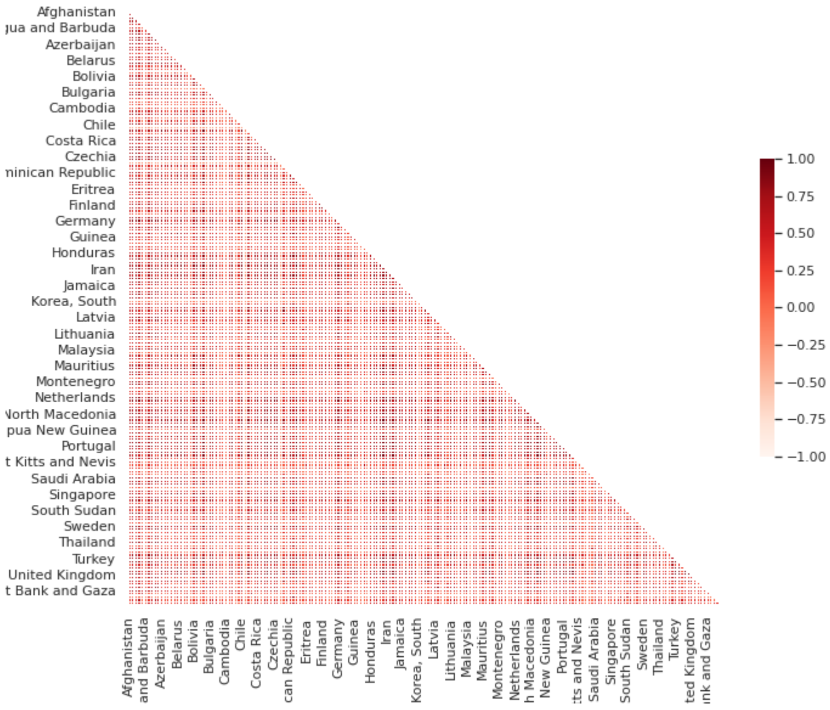
# DL HW2_0513460 資財09胡明秀

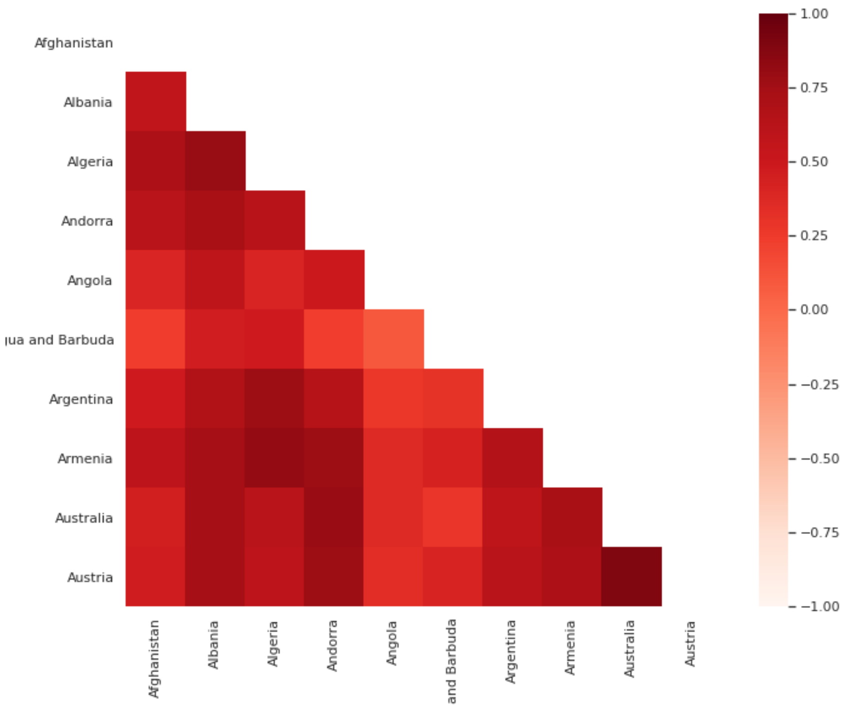
## Q1 Recurrent Neural Network for Classification

i. Correlation between two countries

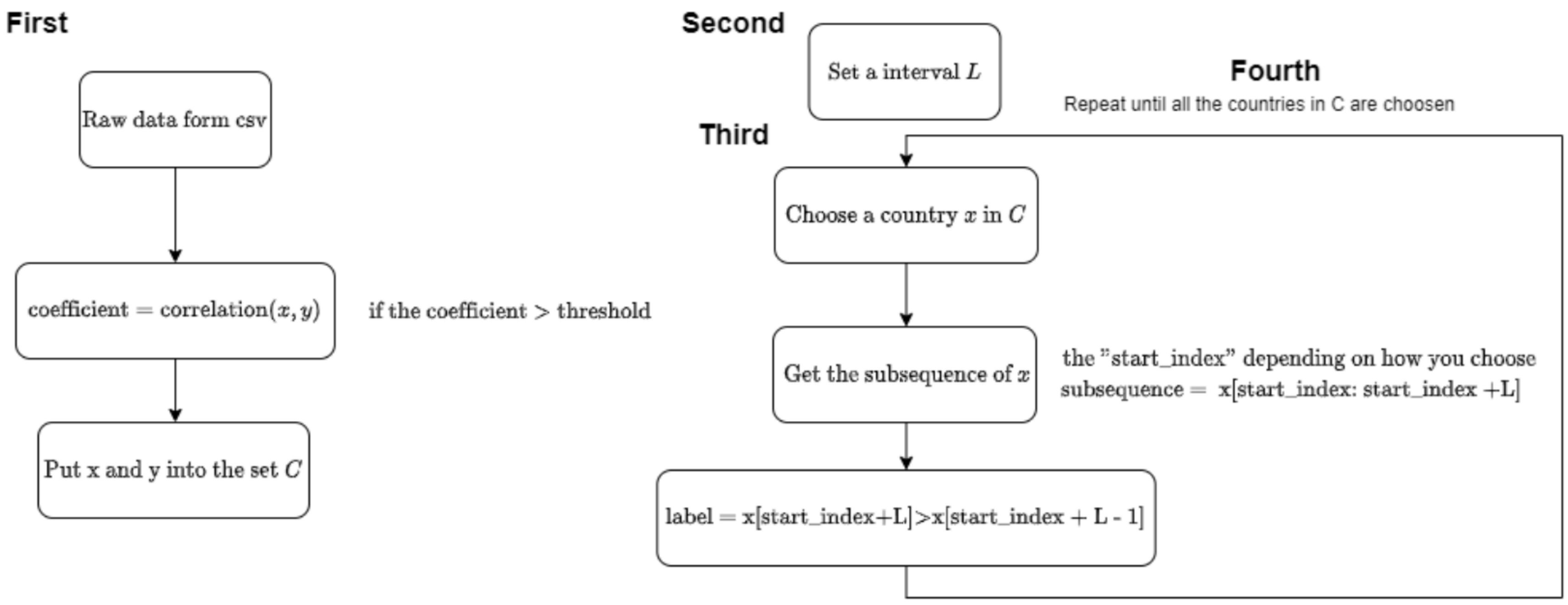
- Data Description: Number of confirmed cases in different countries
- 以下是全部國家的correlation heatmap




- 以下是取前面10個國家的correlation heatmap
  - ▼ 顏色較深的country pair為確診數的correlation較高的組合




ii. Process the data with the following workflow




- 根據統計學上correlation的相關經驗，若corr(x, y) > 0.7, 則代表x, y為高度正相關。故選取threshold = 0.7來取得相關性較高的country pair放入Set C:
  - ▼ 共選取143個國家（占原來完整training data的 143/185 = 77.29%)
  - ▼ 選取的143個國家（註：中國和其他大多數國家的correlation < 0, 故在這樣的workflow下不會被選入traning data中，但經實驗後似乎對ACC沒有太大的影響。）

```
['Afghanistan' 'Albania' 'Algeria' 'Andorra' 'Argentina' 'Armenia'
 'Australia' 'Austria' 'Azerbaijan' 'Bahamas' 'Bahrain' 'Bangladesh'
 'Barbados' 'Belarus' 'Belgium' 'Belize' 'Benin' 'Bolivia'
 'Bosnia and Herzegovina' 'Botswana' 'Brazil' 'Bulgaria' 'Burkina Faso'
 'Cameroon' 'Canada' 'Chad' 'Chile' 'Colombia' 'Congo (Kinshasa)'
 'Costa Rica' "Cote d'Ivoire" 'Croatia' 'Cuba' 'Cyprus' 'Czechia'
 'Denmark' 'Djibouti' 'Dominican Republic' 'Ecuador' 'Egypt' 'El Salvador'
 'Eritrea' 'Estonia' 'Ethiopia' 'Fiji' 'Finland' 'France' 'Gabon'
 'Georgia' 'Germany' 'Ghana' 'Greece' 'Grenada' 'Guatemala' 'Guinea'
 'Haiti' 'Honduras' 'Hungary' 'Iceland' 'India' 'Indonesia' 'Iran' 'Iraq'
 'Ireland' 'Israel' 'Italy' 'Japan' 'Jordan' 'Kazakhstan' 'Kenya' 'Kosovo'
 'Kuwait' 'Kyrgyzstan' 'Latvia' 'Lebanon' 'Liberia' 'Libya' 'Lithuania'
 'Luxembourg' 'Madagascar' 'Malaysia' 'Mali' 'Malta' 'Mauritius' 'Mexico'
 'Moldova' 'Monaco' 'Mongolia' 'Montenegro' 'Morocco' 'Mozambique'
 'Netherlands' 'New Zealand' 'Niger' 'Nigeria' 'North Macedonia' 'Norway'
 'Oman' 'Pakistan' 'Panama' 'Papua New Guinea' 'Paraguay' 'Peru'
 'Philippines' 'Poland' 'Portugal' 'Qatar' 'Romania' 'Russia' 'Rwanda'
 'Sao Tome and Principe' 'Saudi Arabia' 'Senegal' 'Serbia' 'Sierra Leone'
 'Singapore' 'Slovakia' 'Slovenia' 'Somalia' 'South Africa' 'Spain'
 'Sri Lanka' 'Sudan' 'Sweden' 'Switzerland' 'Taiwan*' 'Tanzania'
 'Thailand' 'Timor-Leste' 'Togo' 'Trinidad and Tobago' 'Tunisia' 'Turkey'
 'US' 'Ukraine' 'United Arab Emirates' 'United Kingdom' 'Uruguay'
 'Uzbekistan' 'Venezuela' 'Vietnam' 'West Bank and Gaza' 'Yemen']
```

- 將Set C中的143個國家做出label。此為一個binary classification的問題
  - ▼ interval_L = 7
  - ▼ label = 0: subsequence[-1] < next day (趨勢：總確診數上升)
  - ▼ label = 1: subsequence[1] > next day( 趨勢：總確診數下降)
- 按照workflow的取法，training data + testing data會有10582筆
  - ▼ label 0: 7874, label 1: 2708 (2708/7874 = 34.39%)
  - ▼ training data: 10439*0.8 = 8465
  - ▼ testing data: 10439*0.2 = 2117

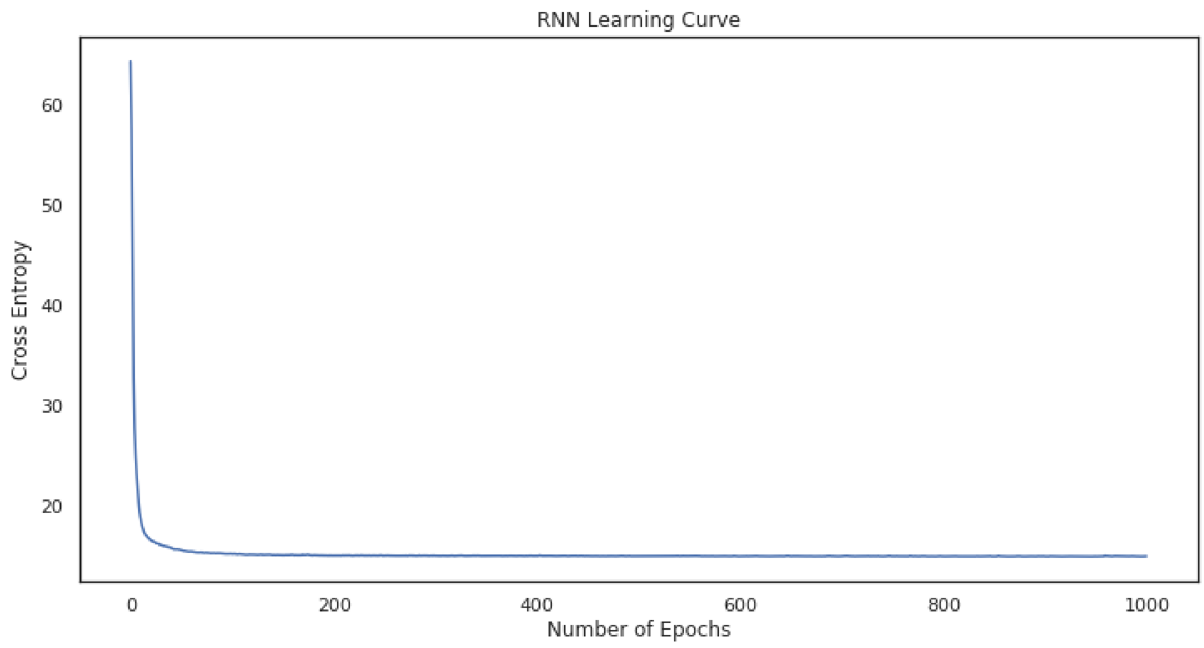
iii. Build a recurrent neural network to predict the label based on the given sequence

- sub sequence的長度為7 ( interval_L = 7), 但是每個time step我們只會放入一天，故 input = 1
- batch_first = True —> input format: (batch_size, time step, input)

- hidden size = 128, num_layers = 2

```
RNN(
  (rnn): RNN(1, 128, batch_first=True)
  (out): Linear(in_features=128, out_features=1, bias=True)
)
```

- Training Accuracy and Test Accuracy
    - ▼ batch size = 128, EPOCH = 1000
    - ▼ Training Accuracy: 77.259% (6540/8465)
    - ▼ Testing Accuracy: 74.681% (1581/2117)





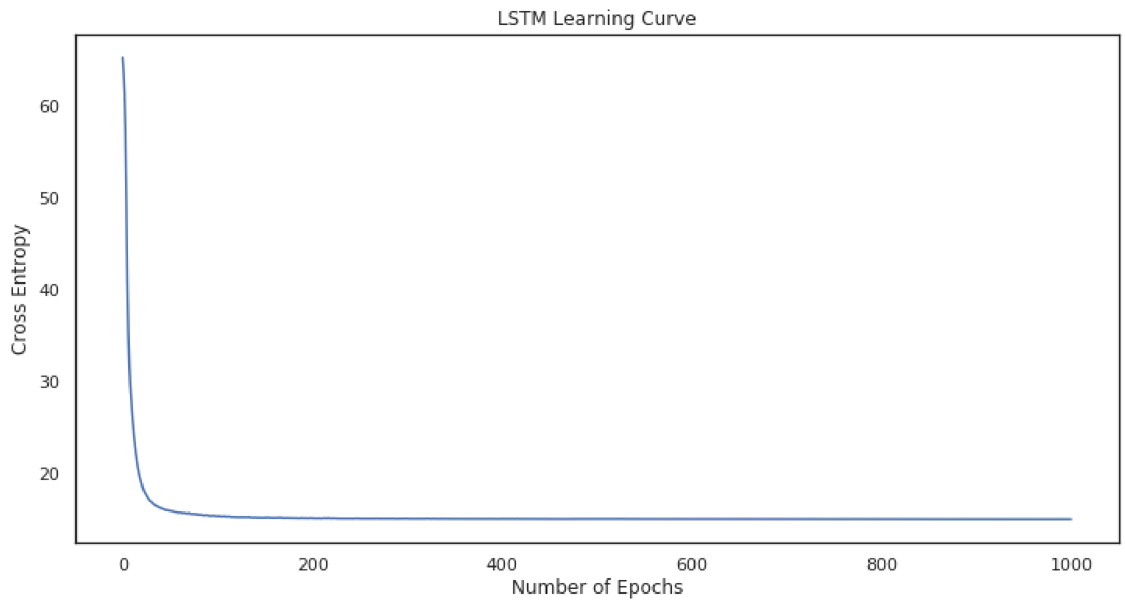
iv. LSTM & GRU

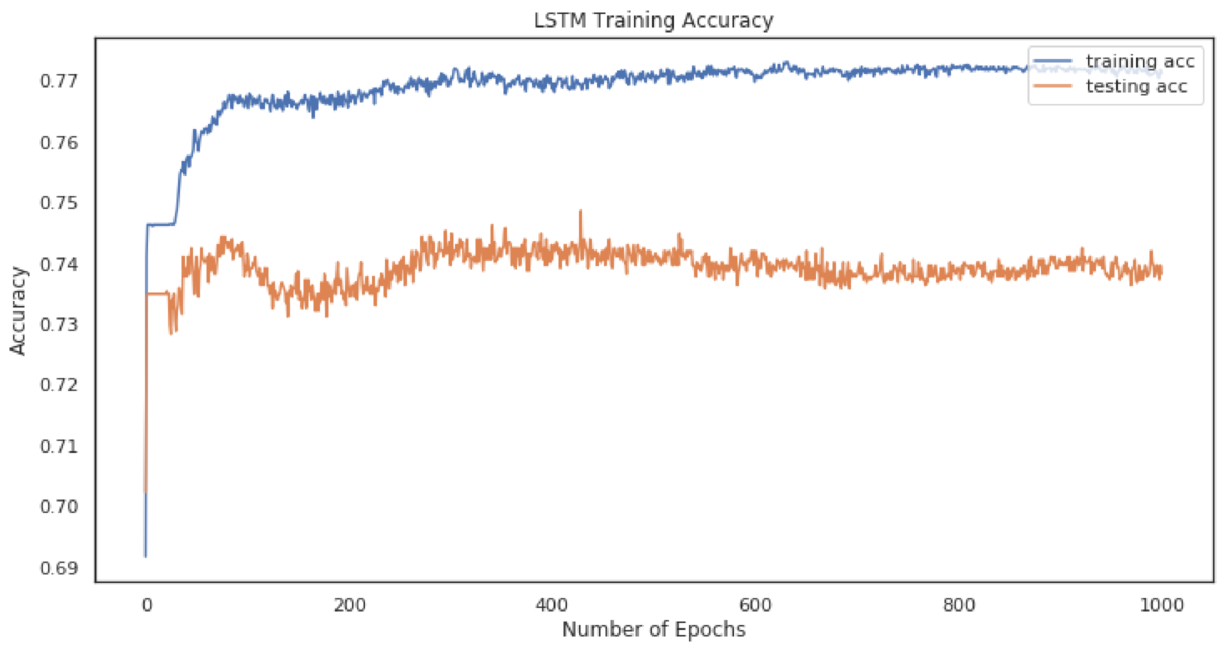
- LSTM
    - ▼ hidden size = 128, num_layers = 1

```
LSTM(
  (lstm): LSTM(1, 128, batch_first=True)
  (out): Linear(in_features=128, out_features=1, bias=True)
)
```

- Training & Testing Accuracy

- ▼ batch size = 128, EPOCH = 1000
- ▼ Training Accuracy: 77.129% (6529/8465)
- ▼ Testing Accuracy: 73.831% (1563/2117)

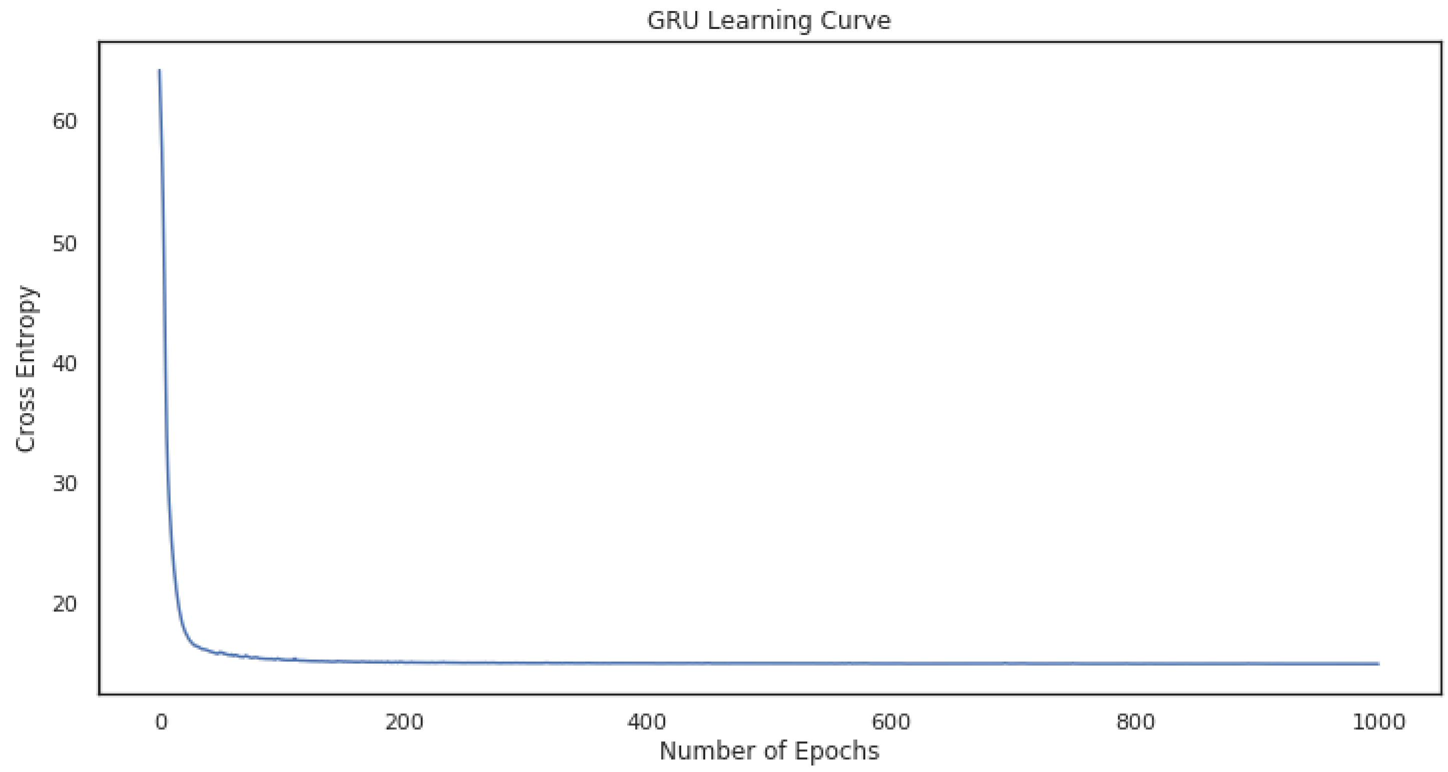


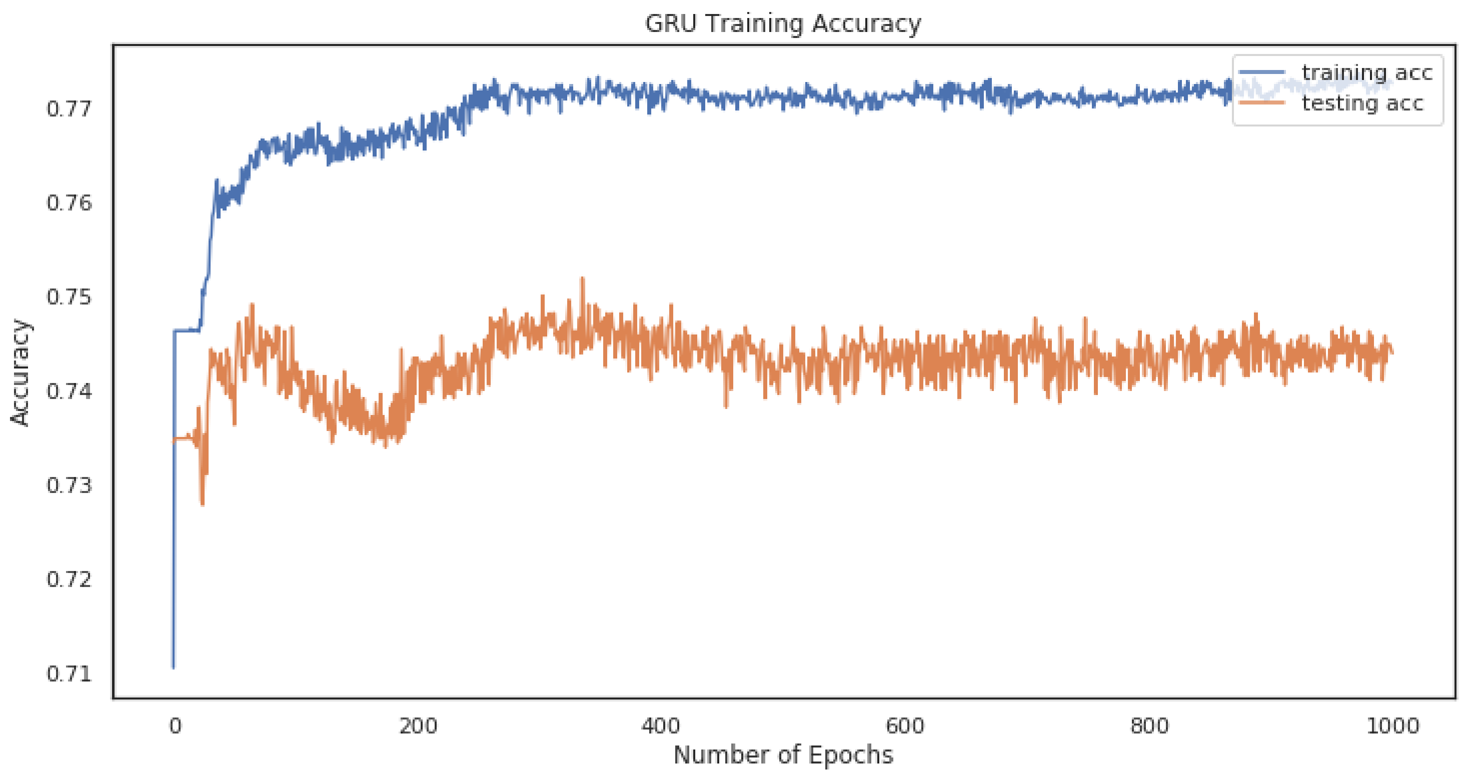




- GRU
  - ▼ hidden size = 128, num layers = 1

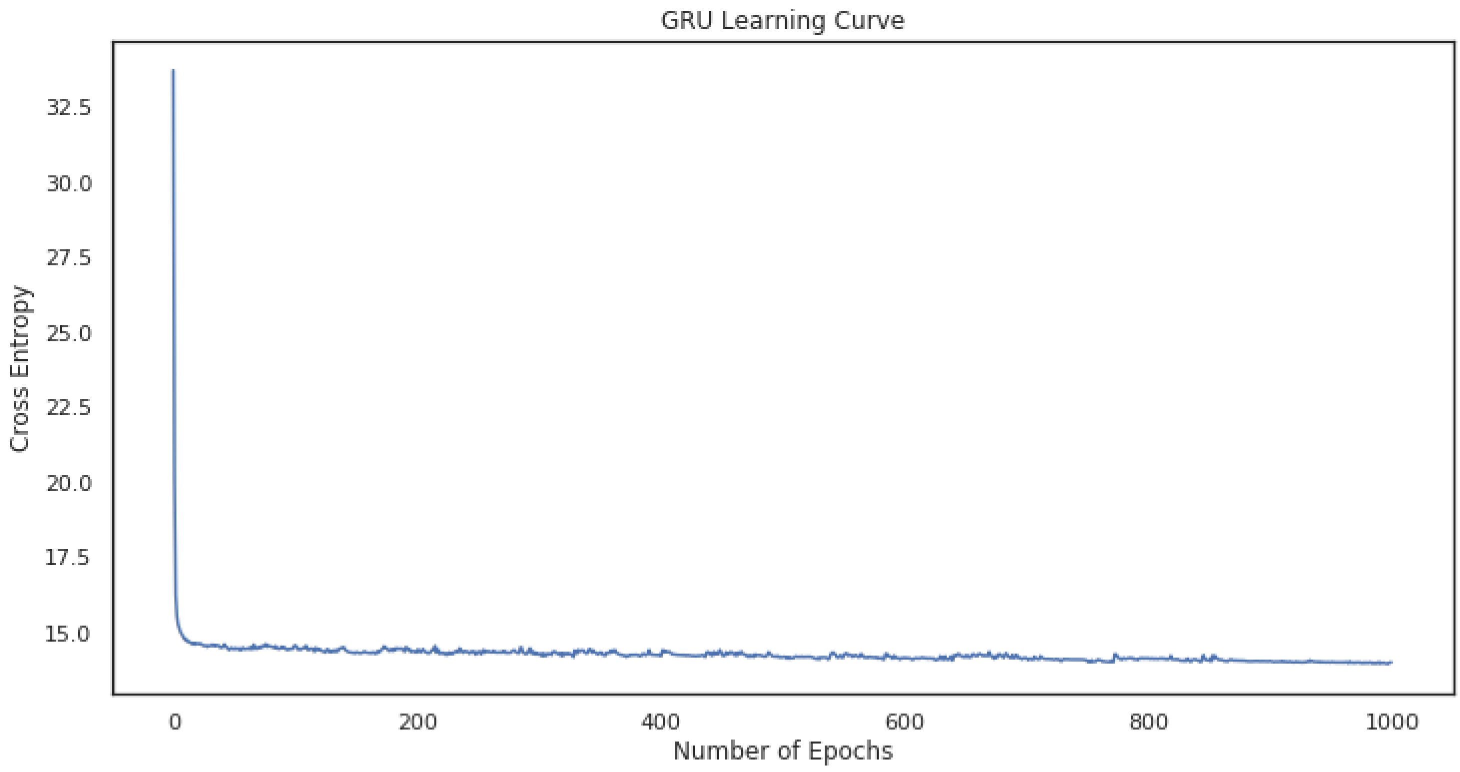
```
GRU(
  (gru): GRU(1, 128, batch_first=True)
  (out): Linear(in_features=128, out_features=1, bias=True)
)
```

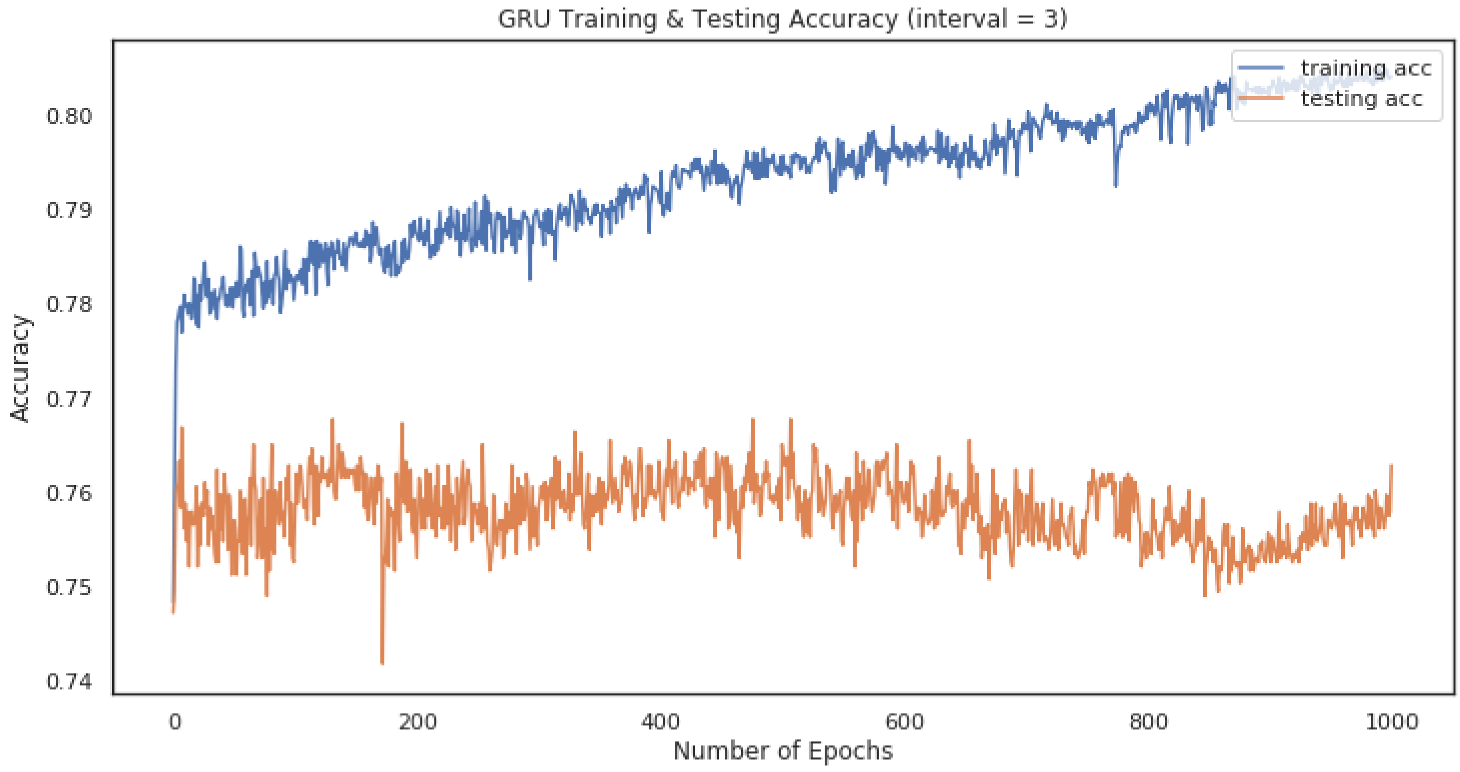
  - ▼ Training & Testing Accuracy
    - ▼ batch size = 128, EPOCH = 1000
    - ▼ Training Accuracy: 77.283% (6542/8465)
    - ▼ Testing Accuracy: 74.445% (1575/2117)



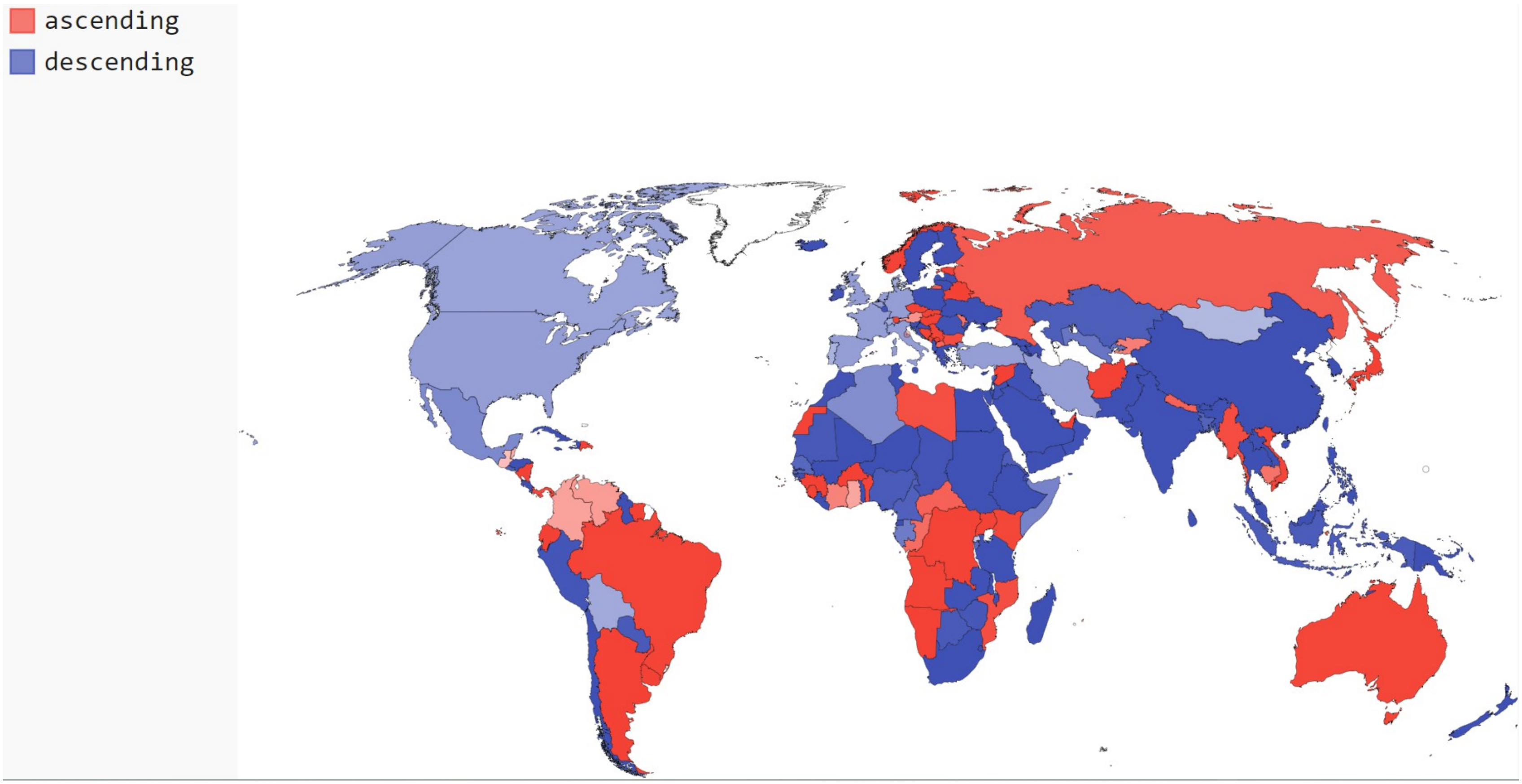




- Changing the value of interval L to analyze its effect on the result
    - ▼ 原本RNN, LSTM以及GRU第一次train的interval L = 7
    - ▼ 重新檢視raw data發現大部分的國家前半部分dataframe的確診數大多為0，故推論interval若取的太大會導致model學習前半部分data的參數沒辦法準確預測後半部分dataframe的確診數
    - ▼ 此外，原本的label = 0有7874筆, label = 1有2117筆數，似乎為imbalanced data
    - ▼ 第二次取interval_L = 3，希望可以改善前半部分dataframe的確診數大多為0導致model預測能力下降的問題。
        - ▼ training data變8923筆, testing data變2230筆
        - ▼ label 0: 6906, label 1: 2675 (2675/6906 = 38.73%)
        - ▼ 此次的GRU表現最佳，超越interval_L = 7時的三個model
        - ▼ Training Accuracy: 80.41%
        - ▼ Testing Accuracy: 76.289%




- Compute the probability for each country and plot on a world map by using ”pygal”



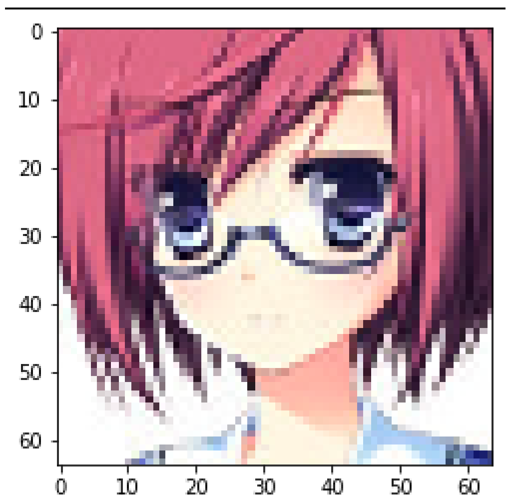
vi. Disucssion

- 第一次實作時，每個國家皆取一筆subsequence + 一個label。若threshold設定為0.7，只會取143筆data，資料非常少。這樣的資料規模下去train RNN/LSTM/GRU testing accuracy都只有55%左右，只比隨機亂猜好一點而已。
- 第二次實作時，為了增加資料的數量，每個國家從index=0, 每次移動一個time step取得subsequence和label。同樣threshold = 0.7的設定下，資料筆數變成10000多筆。雖然仍有imbalanced data的問題存在，但training accuracy以及testing accuracy皆有顯著的提升。
- 第三次實作時，為了解決前半部分dataframe很多0值的問題，故每個國家從index = 40開始取，其餘流程與第二次實作相同。如此取法可以避免model training學了很多連續0的確診數，但面臨非連續0的確診數時預測能力變下降許多的問題。此次實作training accuracy 和 testing accuracy都比第二次實作更好一點（皆接近8成）
- 每次實作皆有試著調整EPOCH, batch size, hidden units, num layers等超參數，但發現其實對於此問題的accuracy並沒有顯著的影響（num layers 多一點有稍微影響一點）。故推論這個問題data preprocessing可能才是提升accuracy的關鍵。

## Q2 Variational Auto encoder for Image Generation

i. Describe in details how to process images (such as resizing or cropping) and design the network architecture

- Image Processing
  - ▼ 全部的照片共有21551張。每張照片都先做normalization ( /255)，再將大小resize成64*64。preprocessing後的shape: (21551, 3, 64, 64)
  - ▼ sample image after preprocessing:

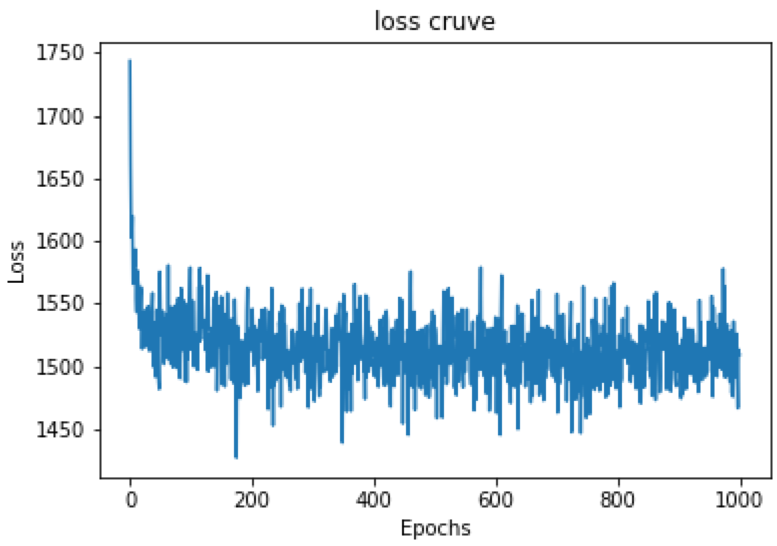



- Model Architecture
  - ▼ 建構VAE class
    - ▼ convolution encode的部分建構兩層。原本想建立3層，但參數太多超過memory limit無法訓練。

```
VAE_1(
  (convolution_encode): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): Conv2d(64, 128, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): ReLU()
  )
  (fc1): Linear(in_features=32768, out_features=256, bias=True)
  (fc2): Linear(in_features=32768, out_features=256, bias=True)
  (convolution_decode): Sequential(
    (0): ConvTranspose2d(128, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
    (3): ConvTranspose2d(64, 3, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): Sigmoid()
  )
)
```

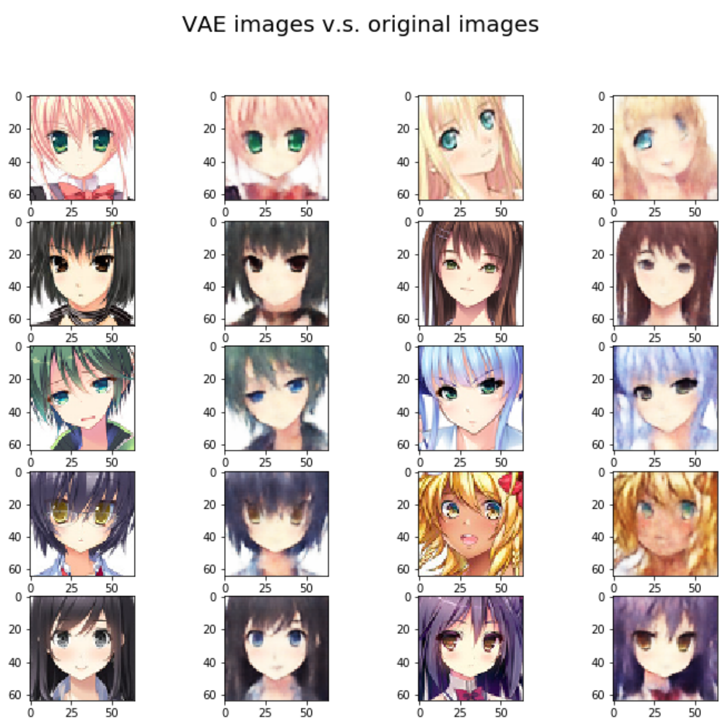
ii. Plot the learning curve of the negative evidence lower bound (ELBO) of log likelihood of training images

- ▼ loss curve of training images:



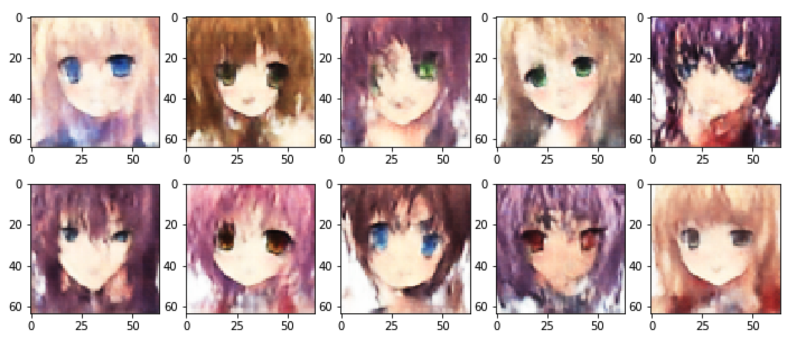

iii. Show some examples reconstructed by your model

- ▼ original images v.s. reconstructed images
- ▼ column=2 為 column=1的reconstruction, column=4為column=3的reconstruction



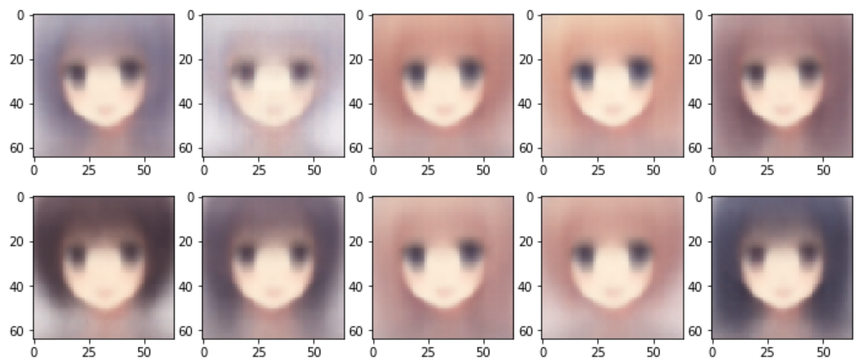

iv. Sample the prior p(z) and use the latent codes z to synthesize some examples when your model is well-trained.

- ▼ Synthesized samples drawn from VAE



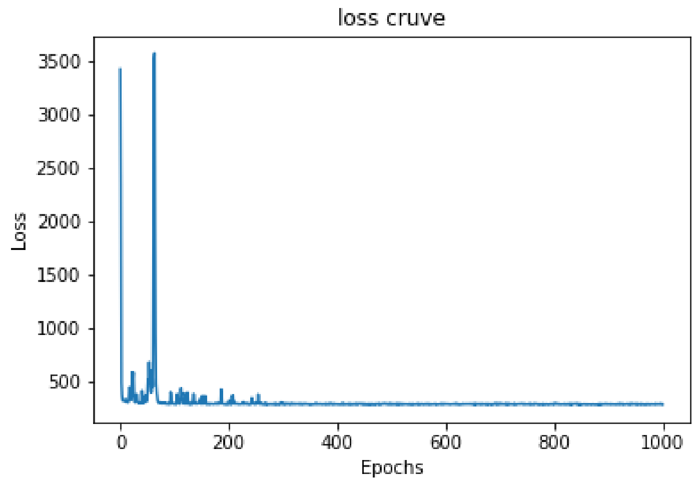
v. Show the synthesized images based on the interpolation of two latent codes z between two real samples.

- ▼ Generated images based on interpolation of latent codes from two real images

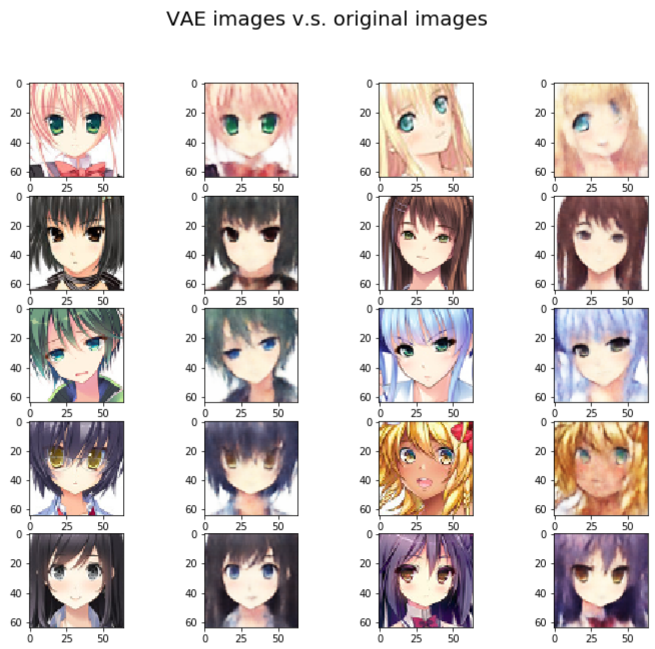


vi. Multiply the KL term in ELBO by 100 in your loss and repeat ii ~ v.

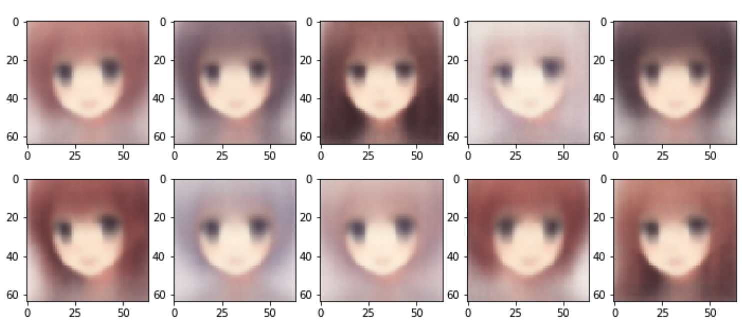
▼ ii. loss curve of training images:




▼ iii. Show some examples reconstructed by your model

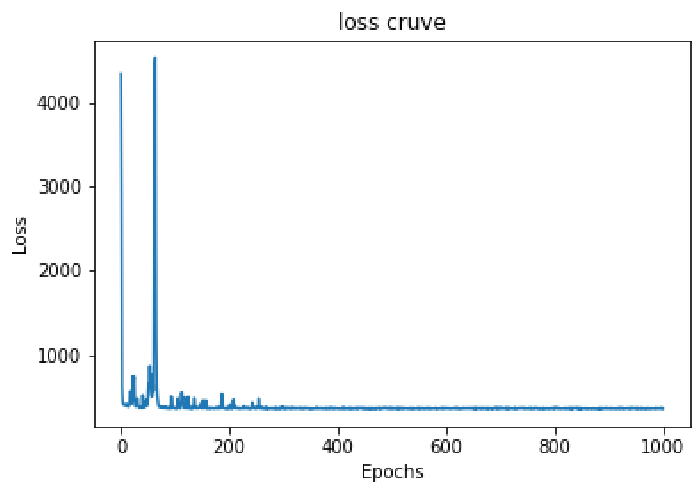



▼ v. Show the synthesized images based on the interpolation of two latent codes z between two real samples.



vii. Multiply the KL term by 0 in ELBO and repeat ii ~ v.

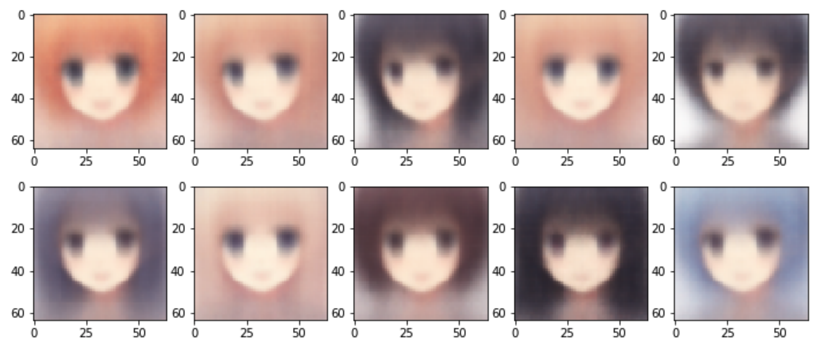
▼ ii. loss curve of training images:




▼ iii. Show some examples reconstructed by your model

VAE images v.s. original images

▼ v. Show the synthesized images based on the interpolation of two latent codes z between two real samples.



viii. Discussion on the effect of KL term

- KL term和reconstruction loss為trade off
- 從本次實驗可以發現KL term越高，reconstruction loss越低
- 從理論上來講，增加KL term即增加ELBO的upper bound，即增加maximum capacity of the encoder channel. ( H-D —> H)
- H-D ⩽ I(x; z) ⩽ R bounds
    - I為maximum capacity of the encoder channel
    - H為empirical data entropy
    - D為distortion
    - R為KL term