DL_2022_HW2 Report

1 News Document Classification

1.1.1 我總共嘗試了三種 tokenize 的方法，分別是 Spacy、Torchtext 的 get_tokenizer(“basic_english”)，以及直接按空格劃分，並分別觀察 tokenize 出來的結果差異。

可以看到按照空格劃分的方式會將句點等等符號也合併在單一個 token 中，Spacy 雖然有把最後的句點分開，但是在”calif.--”之處仍合併為一個 token。相較以上兩種方法，torchtext 的 basic_english tokenizer 則將所有字詞都完好的劃分，因此我選擇使用 Torchtext 的 get_tokenizer(“basic_english”)作為此次實作的 tokenizer。

```
dict_values(['4', ['bea', 'buzzing', 'about', 'beehive'],
['burlingame,', 'calif.--', 'bea', 'systems', 'is', 'expanding', 'the',
'open', 'source', 'beehive', 'initiative', 'but', 'still', 'has', 'no',
'plans', 'to', 'participate', 'in', 'the', 'eclipse', 'open', 'source',
'tools', 'organization,', 'despite', 'the', 'embrace', 'of', 'beehive',
'by', 'eclipse.']])
```

Fig. 1 按空格 tokenize

```
dict_values(['4', ['bea', 'buzzing', 'about', 'beehive'],
['burlingame', ',', 'calif.--', 'bea', 'systems', 'is', 'expanding',
'the', 'open', 'source', 'beehive', 'initiative', 'but', 'still',
'has', 'no', 'plans', 'to', 'participate', 'in', 'the', 'eclipse',
'open', 'source', 'tools', 'organization', ',', 'despite', 'the',
'embrace', 'of', 'beehive', 'by', 'eclipse', '.']])
```

Fig. 2 Spacy_en = English() tokenizer

```
dict_values(['4', ['bea', 'buzzing', 'about', 'beehive'],
['burlingame', ',', 'calif', '.', '--', 'bea', 'systems', 'is',
'expanding', 'the', 'open', 'source', 'beehive', 'initiative', 'but',
'still', 'has', 'no', 'plans', 'to', 'participate', 'in', 'the',
'eclipse', 'open', 'source', 'tools', 'organization', ',', 'despite',
'the', 'embrace', 'of', 'beehive', 'by', 'eclipse', '.']])
```

Fig. 3 torchtext.data.get_tokenizer(“basic_english”)

1.1.2 有時候在 training data 所建立的 word vocab 中並不一定有辦法涵蓋所有的 word，當遇到沒有看過的 word 時就用<unk>的 token 代替，即代表這個 word 是 unknown 未看過的。

<pad>是用作填充長度的作用，在 training 或 testing 的時候容易碰到 input sequence 的長度不一，就使用<pad>的 token 將長度補齊。

1.1.3 在這次作業中，我的 text data proprocessing：

1. 使用 torchtext.data.get_tokenizer(“basic_english”)進行 tokenize

2. 把 token 轉為 lower case
3. 建立 vocab，設定 min_freq=1，因為大多數的 word 好像都只出現一次，調大的話在 vocab 中會少很多字。

1.2.1 -

1.2.2 -

1.2.3 在 transformer 之中，我設定 emsize = 3000(embedding dim to FC)、d_hid = 128(dim of FC in nn.TransformerEncoder)、nlayers = 4(number of nn.TransformerEncoderLayer in nn.TransformerEncoder)、nhead = 4(number of heads in nn.MultiheadAttention)。其中 nhead 的部分，在嘗試過 2~6 後發現 4 的表現最好，是最適於此次作業的 head 數。

2 VAE for Image Reconstruction

1.

1.1 Anime：

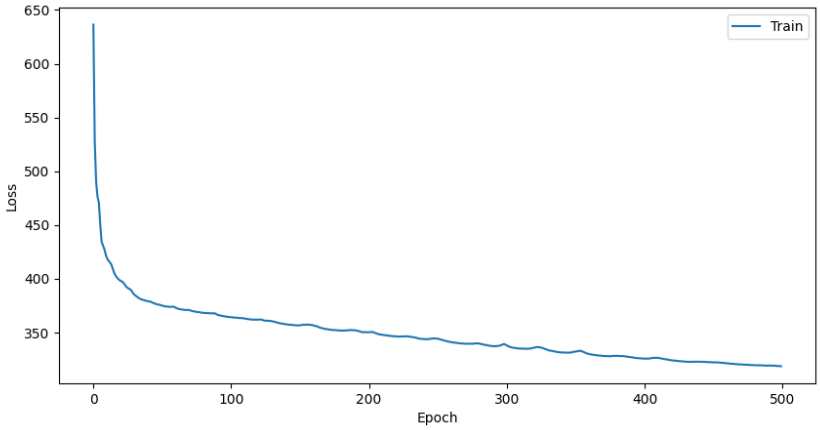


Fig. 4 Learning curve of VAE on anime

Fig. 5 Real data

Fig. 6 Reconstructed sample

1.2 Mnist：

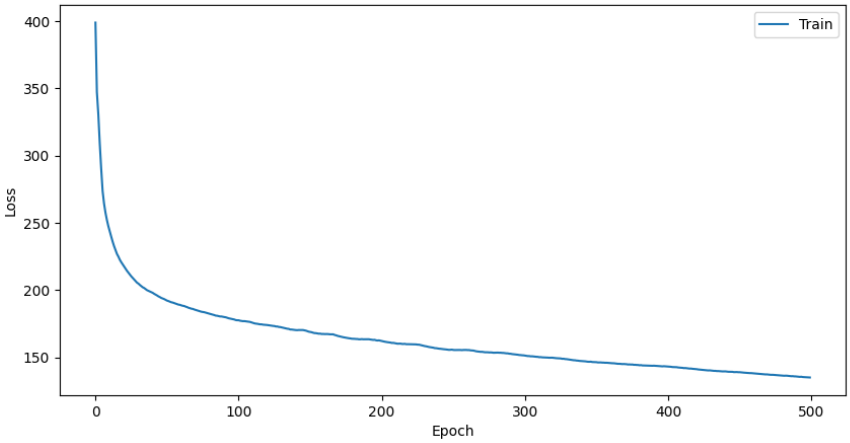


Fig. 7 Learning curve of VAE on mnist

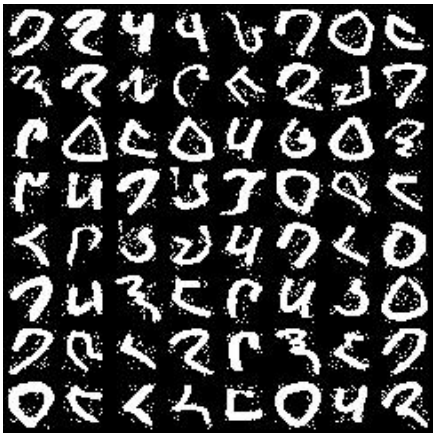


Fig. 8 Real data

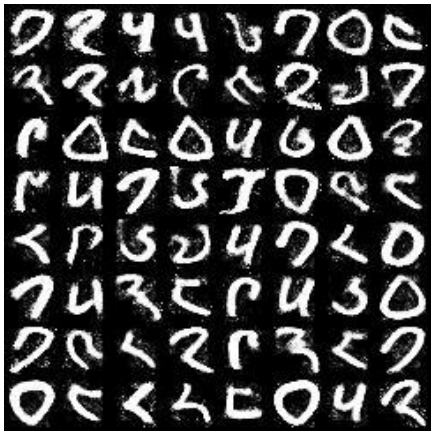
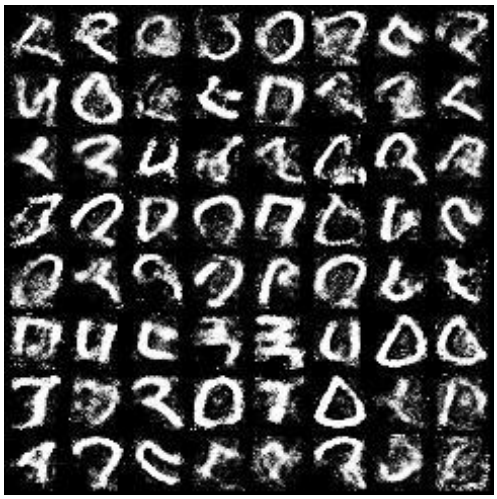
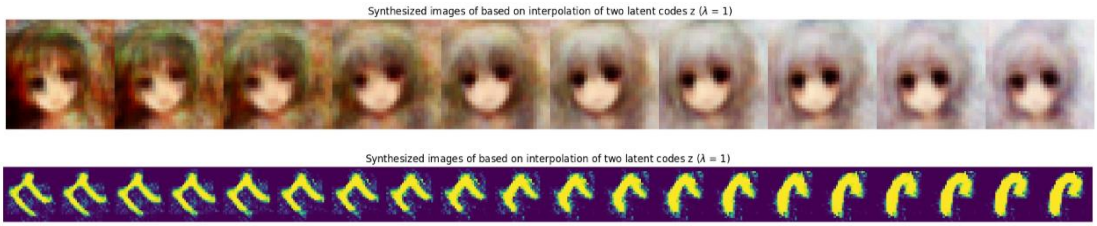


Fig. 9 Reconstructed sample

2.

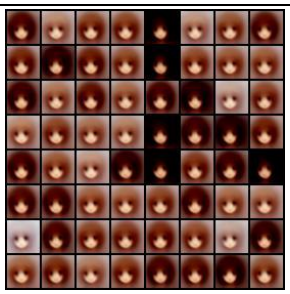
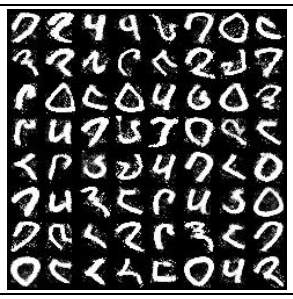
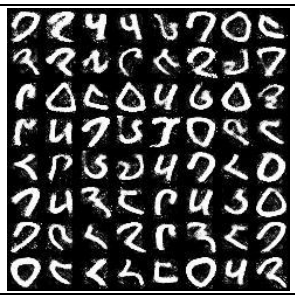
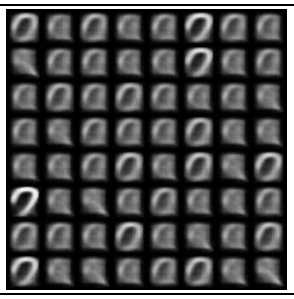
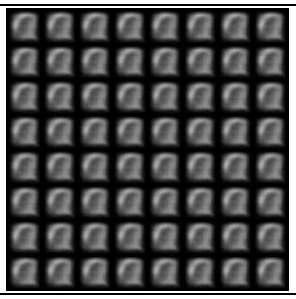


3.

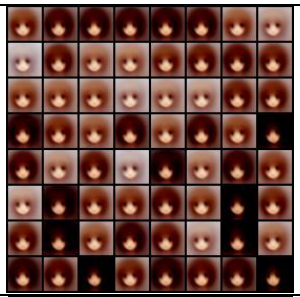
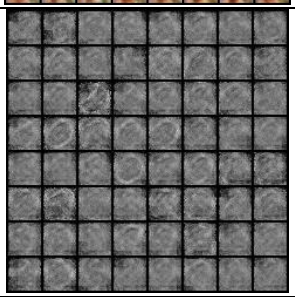
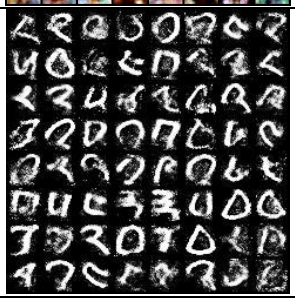
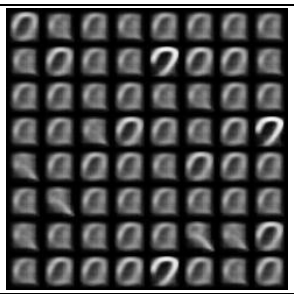
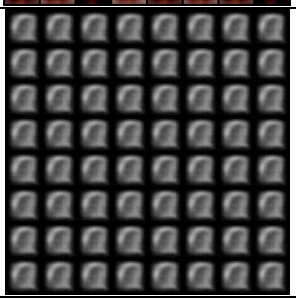


4.

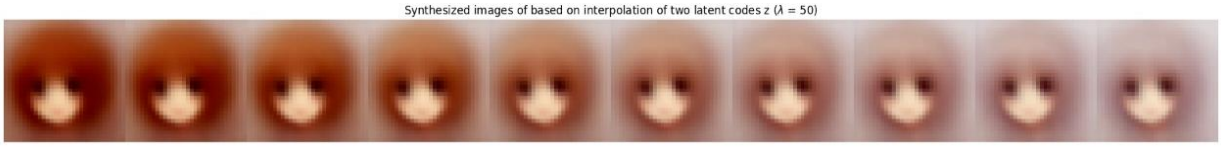
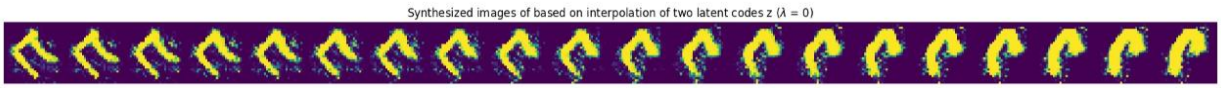
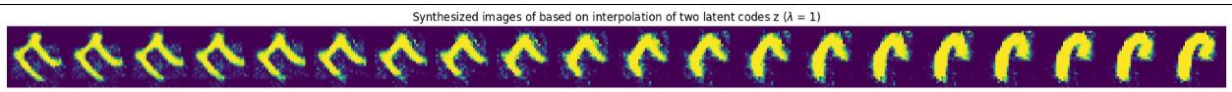
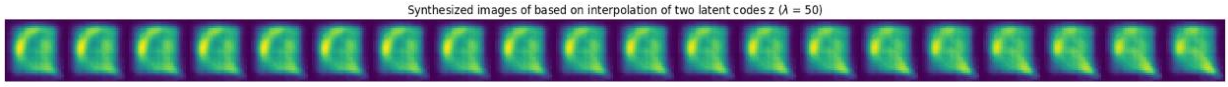
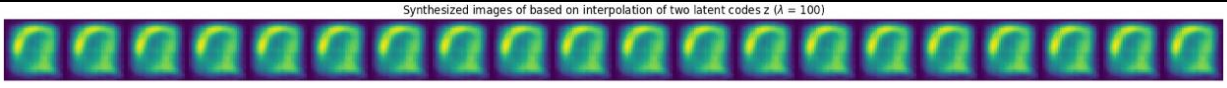
Step 1.

λ = 0	λ = 1	λ = 50	λ = 100
			
			

Step 2.

λ = 0	λ = 1	λ = 50	λ = 100
			
			

Step 3.

Anime λ = 0	
Anime λ = 1	
Anime λ = 50	
Anime λ = 100	
Mnist λ = 0	
Mnist λ = 1	
Mnist λ = 50	
Mnist λ = 100	

λ 在 loss function 之中主要是控制 KL divergence 那一項的重要程度，讓 Encoder 所產生的 latent 盡可能與 prior distribution 相近，意義是假設今天可以將 Encoder 所產生的 latent 結果跟 prior distribution 一模一樣，那就可以直接從 prior distribution 中 sample 出 noise 並餵給 Decoder 生成圖片即可。

那在 λ = 0 之下，由於沒有了限制 latent 的項，每張生成的圖片看起來都長一樣。但在 λ=100 情況之下，對於 latent 來說要與 prior distribution 盡可能非常接近這件事比較重要，反而 reconstruct 圖片的重點消失，從上面範例可以看到 λ=100 中 reconstructed 圖片的生成品質不太好。