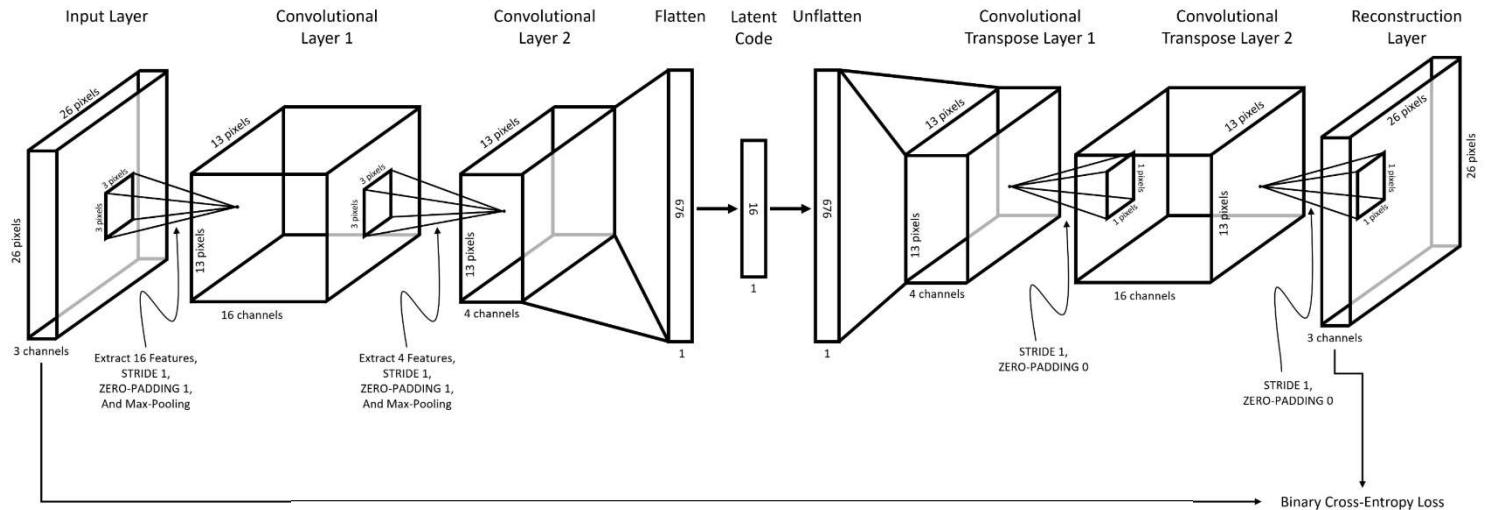


Homework 2

10910COM 526000 Deep Learning 王傳鈞 109062631

第一題

關於我的 convolutional autoencoder (CAE) 架構，可以參考下圖：



圖一：架構圖

在訓練過程，我採用 mini-batch SGD 的方法來更新模型參數，因此一次輸入模型的資料實際上是具備維度 (batch_size, 3, 26, 26)，上圖為了簡單易懂，所以只畫出丟入一筆資料的完整流程。

我的模型總共有兩個 convolution layer (包含各一次的 max-pooling)，與兩個 convolution transposed layer。經過一開始的兩次卷積，我們可以得到一個具有維度 (4, 13, 13) 的 PyTorch tensor，接著我們將它展平成為 (676, 1) 維的向量，並通過一個 fully connected linear layer，就得到了只有十個維度的 latent code。

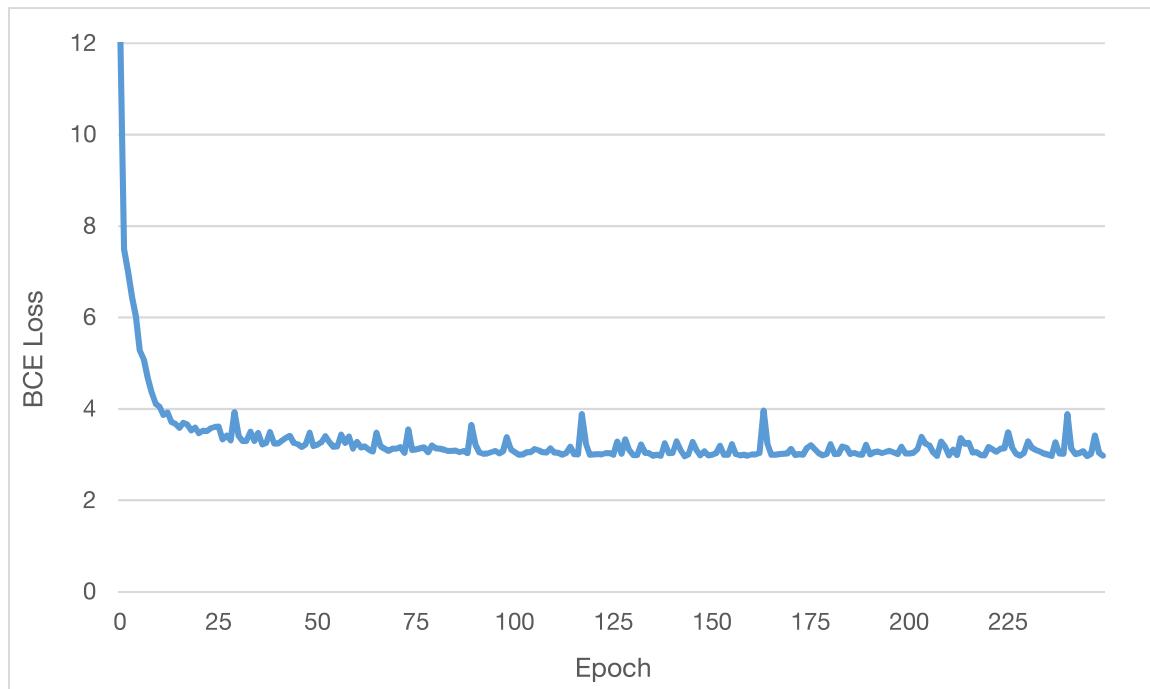
如果是在訓練模式之下，則此 latent code 會經過 CAE 後半段完全與前半段對稱的解碼過程，還原回原本的 (3, 26, 26) 維度之 PyTorch tensor。

比對原始輸入 PyTorch tensor 與 CAE 吐出的還原版 PyTorch tensor，我們可以計算出 binary cross-entropy loss (BCE loss)，進而進行最重要的 CAE backward propagation，調整模型參數、增加還原 PyTorch tensor 的準確率。

如果進入到生成參雜高斯雜訊的模式，則 CAE 會針對每一個 PyTorch tensor 的 latent code，加入服從標準常態分佈的亂數雜訊向量，接著再把它丟回 CAE 後半段的解碼器，還原出與原始輸出有點不同的 PyTorch tensor。

第二題

下圖為某一次的訓練過程所輸出的 BCE training loss。



因為我的模型採用 mini-batch SGD 方法訓練與更新參數，所以每一次執行訓練模型所得的 BCE loss 不盡相同。若助教想要節省執行 python script 的時間，我在資料夾有提供使用 GPU 平行運算，來取得的最佳 BCE loss（約為 1.0603）pre-trained model，只需要按照 readme.md 第三步驟所述輸入「Y」，即可使用最佳模型來生成 gen_data.npy 與處理其他流程。

第三題

Class	From data.npy	Reconstruct by CAE	Generated 5 Samples
Center (0)			
Donut (1)			
Edge-Loc (2)			
Edge-Ring (3)			
Loc (4)			
Near-Full (5)			
Random (6)			
Scratch (7)			
None (8)			