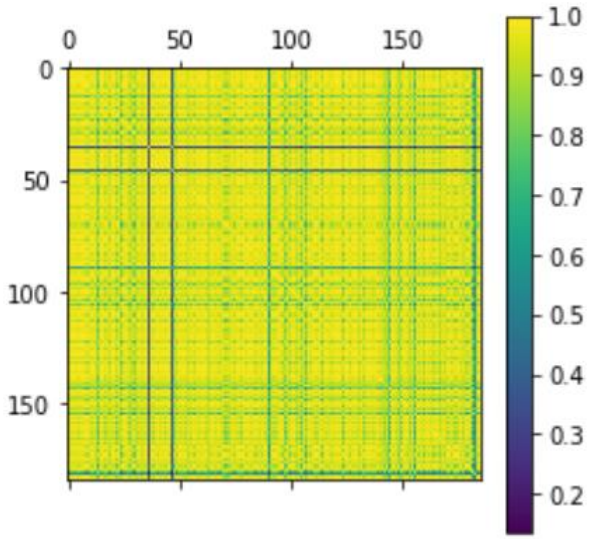
第一題:

i.畫出 correlation matrix:



可以明顯看出，各個國家之間都有蠻高的 correlation，其實也蠻正常的，蒐集資料時是疫情爆發初期，大部分國家都是 increasing，甚至沒有負相關的國家。

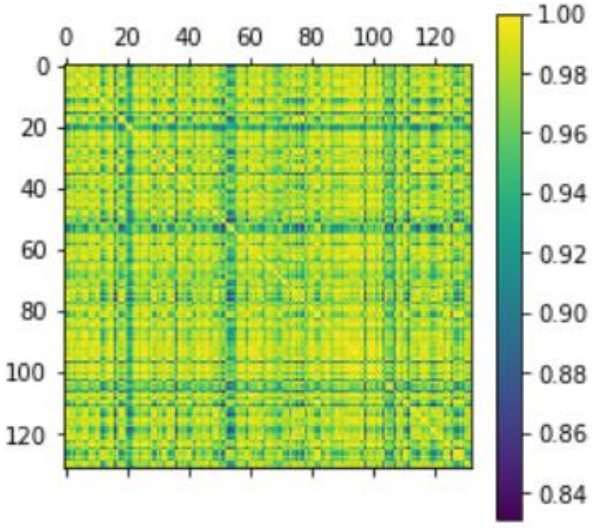
ii.把高 correlation 的國家們都蒐集在一起:

其實這題有個小 bug，讀了一下 flow chart 以及題目的敘述；又讀了一下討論區中助教提到取高 correlation 的目的。發現題目的 flow chart 沒辦法達到目的；舉個簡單的例子:

A = (1, 1, 2, 2), B = (1, 1, 2, 2), C = (1, -1, 1, -1), D = (1, -1, 1, -1)，A、B 是正相關、C、D 也是正相關；然而 A、C 卻是 0 相關啊!

因此我換了 flow chart 的邏輯: 以'US'為基準，找尋跟'US'的 correlation>0.95 的國家們，進行 LSTM 跟 RNN。

下圖是選出來的那些國家的 correlation matrix。總共 132 個國家。



iii.創建 RNN 模型，並畫出 loss, acc

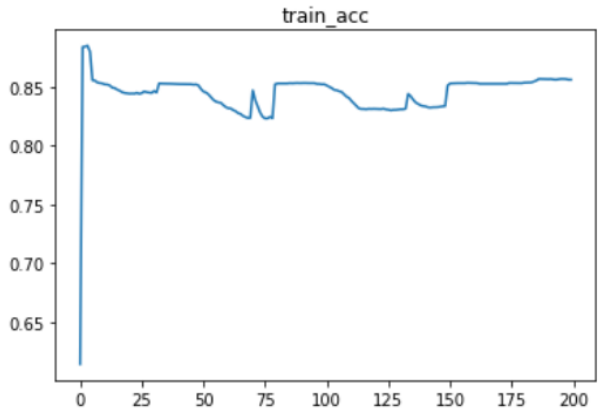
把前面 100 個國家當作 training、後面 32 個國家當作 testing:

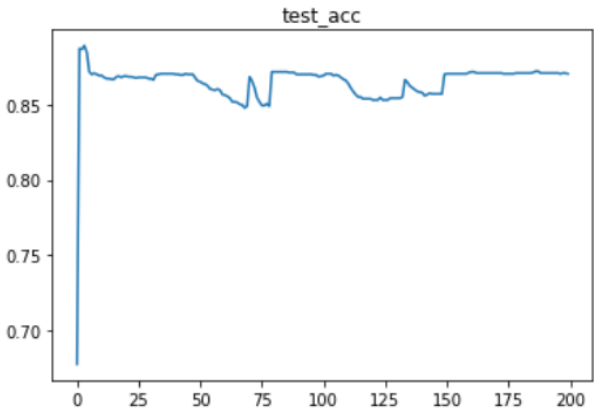
L = 5

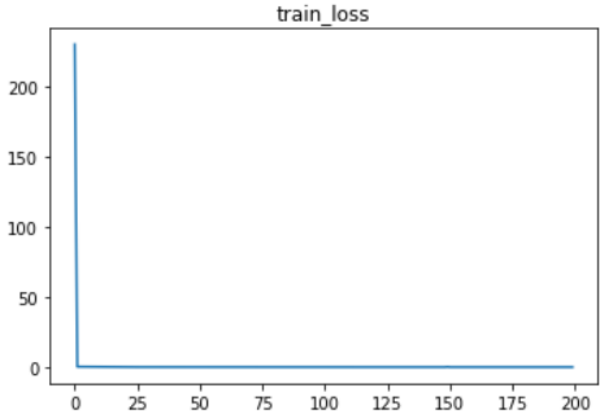
batch_size = 77

num_epochs = 10000

learning_rate = 0.0005







其實仔細看 acc 會發現它其實是往下掉的:
我將第二個以及最後一個 epoch 的 confusion matrix 列印出來:
第二個 epoch 的 train \ test 的 confusion matrix:

```
[[4729 2971]
 [   0    0]]
```

```
[[1670  794]
 [   0    0]]
```

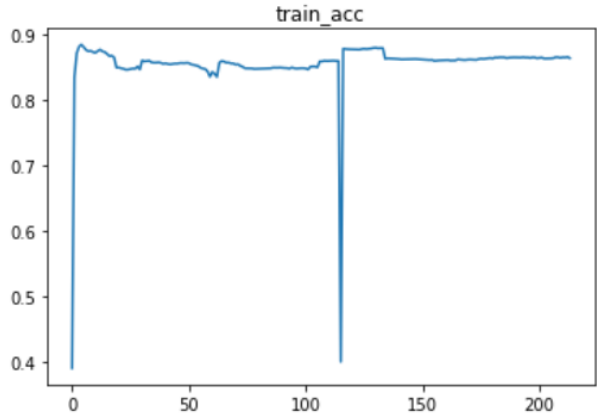
最後一個 epoch 的 train \ test 的 confusion matrix:

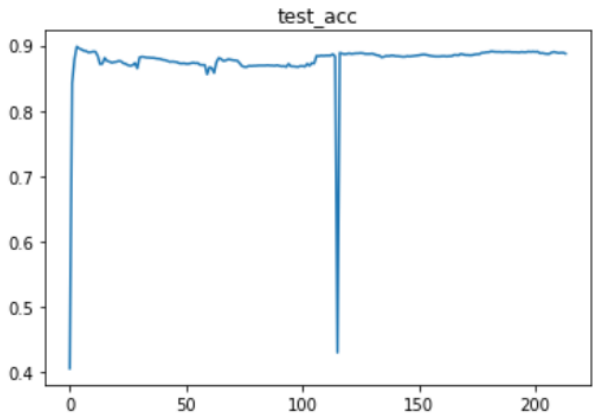
```
[[4074  453]
 [ 655 2518]]
```

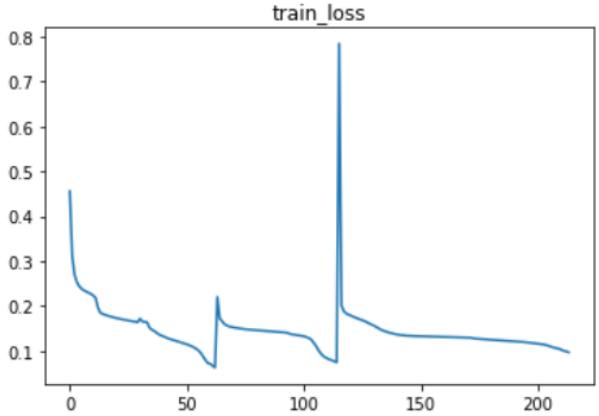
```
[[1500  149]
 [ 170  645]]
```

因為極度的 unbalance，所以全部往 0 猜的話，accuracy 會比較高，但這並不是我們想要的: 我們要的是 cross entropy 最小化，這也反映到了 confusion matrix 上面: 犧牲 0 的 acc，得到了 1 的 acc。

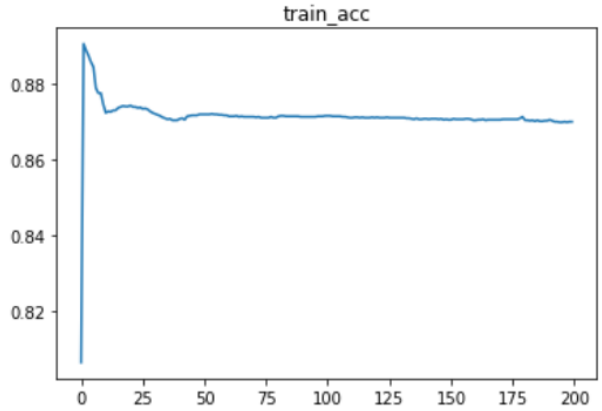
iv.分別使用 L=5、L=10、RNN、LSTM 來製作模型並比較 learning curve。
RNN、L=10

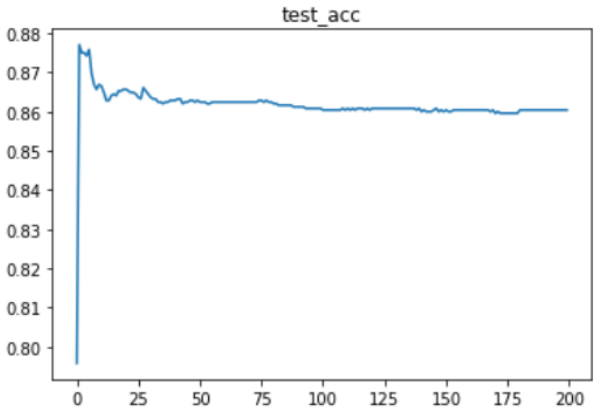


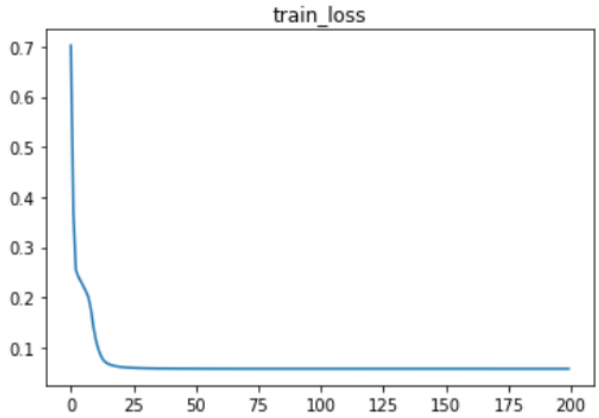




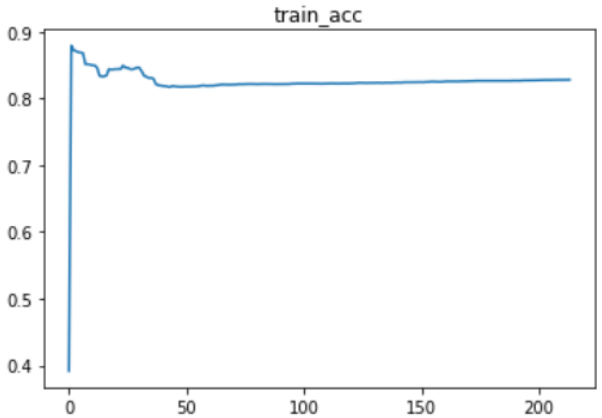
LSTM、L=5

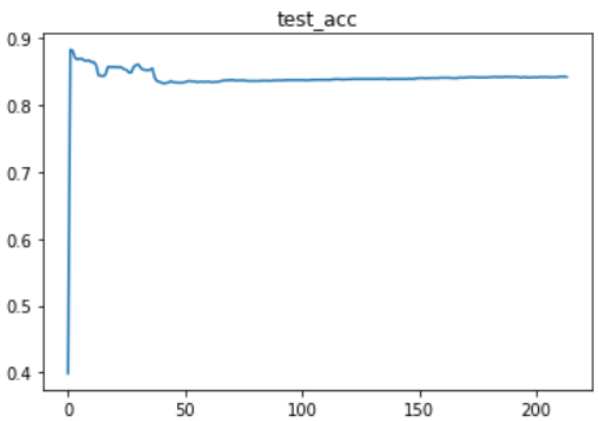


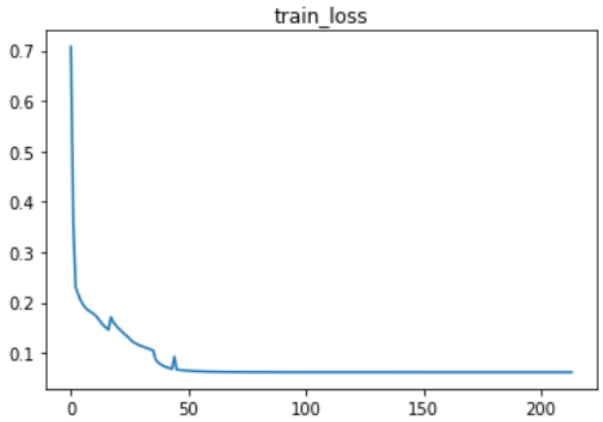




LSTM、L=10



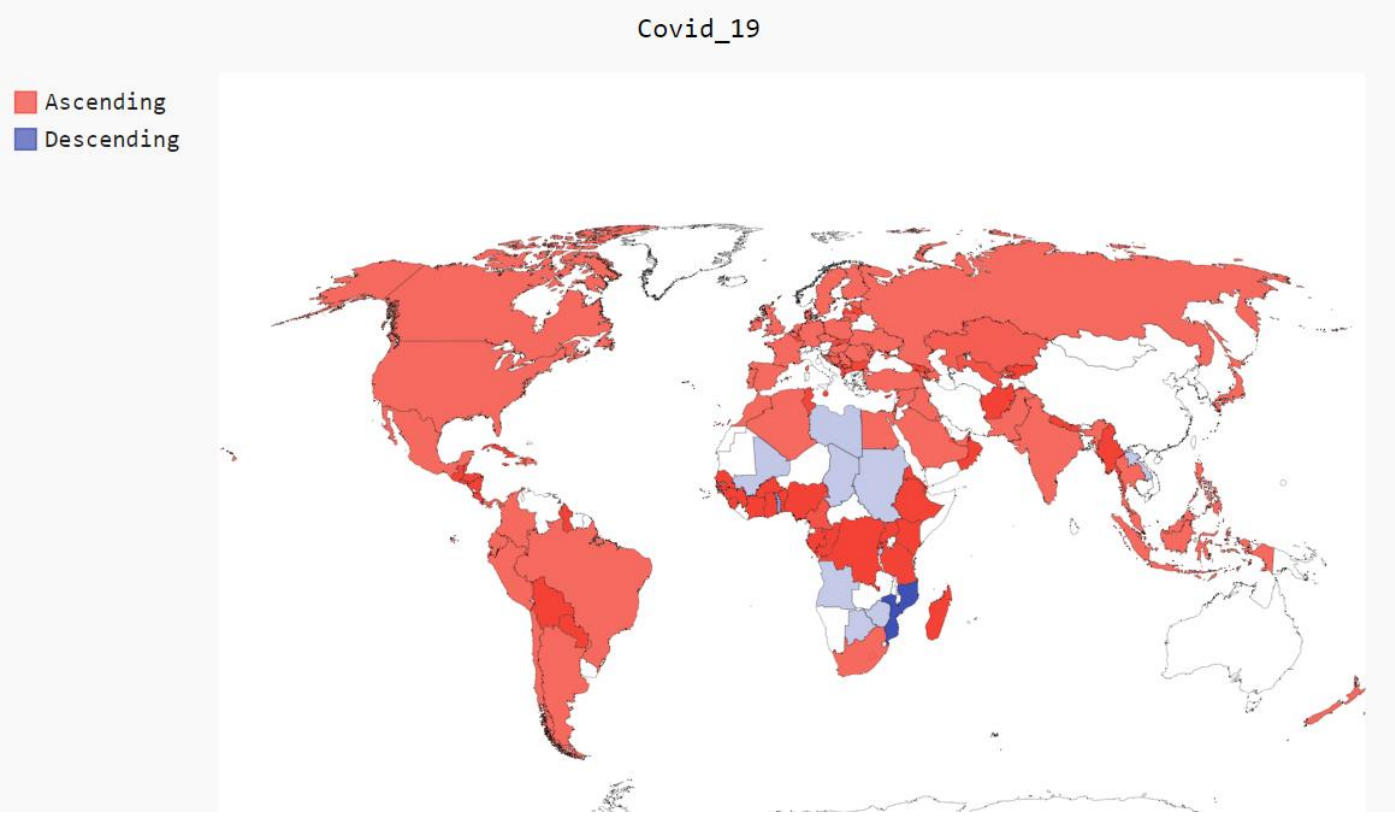




在比較 L=5、L=10，會發現預測率都是 L=5 比較強；再比較 RNN 與 LSTM，會發現 LSTM 的 learning curve 比較穩定。

v.使用 pygal 畫地圖:

繪畫地圖時，有在重新 tune 一次 model: 因為之前是有把資料切成 train and test，畫地圖是使用所有資料 train model，然後預測最後 L 天後確診人數會不會上升，使用 LSTM、L=5。



vi.結論:

其實我很期待 RNN 會梯度爆炸，可能是因為我的 learning rate 調得比較小，所以梯度爆炸沒那麼明顯。不過 LSTM 的效果確實相對比較好。

或許是因為梯度爆炸的問題，所以 RNN 相對 LSTM 沒有那麼穩定。

看來當時大部分的國家的確診人數都還是會持續上升。

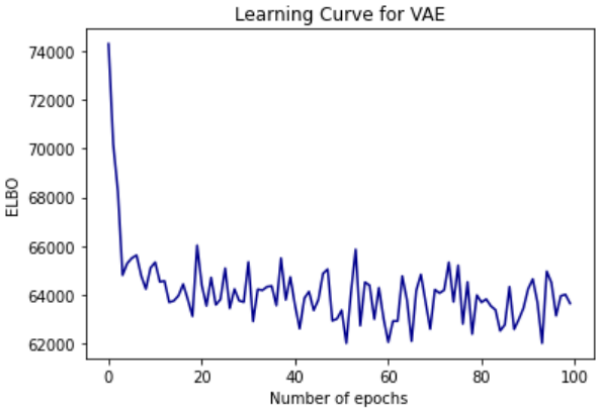
第二題:

i. 我先把圖載下來，然後針對各個小圖做 resize。

用 opencv 中的 cv2.size: 把圖片壓成 28*28 的樣子。

解釋: 原本大小為 M*N，要把它壓成 m*n: 我就把圖片分成(M*N)/(m*n)個長方塊，然後針對其值做(INTER_CUBIC 就是做立方插值)。

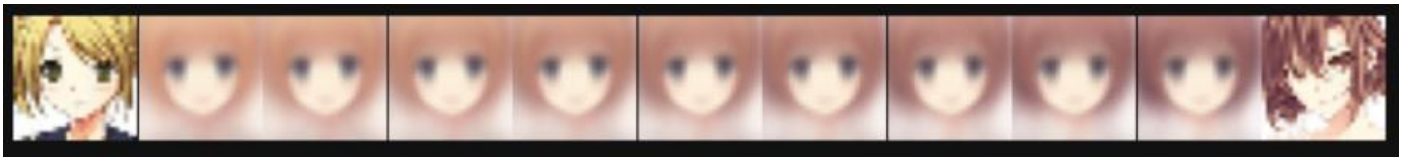
ii.


Learning Curve for VAE
ELBO
Number of epochs
74000
72000
70000
68000
66000
64000
62000
0
20
40
60
80
100

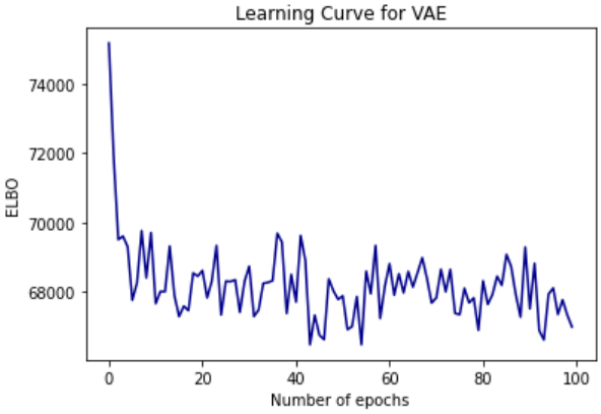
iii.

iv.

v.



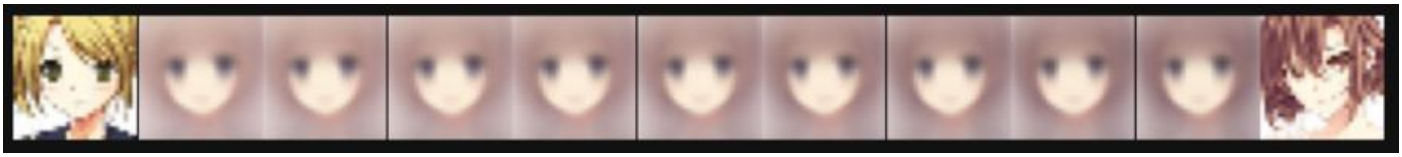
100*KL: ii



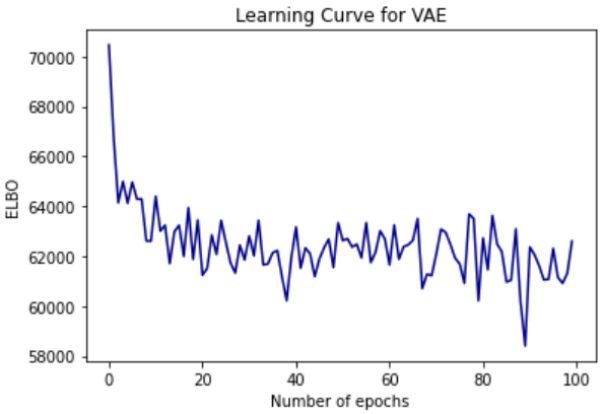
100*KL: iii

100*KL: iv

100*KL: v

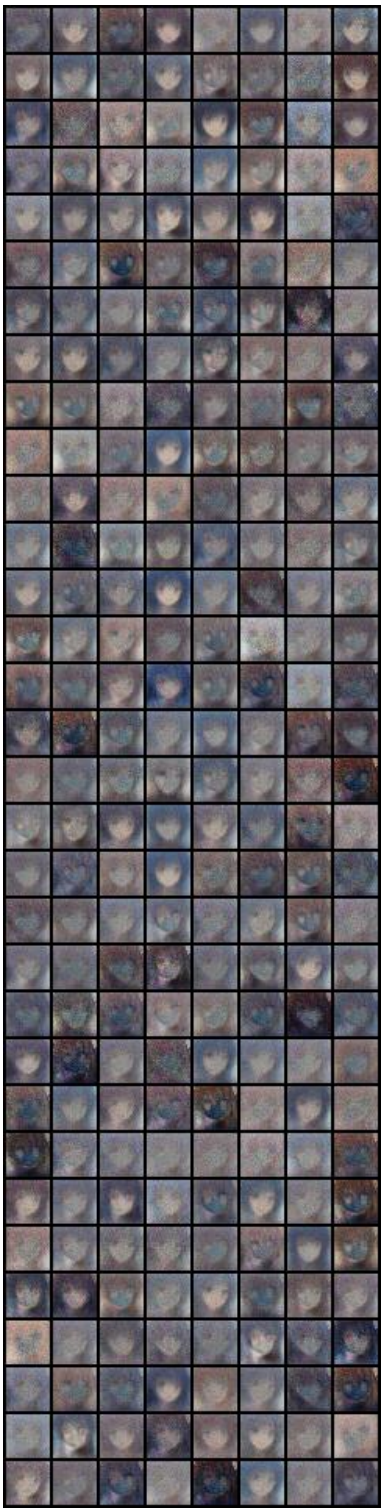


0*KL: ii

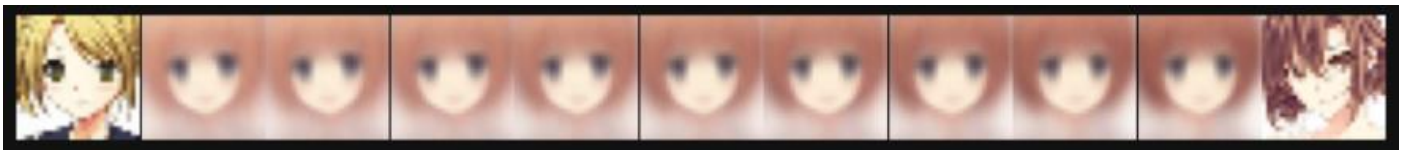


0*KL: iii

0*KL: iv



0*KL: v.



Viii:

VAE 確實難 train，也或許是因為我把解析度調整地太低。有些像怪物的 generator 我就沒放了。調整 KL 的倍率最直接影響的就是 loss function 的大小，但 0*KL 的時候，因為 KL 是下界，如果 0*KL 會讓不等式不夠 sharp，所以會造成模型可能不好。