Deep Learning Homework2

統研所碩一 曾鈺評 0852617

1 Recurrent Neural Network for Classification

(1) Please compute the correlation coefficient between two countries. You should plot these coefficients in all the pairs of two countries

原始資料如下，記得先將前兩行與前兩欄不必要的資料刪除掉

	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20
NaN	sum	sum	sum	sum	sum	sum	sum	sum
Country/Region	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Afghanistan	33.0	65.0	0	0	0	0	0	0
Albania	41.1533	20.1683	0	0	0	0	0	0
Algeria	28.0339	1.6596	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...
West Bank and Gaza	31.9522	35.2332	0	0	0	0	0	0
Western Sahara	24.2155	-12.8858	0	0	0	0	0	0
Yemen	15.552726999999999	48.516388	0	0	0	0	0	0
Zambia	-15.4167	28.2833	0	0	0	0	0	0
Zimbabwe	-20.0	30.0	0	0	0	0	0	0

187 rows × 84 columns

我們先用後一欄減去前一欄的方式算出每一天增加的確診人數，然後取最後四月份的增加確診人數查看如下：

	4/1/20	4/2/20	4/3/20	4/4/20	4/5/20	4/6/20	4/7/20	4/8/20	4/9/20	4/10/20	4/11/20	4/12/20
Afghanistan	63	36	8	18	50	18	56	21	40	37	34	52
Albania	16	18	27	29	28	16	6	17	9	7	17	13
Algeria	131	139	185	80	69	103	45	104	94	95	64	89
Andorra	14	38	11	27	35	24	20	19	19	18	0	37
Angola	1	0	0	2	4	2	1	2	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
West Bank and Gaza	15	27	33	23	20	17	7	2	0	4	1	22
Western Sahara	0	0	0	0	4	0	0	0	0	0	0	2
Yemen	0	0	0	0	0	0	0	0	0	1	0	0
Zambia	1	3	0	0	0	0	0	0	0	1	0	3
Zimbabwe	0	1	0	0	0	1	1	0	0	2	1	0

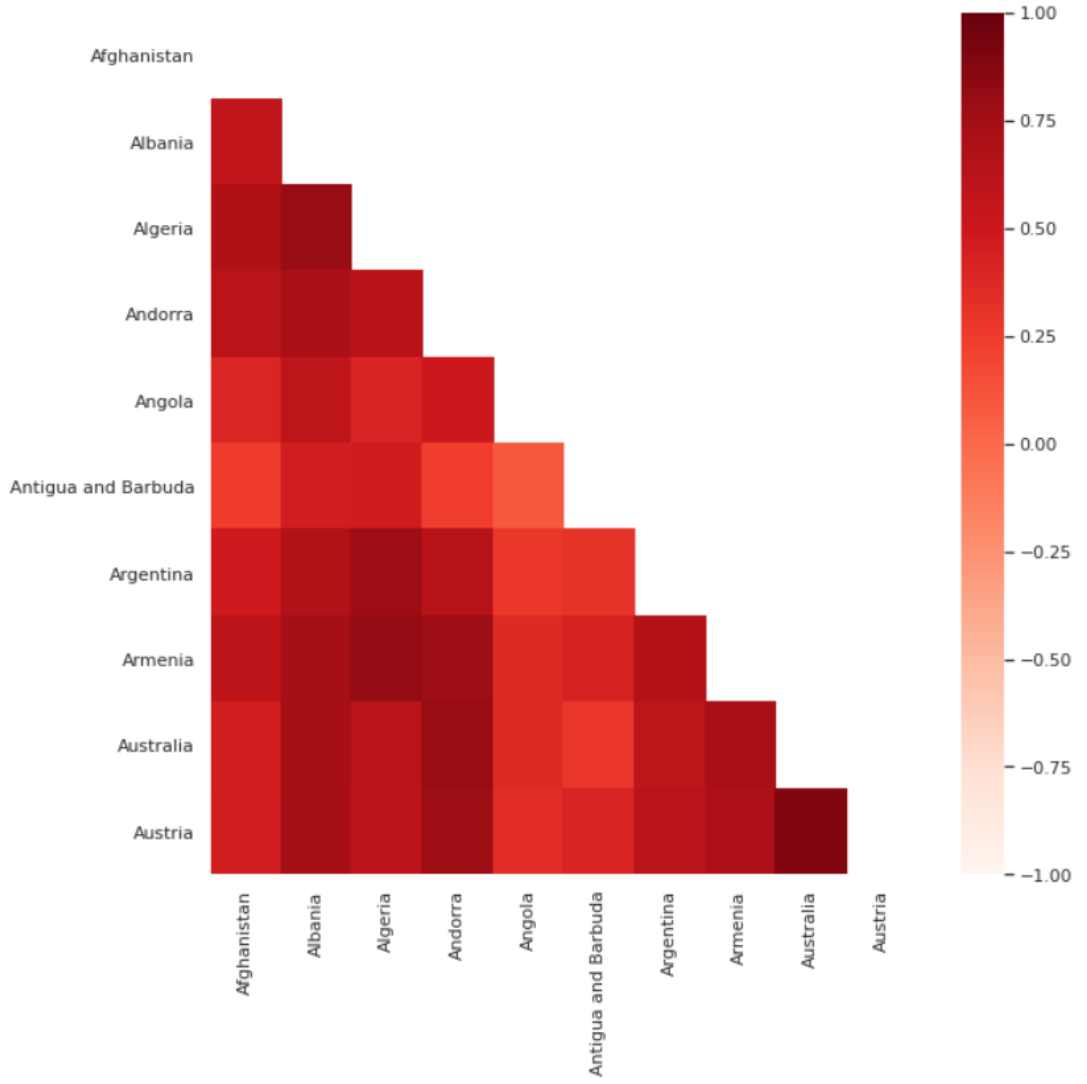
185 rows × 12 columns

接著我們用上面增加確診人數計算出 correlation matrix 如下，可發現大部分國家都是正相關，僅有少數幾個是負相關：

	Afghanistan	Albania	Algeria	Andorra	Angola	Antigua and Barbuda	Argentina	Armenia	Australia	Austria	Azerbaijan	Bahamas	Bahrain	Bangladesh
Afghanistan	1.000000	0.576543	0.687565	0.621826	0.399233	0.246526	0.479992	0.595829	0.456688	0.469930	0.814656	0.639980	0.545571	0.624718
Albania	0.576543	1.000000	0.800205	0.723154	0.584480	0.463324	0.675384	0.749470	0.739845	0.739458	0.666420	0.521626	0.506595	0.279333
Algeria	0.687565	0.800205	1.000000	0.626607	0.409259	0.476803	0.775745	0.817832	0.623110	0.600952	0.790705	0.575090	0.585387	0.474533
Andorra	0.621826	0.723154	0.626607	1.000000	0.506431	0.236505	0.644857	0.781346	0.794617	0.785611	0.601427	0.415097	0.434228	0.351829
Angola	0.399233	0.584480	0.409259	0.506431	1.000000	0.099321	0.277096	0.368090	0.372372	0.350925	0.537850	0.412305	0.137696	0.147385
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
West Bank and Gaza	0.512137	0.744971	0.748990	0.616715	0.493869	0.587319	0.589197	0.694183	0.497694	0.504904	0.529555	0.401918	0.470089	0.247990
Western Sahara	0.426037	0.326953	0.192540	0.322179	0.561456	-0.040469	0.086609	0.276255	0.020536	0.030589	0.246322	-0.059827	0.166279	0.321307
Yemen	0.215939	0.020705	0.195843	0.087040	-0.040159	-0.030012	0.337516	0.019091	0.025459	0.058141	0.224946	0.033759	0.113821	0.406029
Zambia	0.299311	0.418388	0.302966	0.621416	0.034358	0.086528	0.303384	0.484861	0.653173	0.525080	0.222608	0.180709	0.215146	0.081094
Zimbabwe	0.248334	0.242543	0.380363	0.443605	0.181338	0.113626	0.467380	0.445378	0.433317	0.374484	0.338043	0.176335	0.335495	0.253151

185 rows × 185 columns

把 correlation plot 畫出如下，由於國家數太多共有 185 個，故我們只取前 10 個：



(2) To process on your data,

I. **first, collecting the pairs of the country with high correlation with the other country. Here, you therefore collect a lot of countries in a set denoted as C.**

   我們把 threshold 設在 0.7，可以選到 143 個國家，這些國家的資料我們將用做 trainging data，比例大概有 77.3%，其他國家則可以當作 testing data，這樣做的目的是希望我們 train 出來的 model 可以學習到某個 pattern

II. **second, setting an interval (sequence length) L.**

   首先設定 L=5

III. **third, using the sequence x of the country in C to generate the subsequence with the length equal to L in x, And based on the next day of the last day in the subsequence, you can give it a label to specify if it is increasing or not.**

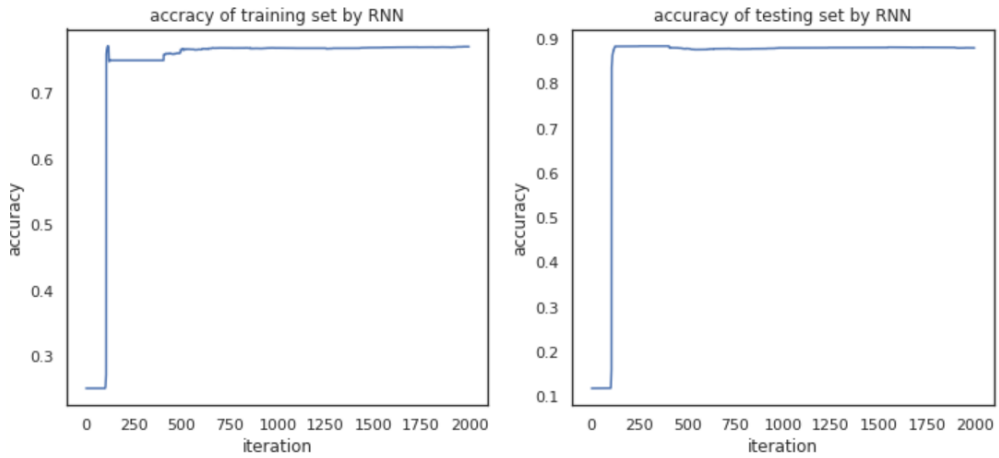
   根據 L=5 將每五天的資料切成 subsequence 並收集起來，在 label 的部分，我們根據每日的增加確認人數有沒有比前一天來的多來判斷，如果有增加則為 1，沒有增加確認人數或是增加確認人數反而減少則為 0。

IV. **fourth, repeating third step until all the countries in C are used to generate the data.**

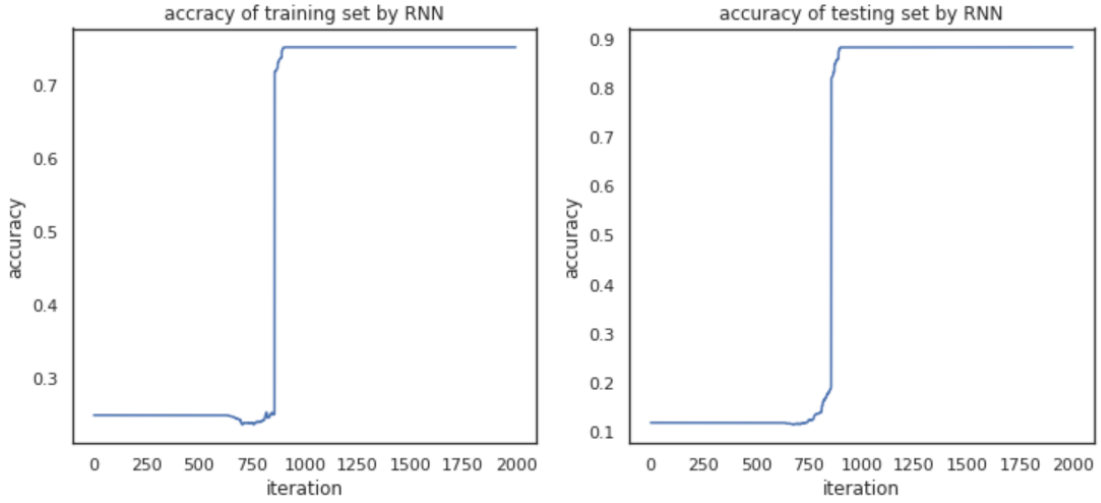
   重複上述步驟直到所有國家都被切成 subsequence

(3) Build a recurrent neural network to predict the label based on the given sequence. And show the accuracy of training and test.

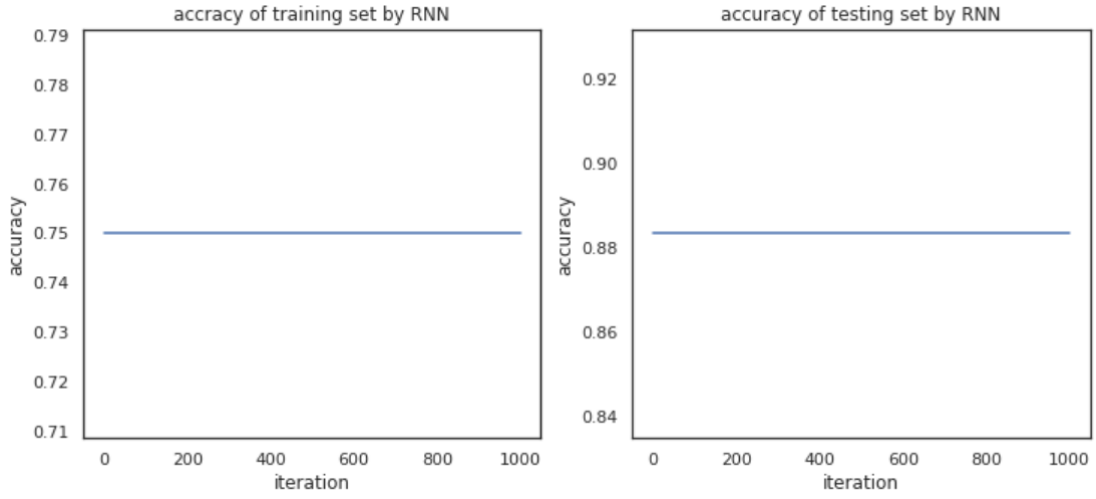
設計 RNN 架構：L=5，input dimension=1，hidden layer dimension=2，number of hidden layers=3，output dimension=1，出來的 output 會有(5 X batch size X 2)維，learning rate=0.01，loss 使用 cross entropy 計算，梯度優化演算法為 Adam，最後我們選取 L 的最後一個 output 經過 softmax 後即可得到預測為 0 或 1 的機率為何，再選擇機率大的做為我們預測 0 或 1 的依據。我們從圖可發現到 training accuracy 從 0.2502 增加到 0.7715，testing accuracy 範圍介於 0.1169 到 0.8831 之間。



選用 SGD 梯度優化演算法去做訓練，可能會出現以下情況，我們可發現學習曲線跟上面略有差別，有可能需要 train 比較多次，但是最後整體的 accuracy 是差不多的。



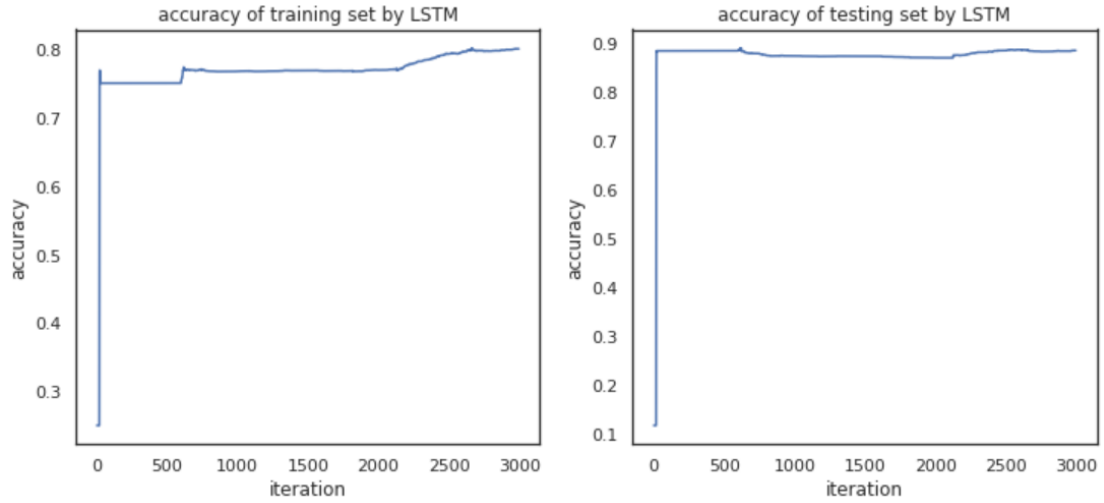
有時候會遇到一開始就剛好達到最大 accuracy 的情況，表示一開始可能就剛好預測到答案，此時不管怎麼 train 都不會再改變 accuracy。



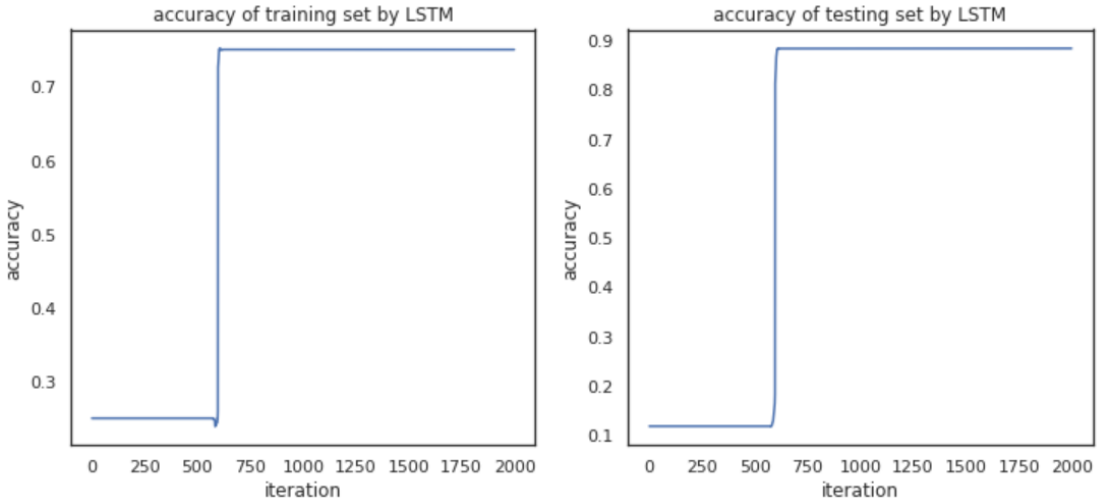
(4) Please implement different recurrent neural network like LSTM and GRU and change the value of interval L to analyze its effect on the result

LSTM:

LSTM 架構為 L=5，input dimension=1，hidden layer dimension=2，number of hidden layers=3，output dimension=1，出來的 output 會有(5 X batch size X 2)維，learning rate=0.01，loss 使用 cross entropy 計算，梯度優化演算法為 Adam，最後我們選取 L 的最後一個 output 經過 softmax 後即可得到預測為 0 或 1 的機率為何，再選擇機率大的做為我們預測 0 或 1 的依據。我們從圖可發現到 training accuracy 從 0.25 增加到 0.8006，testing accuracy 從 0.1169 增加到 0.8885，相較於 RNN，LSTM 似乎能表現出更佳的正確率



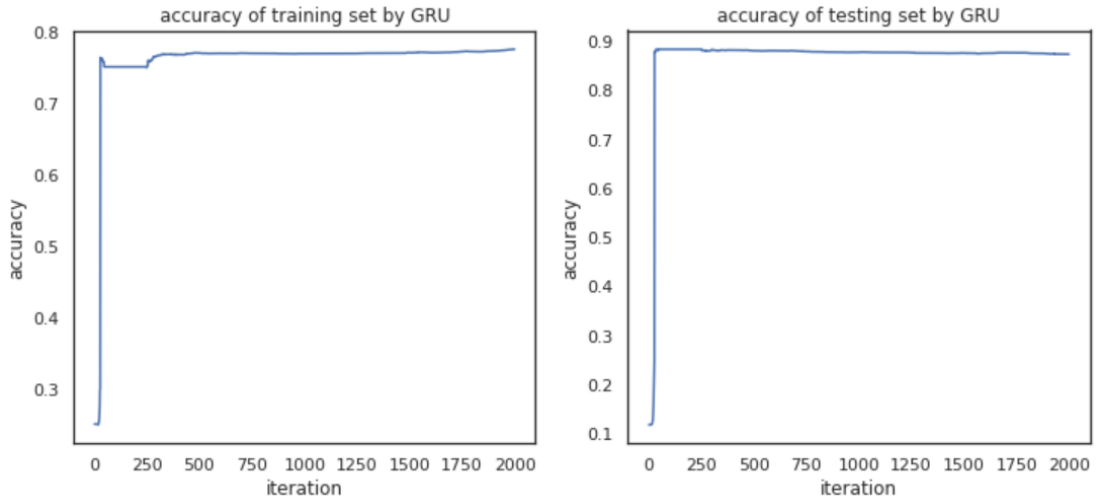
另外嘗試使用 SDG，可以發現到需要 train 比較多次才能達到與使用 Adam 差不多的 accuracy，不過 accuracy 會稍微略低。



GRU:

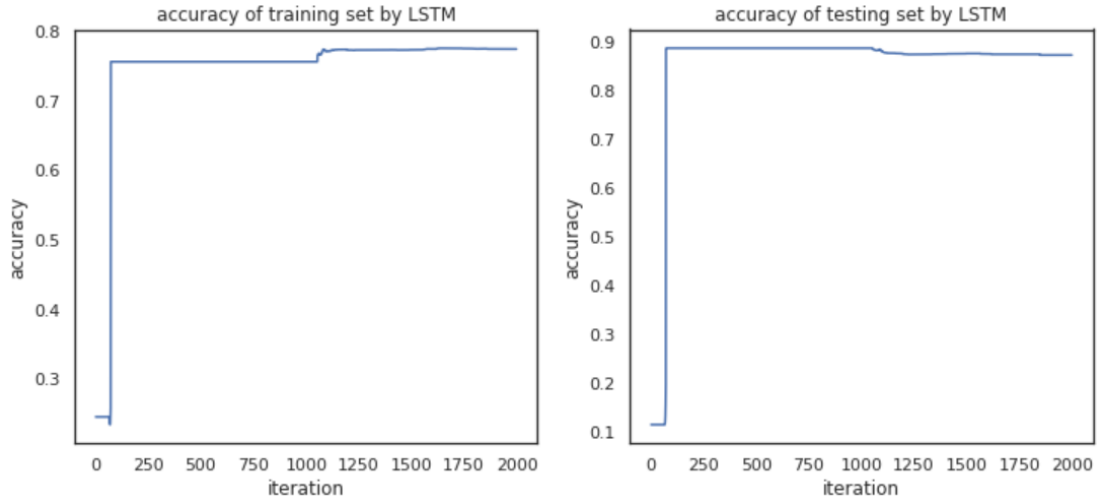
GRU 架構為 L=5，input dimension=1 ，hidden layer dimension=2，number of hidden layers=3，output dimension=1，出來的 output 會有(5 X batch

size X 2)維，learning rate=0.01，loss 使用 cross entropy 計算，梯度優化演算法為 Adam，最後我們選取 L 的最後一個 output 經過 softmax 後即可得到預測為 0 或 1 的機率為何，再選擇機率大的做為我們預測 0 或 1 的依據。我們從圖可發現到 training accuracy 從 0.2502 增加到 0.768，testing accuracy 從 0.1169 至 0.8831。

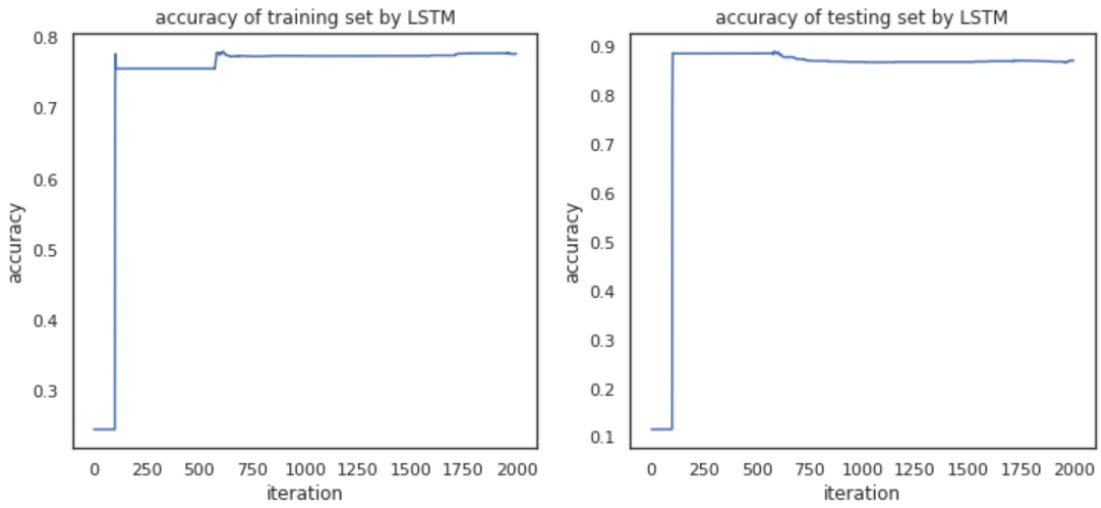


由以上結果看起來 LSTM 表現最佳，故我們接下來用 LSTM 選用不同的 sequence L 來進行。

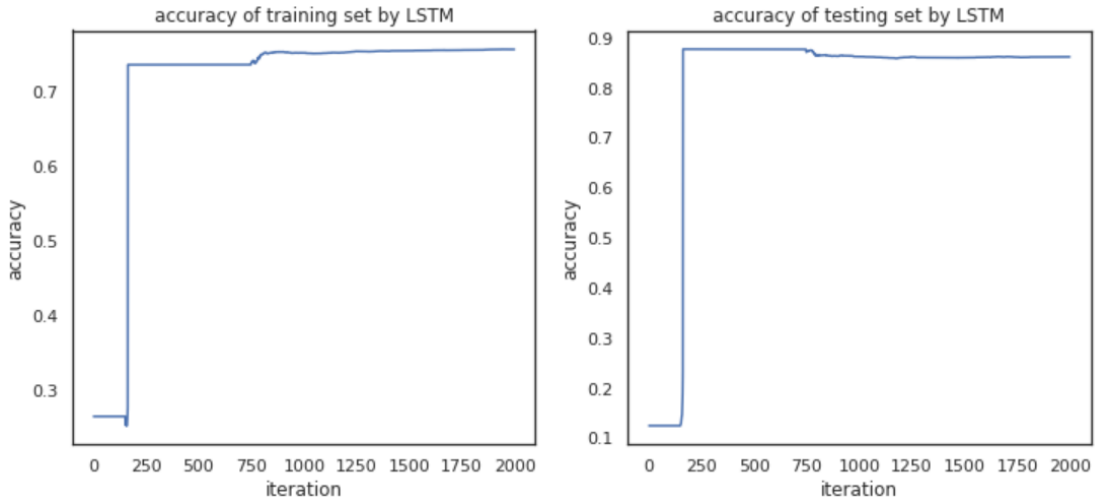
L=3 時，表現與 L=5 差異不大，training set 大約是從 0.2333 增加到 0.7751，testing accuracy 從 0.1151 上升至 0.8849。



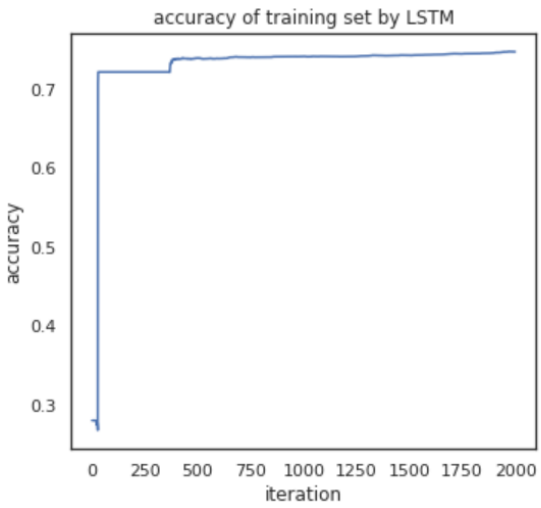
L=7 時，training set 大約是從 0.2693 增加到 0.7661，testing accuracy 從 0.1197 上升至 0.8851。

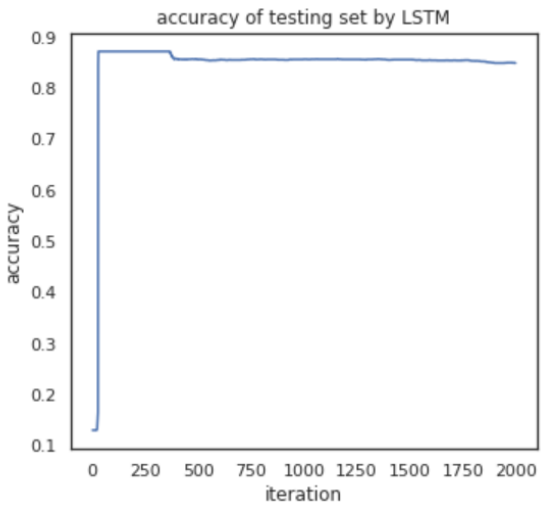


L=10，training set 是從 0.2523 增加到 0.7561，testing accuracy 從 0.1237 上升至 0.8763。

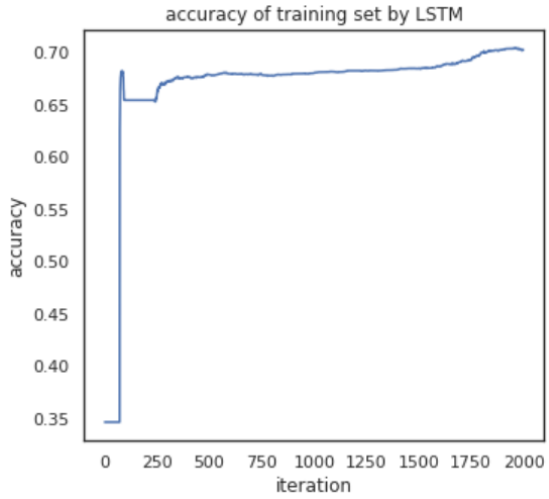


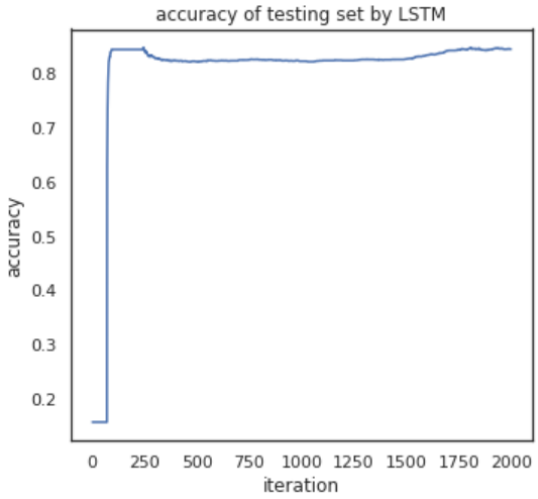
L=14，training set 是從 0.2669 增加到 0.7469，testing accuracy 從 0.1294 上升至 0.8706。



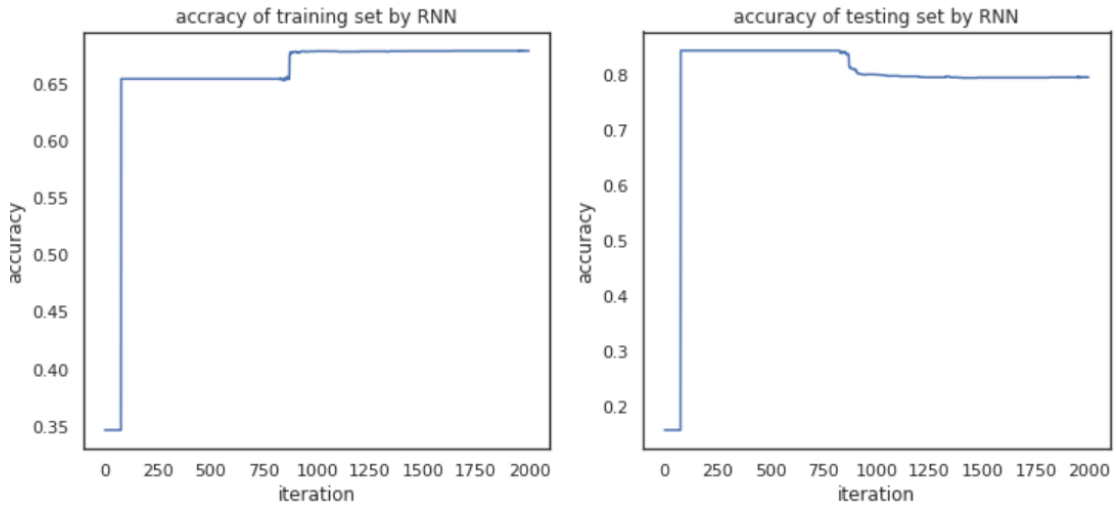


L=28，training set 是從 0.3462 增加到 0.7038，testing accuracy 從 0.1563 上升至 0.8468。

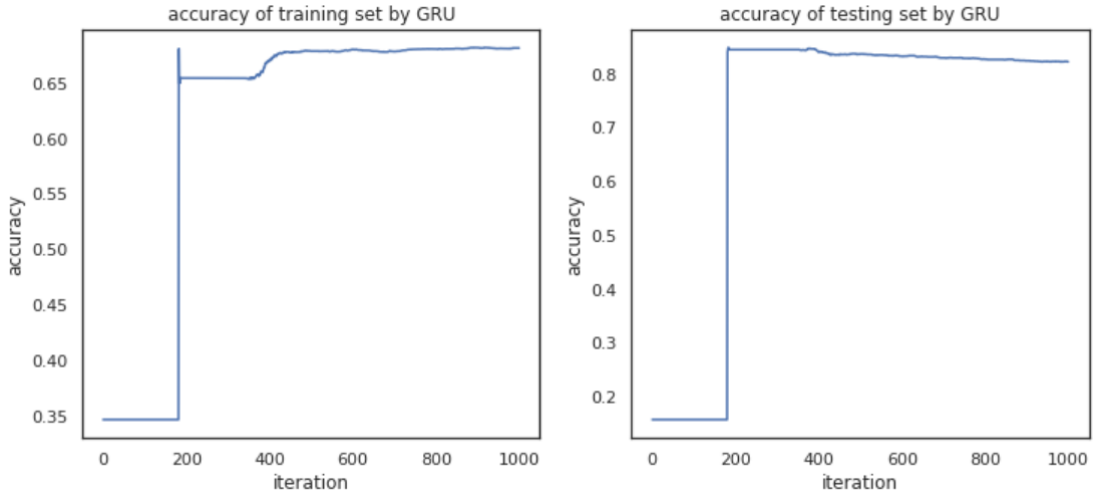




另外在 L=28 時，我們也同樣觀察 RNN 及 GRU 的表現，RNN 的 training set 是從 0.3462 增加到 0.6786，testing accuracy 從 0.1563 上升至 0.8437。



在 L=28 時， GRU 的 training set 是從 0.3462 增加到 0.6814，testing accuracy 從 0.1563 上升至 0.8477。



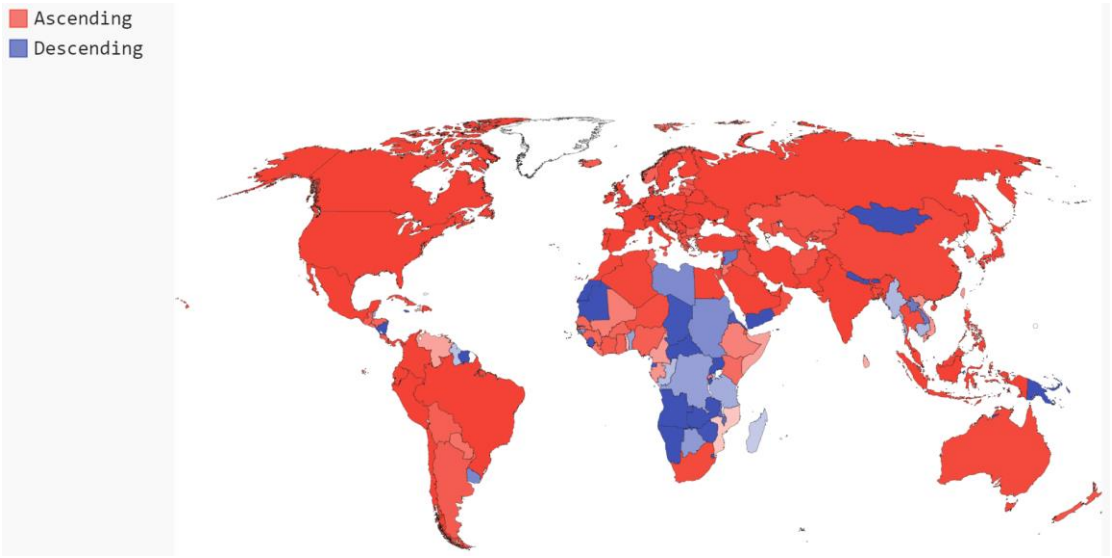
(5) Compute the probability for each country and plot on a world map by using "pygal" package in python as below example.

在使用 pygal package 時，我們需要先把各個國家的名稱轉成 country code，不過本題的國家名稱與 pygal 的名稱對不太起來，因此我們可先用以下的 dictionary 把國家名稱更換。

{'Bolivia': 'Bolivia, Plurinational State of', 'Brunei': 'Brunei Darussalam', 'Congo (Brazzaville)': 'Congo, the Democratic Republic of the', 'Congo (Kinshasa)': 'Congo', 'Dominica': 'Dominican Republic', 'Holy See': 'Holy See (Vatican City State)', 'Iran': 'Iran, Islamic Republic of', 'Korea, South': 'Korea, Republic of', 'Laos': "Lao People's Democratic Republic", 'Libya': 'Libyan Arab Jamahiriya', 'Moldova': 'Moldova, Republic of', 'North Macedonia': 'Macedonia, the former Yugoslav Republic of', 'Russia': 'Russian Federation', 'South Sudan': 'Sudan', 'Syria': 'Syrian Arab Republic',

'Taiwan*': 'Taiwan, Province of China', 'Tanzania': 'Tanzania, United Republic of', 'Venezuela': 'Venezuela, Bolivarian Republic of', 'Vietnam': 'Viet Nam', 'Burma': 'Myanmar', 'Cabo Verde': 'Cape Verde', 'Eswatini': 'Swaziland', 'Czechia': 'Czech Republic', 'US': 'United States'}

以下我們以 RNN 去算出最後 L=5 天所預測出來的機率，並根據它預測出的增加確診人數有無增加區分成 ascending 與 descending，可以得到以下地圖。



(6) Do some discussion based on your result.

I. 由上面結果我們可以推論出 LSTM 正確率的表現最佳，GRU 幾乎能追上 LSTM 的表現，而 RNN 相較起來表現比較不好，推測是因為沒有考慮 forget gate 是否把記憶清除掉導致的關係。

II. 序列的長度會影響到訓練模型的表現，當長度為 L=5 左右的時候正確率的表現比較好，當長度越長，到 L=28 時，表現會越來越不好，推測是因為過多資訊反而會有不必要的資訊影響模型訓練。

III. 整體資料有偏頗於增加確診人數上升的情況(也就是 y label 幾乎為 1 的比例太多了)，也許我們可以考慮用不同於 accuracy 的方法來衡量我們模型的好壞，比如 f1-score 等等。

2 Variational Autoencoder for Image Generation

In this exercise, you will construct a Variational Autoencoder (VAE) for image reconstruction by the animation faces dataset here.

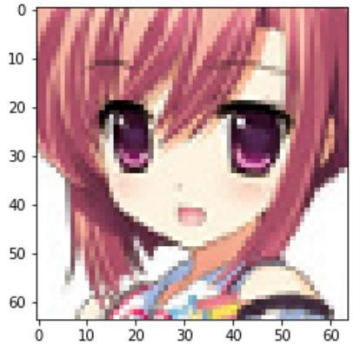
VAE paper can be downloaded here. You should preprocess the images such as resizing or cropping by yourself before implementation.

(1) Describe in details how to preprocess images (such as resizing or cropping)

and design the network architecture.

在前處理的部分我們把圖片讀取進來後轉換成三維0-255像素的numpy array，這裡我們沒有特別選擇用resizing或是cropping，因為所有圖片皆為清楚可見的頭像，不太需要重新切割；而圖片大小為64X64，不算太大，算是在電腦可處理的範圍。

檢查其中一張圖片如下所示：



架構設計：

在encoder的部份我們使用兩層convolution layers，一開始原始資料共有3個channel進入後，通過這兩層convolution layers，會變成64與128個channel，kernel size皆為3，padding皆為1，第一層convolution layers的stride設為1，第二層convolution layers的stride設為2。兩層convolution layers皆使用batch normalization、ReLU，與size=2的max pooling。

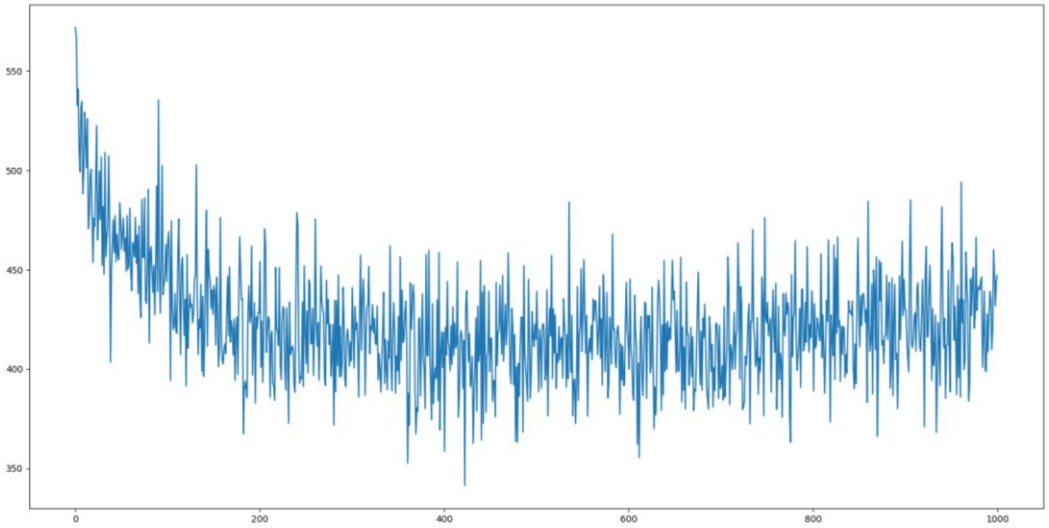
接著是mean與log variance的計算，我們把encoder出來的結果flatten為128*8*8，分別經過三層linear後得到256、100，最後得到latent dimension=32的向量，其中mean每通過一層linear都會再連接一層ReLU，log variance每通過一層linear都會再連接一層LeakyReLU。

最後是decoder的部分

把mean加上standard error呈上一個隨機向後，分別經過四層linear，得到128、512、1024，最後得到64*64*3的向量，在通過每一層linear後都會再連接一層ReLU，最後一層則是連接Sigmoid把數值介於0-1。

(2) **Plot the learning curve of the negative evidence lower bound (ELBO) of log likelihood of training images.**

我們觀察到loss在前200次的iteration有比較明顯的下降，大約從570下降到410，200次之後的loss一直在410附近上下震盪

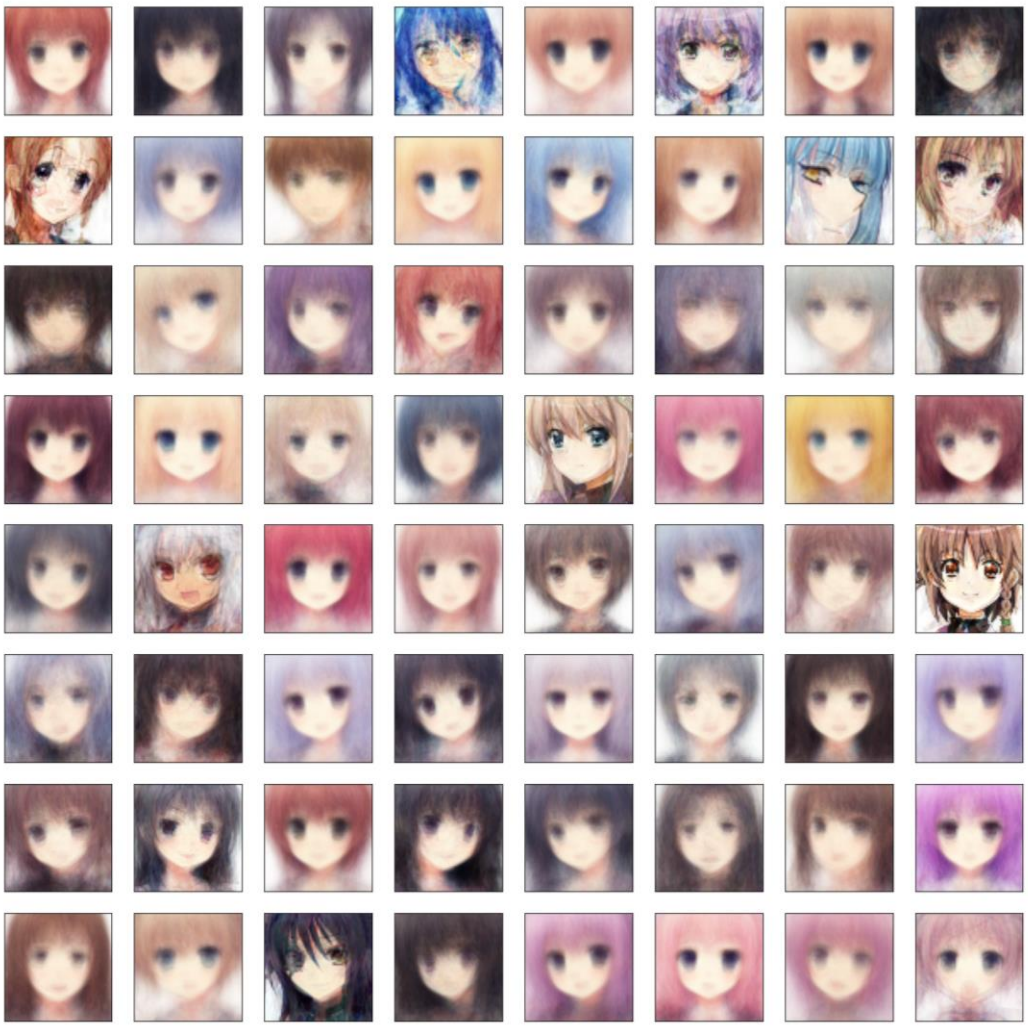


(3) Show some examples reconstructed by your model.

選取前64個原圖如下：

用我們的auto-encoder訓練出來的圖形如下：

我們可以發現VAE有學到某些特定的特徵，比如不同顏色的頭髮，人頭不同的轉向，但也不盡然相同，在KL的作用下，會使得mean接近0，variance接近1，模型因為增加了一些噪音而使生成的圖片有了隨機性，比如倒數第二排最右邊三個圖片原本長的有些相像，但經過VAE的圖片卻各自有了不同的髮色與髮型。



(4) Sample the prior p(z) and use the latent codes z to synthesize some examples when your model is well-trained.

透過隨機抽取Normal(0,1)的分布我們可以看到VAE可產生不同髮色、髮型、臉部朝向等特徵。

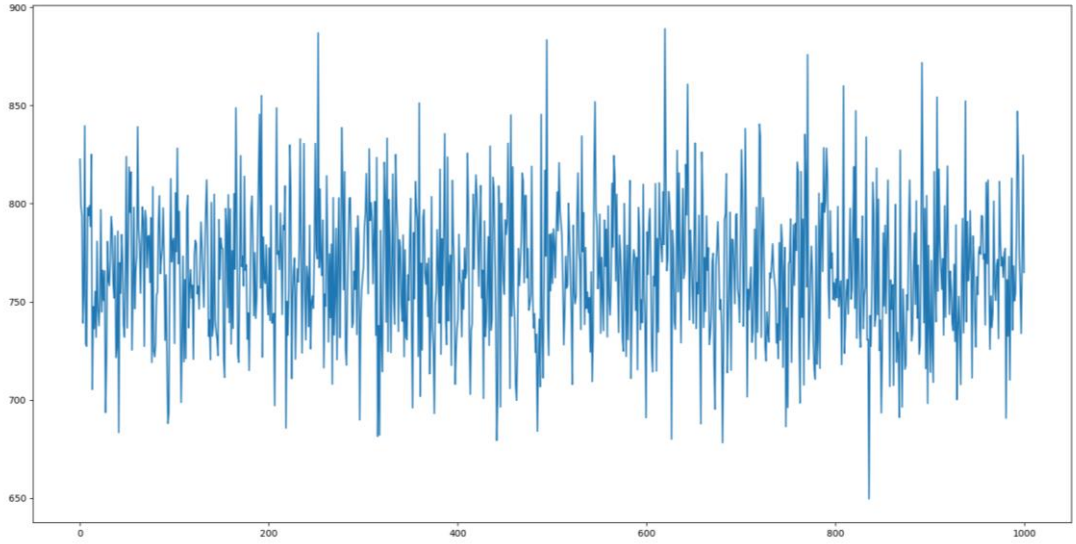
(5) Show the synthesized images based on the interpolation of two latent codes z between two real samples. (Hint: you need to interpolate two latent variables encoded form two real images. Several new latent variables in two

images should be drawn. Finally, decode those new latent variables into images.)

我們拿兩張圖，分別是一男一女，臉部一個朝右一個朝左的人臉來觀察，中間7張圖片是以這兩張圖片的特徵用內插法生成，再用decoder產生圖片，我們可以發現到中間的圖片有遵循某些固定的特徵，像是人臉會由朝右慢慢朝向左，但是也有些地方某些圖片會表現的跟左右圖片不一樣，比如左邊數過來第六張圖片的髮型帶些微卷，臉型特別偏左，最中間那張髮色跟左右兩張都不連貫，這些都是KL產生的效果。

(6) Multiply the Kullback-Leibler (KL) term in ELBO by 100 in your loss and repeat ii～v.

把KL乘上100後可發現，因為加入的噪音項太大，導致mean不接近0，variance不接近1的趨勢增加過大，原來的log likelihood失去原本的功能，整體loss一直在760左右上下震盪，並沒有明顯下降的趨勢。



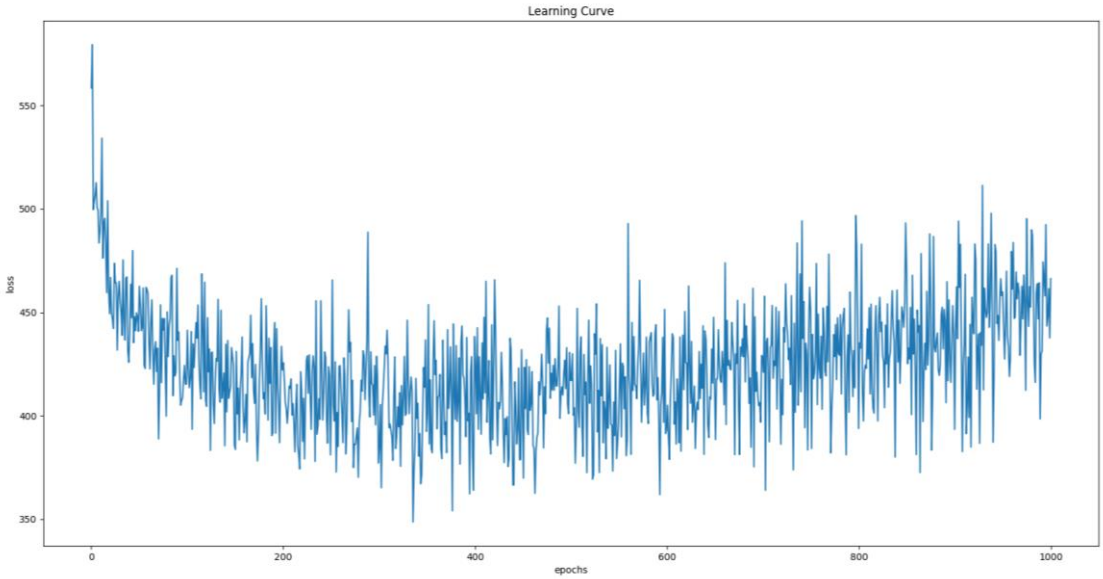
選取64個原圖同上一小題(這裡就不再貼一次了)，以下是用原來64個原圖經過VAE得到的結果，從上面Loss我們可以猜測的出來這個訓練效果不佳，果然從圖片我們可以看到圖片幾乎都學習到相同的特徵，幾乎都只學到淺咖啡、深咖啡的正面人臉。

我們一樣從Normal(0,1)的分布抽8組數據出來生成以下圖片，生出來的圖片跟上面結果很像幾乎都長的一模一樣，這種結果也在預期之內，因為我們訓練出來的模型就只有學到特定的特徵。

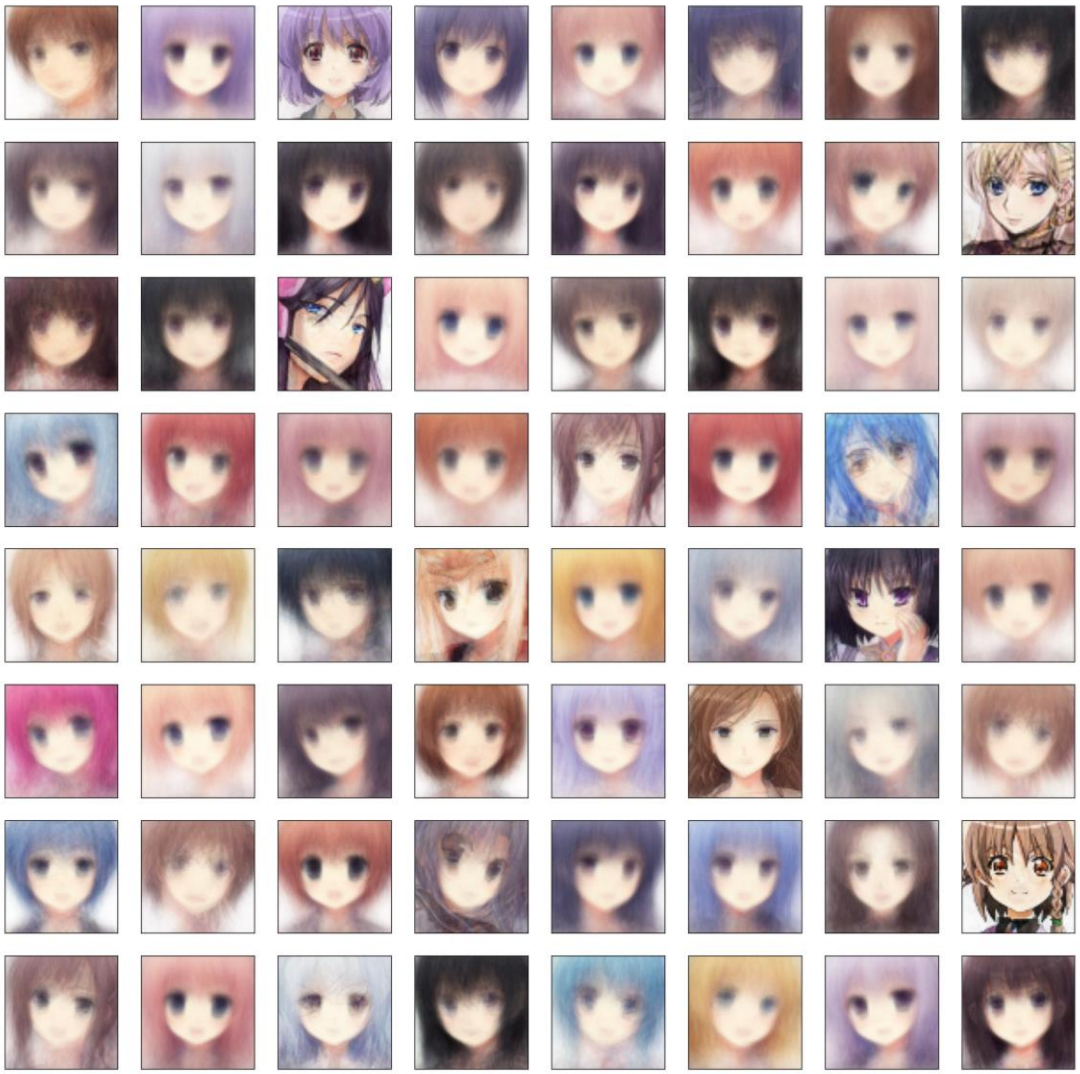
我們拿跟前一小題相同的一男一女照片，同樣用內插算出中間幾張圖的特徵，再用decoder把照片生成，因為左右兩張照片非常相像，所以整排照片可幾乎說是沒什麼太大變化，不過仍然可以發現髮色部分還是有一些不一樣。

(7) Multiply the KL term by 0 in ELBO and repeat ii～v.

把KL乘上0相當於沒有KL這一項，此時VAE就會變為Auto-encoder，整體loss大約從570下降到400，200次之後的loss一直附近上下震盪且有小小上升的趨勢。



選取64個原圖同上一小題(請參閱上面)，在沒有KL的作用下，表示我們假設中間latent feature學習到的是單一個值，而不是呈現一個分布，但通常很難對中間latent feature決定一個具體的數值，這樣的情況會使得decoder解析出來的結果比較不接近原圖。所以我們可以發現有些圖片甚至經過解析後成為了另一張圖片，這樣的學習是我們比較不樂見的。



從Normal(0,1)的分布抽8組數據出來生成以下圖片，看起來生成效果還不錯，似乎沒有受到沒有KL這一項的影響。

我們拿跟上一小題同樣的一男一女兩張圖，分別是臉部一個朝右一個朝左的人臉來觀察，中間7張圖片是以這兩張圖片的特徵用內插法生成，再用decoder產生圖片，我們可以發現到中間的圖片幾乎遵循某些固定的特徵，人臉會由朝右慢慢朝向左，沒有像KL乘上一倍時一樣突然出現較大的變異，推測是因為我們是假設在latent feature固定為某些值的情況下去訓練出這樣的模型所導致的。

(8) Do some discussion on the effect of KL term based on your result.

I. 在KL的作用下，我們假設中間的latent feature是一個mean接近0，variance接近1的常態分配，模型因為增加了一些噪音而使最後decoder生成的圖片更加接近原圖。

II. 把KL乘上100後可發現，因為加入的噪音項太大，導致mean不接近0，variance不接近1的趨勢增加過大，我們訓練出來的模型就只有學到特定的特徵，生出來的圖片幾乎都長的一模一樣。

III. 把KL乘上0後相當於Auto-encoder在作用，在沒有KL的作用下，表示我們假設中間latent feature學習到的是單一個值，而不是一個Normal(0,1)的分布，但圖片的特徵變動範圍很廣，我們對中間latent feature決定一個具體的數值這樣的假設是不太正確的，所以可以發現decoder解析出來的結果比較不接近原圖。不過如果隨機抽取Normal(0,1)的幾組數值去生成新的圖片是沒問題的，沒有KL似乎不會影響隨機生成圖片這部分。