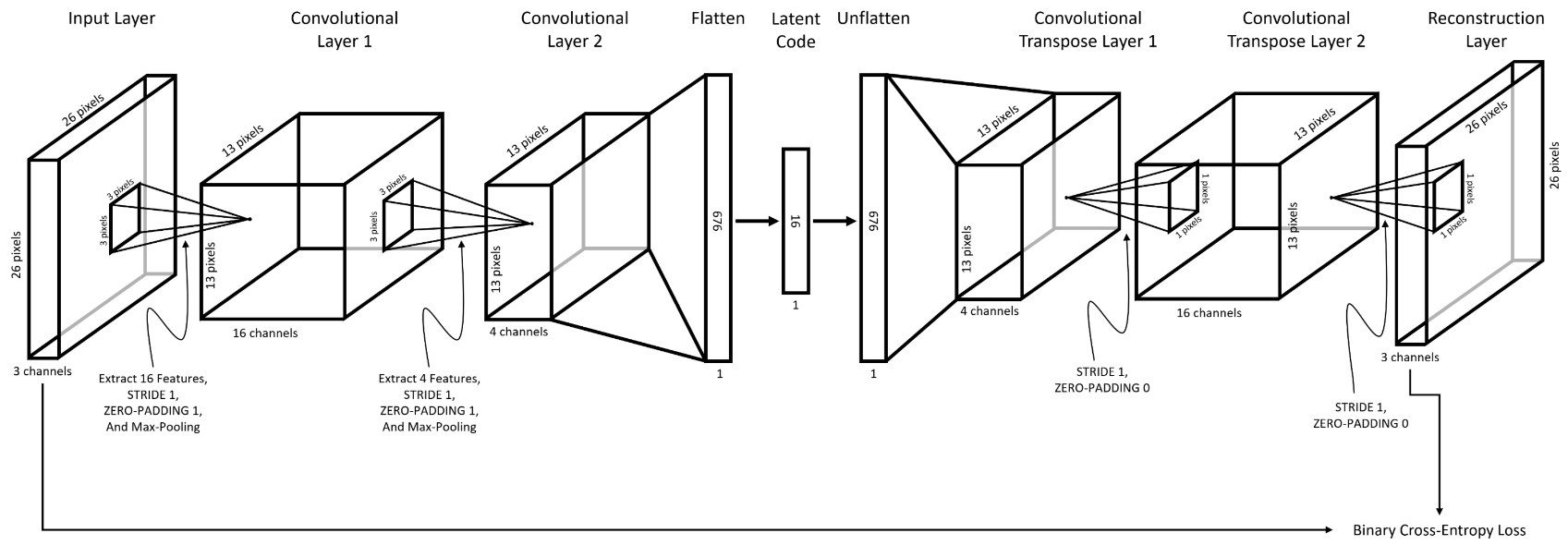
Homework 2

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

