098, R04922086, R04922065

1. Environment

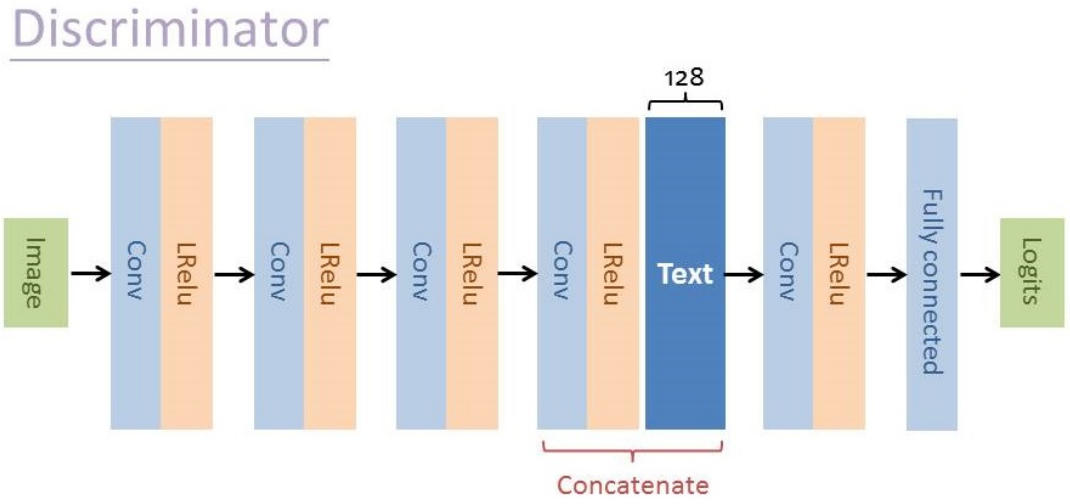
OS	=	Ubuntu 14.04	CPU	=	i76700
Lib	=	Tensorflow 1.0	GPU	=	1080
Python	=	2.7	CUDA	=	8.0

2. Model description



上圖是 Generator 的架構圖，整體目標是使用簡單的機率分佈(Normal)經由多層的 De-Conv Layer，使其逼近於真實圖像的資料分佈。

因此一開始會先 Sample 100 維的亂數 Z，並且疊合 Text Vector，產生 228 維的輸入。接著便進行 De-Conv，也就是與之後的 Discriminator 進行對稱，在 Generator 時，我們會使用 Relu 作為 Active Function，也有加上 Dropout。最後生成假的 Image。



上圖是 Discriminator 的架構，整體目標是用來判別圖像跟文字的關係，除了單

純辨別是生成的圖還是從真實資料取得的圖之外，還有是否真的對應到輸入的文字，為此我們分為四個種類(Real Image, Right Text)、(Fake Image, Right Text)、(Real Image, Wrong Text)、(Wrong Image, Right Text)。接著求出的 Logits 會再經過 sigmoid_cross_entropy_with_logits，使其能趨近於正確判斷，也就是第一種類型要趨近於 1，其他趨近於 0。簡而言之要讓 Discriminator 能夠正確的判別出對的 Image 跟對的文字敘述。我們使用的是 wgan，所以有使用 weight clipping，我們的 objective function 為(real_image_logit – fake_image_logit – wrong_caption_logit – wrong_image_logit)

3. How do you improve your performance

這次作業一開始我們單純只照著 paper 上的作法去 train 一個 gan, paper 上 discriminator 之 loss 定義為 log(sr) + (log(1−sw) + log(1−sf))/2，這個式子單純只考慮到不對應的 caption 讓他的值越小越好，但照理而言，不對應的圖片 discriminator 的值也應該要越小越好，因此我們將 loss 改為 log(sr) + (log(1-sw) + log(1-sf) + log(1-swi))/3，但 train 下去之後發現，得出來的圖片雖然很真實但確沒有與 input caption 很吻合，我們猜想可能原因為除以三的那項造成錯誤對應的 caption 給予 discriminator 的 loss 與 fake image 給予的 loss 差太多，造成更新參數時只要往產生很像真的 image 方向就好，理論上來說這個 loss train 到極限還是能夠學到要與 caption 吻合這件事，但時間應該非常久，因此我們將除以三的那個 term 拿掉，利用新的 loss 後，我們的 model 產生出來的 image 也就更與 caption 吻合了，但在使用這個原本 paper 上的 model 架構時，有一個很大的問題就是有 mode collapse 發生, model 一直不斷重複產生相同的圖案，因此我們後來將整個 model 改成 wgan model，也成功解決了 mode collapse 的問題

GAN Results(mode collaspe)

4. Experiment settings and observation

我們使用的 batch size 為 64, learning rate 為 0.00005，因為使用了 wgan 所以要做 weight clipping，我們是照著 wgan paper 上的作法將 weight clip 到-0.01 與 0.01 之間，最後 optimization 方式使用 rmscrop，每一種 model 都 train 了 70000epoch，以下為實驗結果圖.

WGAN Results

Blue hair blue eyes

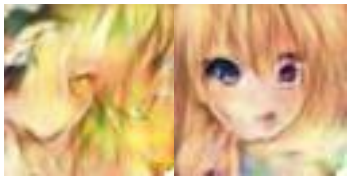
Blonde hair pink eyes

Black hair black eyes

Red hair black eyes

使用 wgan 後雖然解決了 mode collapse 的問題，但我們從上面的結果可以看到，其實 image 和 caption 還是不太吻合，尤其是眼睛的部分，我們思考可能原因為，由於我們 input caption 的取法是非常固定的 pattern, color hair color eyes 這種取法，會不會導致 network 只學到第一個 color 就放在頭髮上第二個 color 就放在眼睛上，而沒有考慮到 hair 跟 eyes 的位置呢，於是我們做了實驗測試 network 吃進 color eyes color hair 的結果

Results



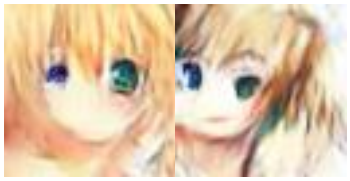
Blonde eyes blue hair

Black eyes blonde hair

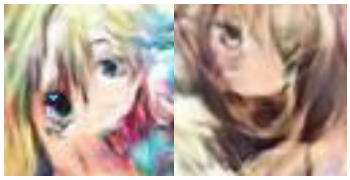
可以看到真的產生這種情況，針對這個問題，我們解決方法為，取 input caption 時，隨機的將前後調換，讓 input caption 有 color hair color eyes 以及 color eyes color hair 等不同的組合，但結果上來看在眼睛的部分並沒有解決 caption 對應的問題(如下圖 Blonde hair pink eyes)，雖然稍微改善了在 hair eyes 順序對調產生的問題，但容易產生不太 real 的圖

Results

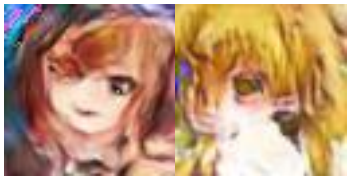
Blue hair blue eyes



Blonde hair pink eyes



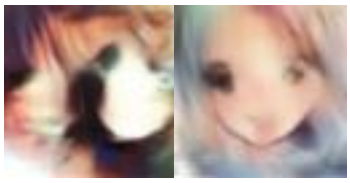
Blonde eyes blue hair



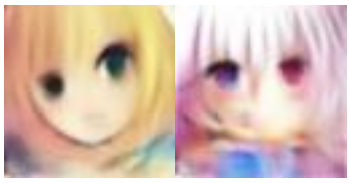
Black eyes blonde hair

另外，我們也實驗了加入 dropout 的結果，但結果看起來並不好，

Results



Blue hair blue eyes



Blonde hair pink eyes

雖然我們想了許多種改善的方法，但似乎都沒有很有效，因此我們最終還是決定使用原本的 wgan model

5. Team division

R04922108	R04922098	R04922086	R04922065
WGAN Report	Report	Report	Report