* Report:
* Introduction:

在lab4中，我們要利用，Seq2Seq model搭配CVAE架構來實作英文單字的時態轉化，及利用CVAE的latent space中抽取出 Gaussian noise，利用那個noise來生成出，隨機的單詞再依該單詞推斷出相應的四個英文時態。

CVAE的全稱為conditional variational autoencoder ，是由最基本的autoencoder(AE)，所搭建成的，AE是由一個encoder和一個decoder所組成的，中間則有一個latent space作為兩者之間的，緩衝當作特徵空間，latent space的維度通常相較如輸入及輸出資料是較小的維度。然AE最基本的的功能就像是對維度進行壓縮在重新解壓縮，讓輸入與輸出有著相同的意義。

但是此時無法對壓縮至latent space的vector做控制，因此我們需要在AE的基礎加上限制，也就是variational autoencoder (VAE)，VAE迫使encoder的輸出為兩個相同維度的向量，分別代表mean 和 standard deviation這樣我們就可以利用這兩個向量，來算出normal distribution，有了normal distribution就能夠對latent space中的vector有著更近一步的了解，而理論上有個分佈的資訊，就可以控制要送入的decoder的latent space vector，進而造成能夠間接的控制所想要的輸出。

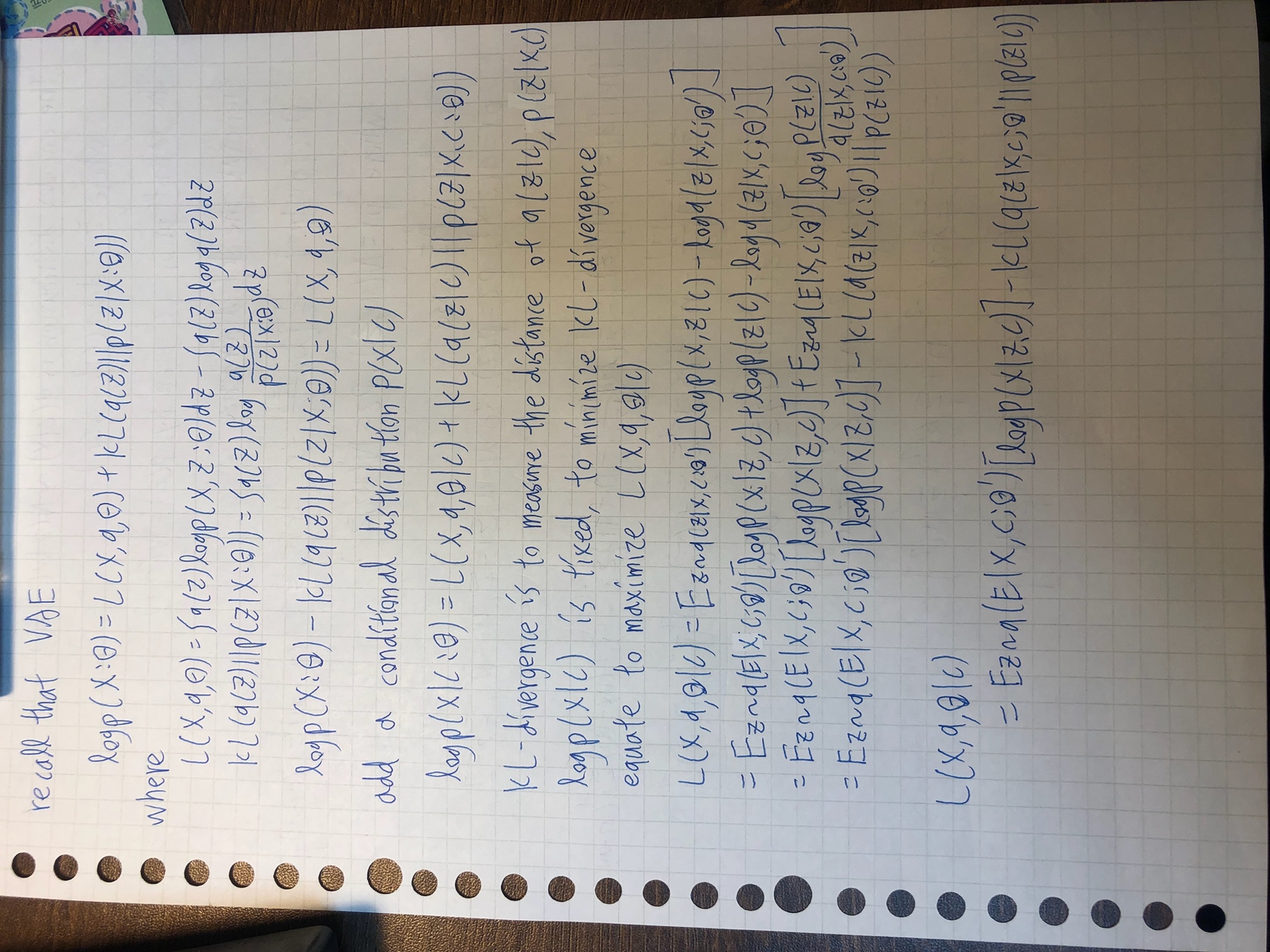
但是這樣還不夠，在某些情況下我們會希望，encoder和decoder可以將所送入的vector帶上condition (label)的資訊，這樣在生成時可以有更多的可控因素，在這個背景之下，就有CVAE的出現，將輸入和相應的condition一起用encoder編碼至latent space中，任意抽出的noise加上condition送回decoder就可以產生出，帶著condition的輸出結果了。

而在此LAB4作業中，在大架構CVAE下，因為是處理NLP的問題，所以我們使用Seq2Seq的模型架構，所以encoder和decoder是用RNN所建構而成。

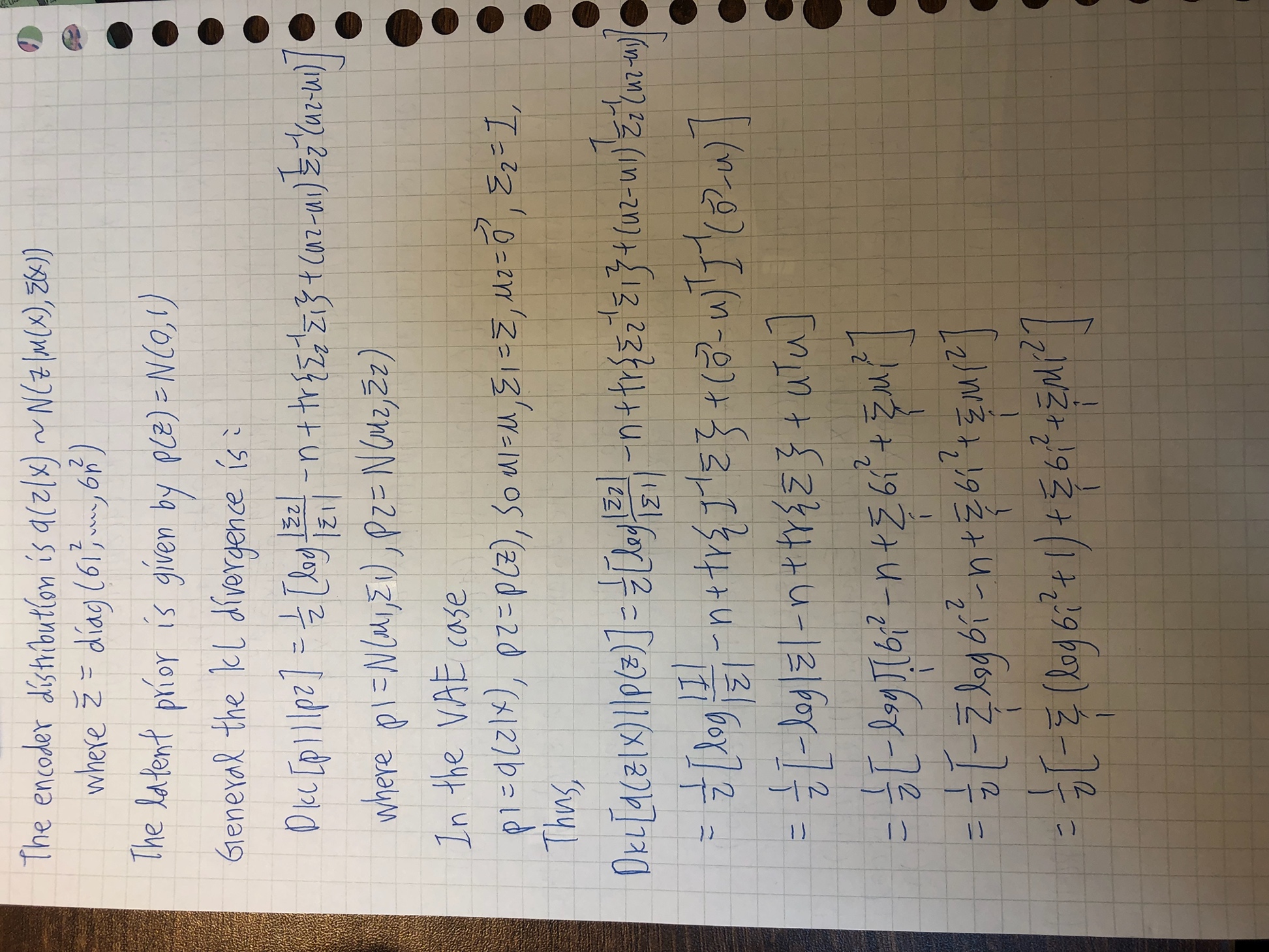
在處理NLP問題中，我們通常會使用embedding的技巧來處理sequence和condition的輸入與輸出，embedding簡單來說是將已知所需的輸入種類，鑲嵌至給定維度的向量，這樣一來，我們所使用的文字就可以變成有限維度的vector，方便我們將此用來送入至NN類的模型當中做訓練。

encoder的輸入為先經過embedding處理過的sequence和condition，decoder的輸入則embedding處理過的sequence（如有需要）和condition及z，而在這個問題中，sequence就是英文單詞，拆成個別的字母再送入embedding中，30維（26個字母加上特殊token）至給定的latent space dim，condition也需要送入embedding中4維（四個英文單詞時態）至給定的condition latent space dim。

* Derivation of CVAE:

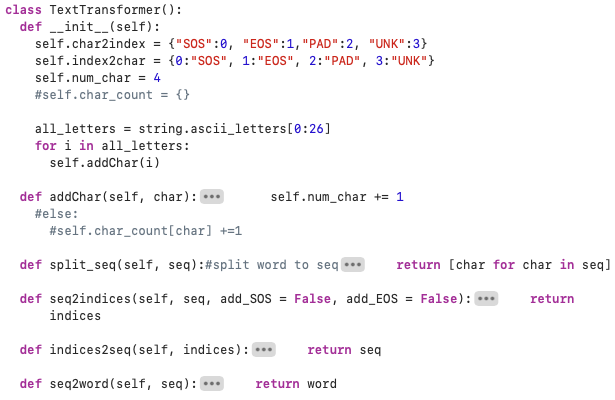


* Derivation of KL Divergence loss



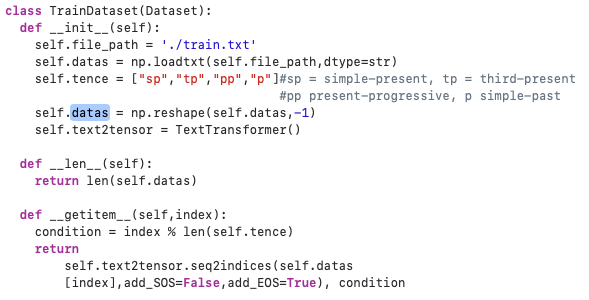
* Implementation details:
  + Describe how you implement your model (encoder, decoder, reparameterization trick, dataloader, etc.)

在處理英文單詞時，我們需要一個class用來處理英文單文對後續所需要用的embedding機制先預作處理，例如：將英文單詞拆分成個別不同的字母，再將每個對應的字種給予相應的編號，好以用來做embedding使用。而反正當model輸出時，我們也需要當所對應的數字轉回原來所對應的字母，再將字母組合而成英文單文。 這就是我們所需的基本功能。如下圖一為實作，其中以下四個特殊token來滿足某些需求。



圖一，TextTransformer，文字處理基本單元。

我們將dataloader分為兩個類別來撰寫，分別為處理訓練集的TrainDataset以及處理測試集的TestDataset，而基本上兩者都沒有什麼特別之處，只是將.txt檔一行一行讀入，在針對每一行分割出所需要的字串來做使用，實作如圖二、圖三。

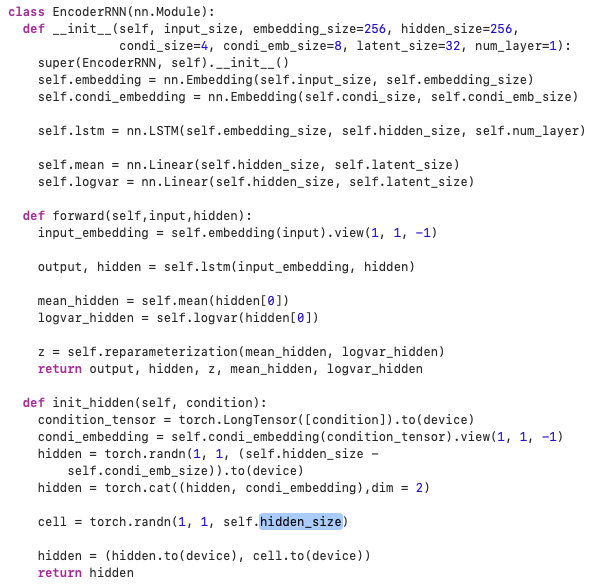


圖二、處理訓練資料集，並給予特定位置對應的時態。



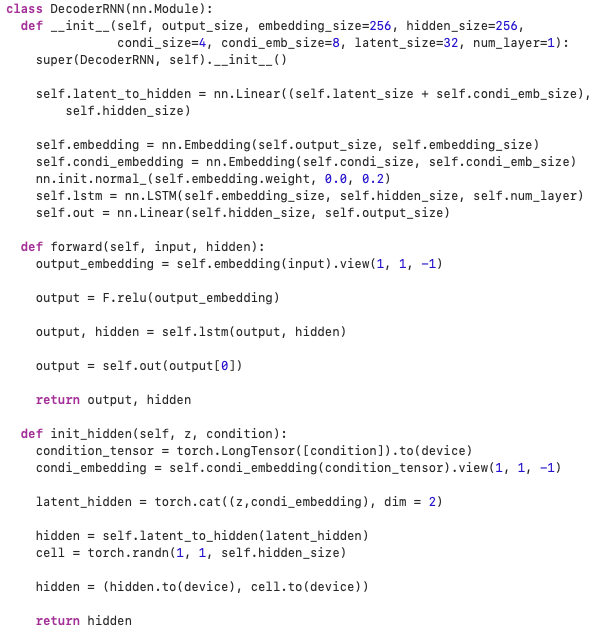
圖三、讀出測試集中的單詞，並給予時態條件用於轉換。

Encoder包含了兩個分別負責處理字母和時態條件的embedding及基本的LSTM用來實作RNN結構，還有兩層linear fc用來迫使encoder生成出mean和standard deviation，好以用來時生成有著normal distribution性質的latent space。而在每次每個英文單詞送入前，我們需要初始化我們的hidden，除了初始化外，也可以先嵌入所需要的condition。具體細節如下圖四。



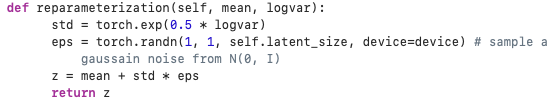
圖四，EncoderRNN。

Decoder也包含了處理包含了兩個分別負責處理字母和時態條件的embedding及基本的LSTM用來實作RNN結構之外，多一個linear fc 用來將輸出最有可能的輸出字母或token，其中當我們需要運行decoder時，如同encoder初始化hidden一樣，我們也需要先初始化decoder的hidden，但與之不同的是，我們除了要初始化參數外，還需要將從latent space 抽出來的vector z與embedding後的condition作為hidden的一部分，好以用來生成（重構）我們所壓縮的特徵。具體細節如下圖五。



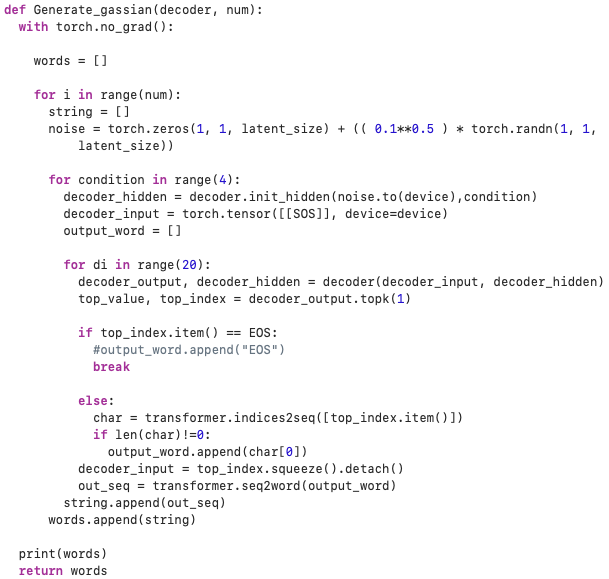
圖五，DecoderRNN

在VAE中我們將latent space 的vector 讓其有normal distribution的性質，但由時隨機抽取出的z我們很難再反向傳播中找出mean和logvar的關係再加以學習調整。這時候就可以使用reparameterization（重新參數化）的技巧，簡單的話，我們可以藉由提供一個epsilon將z改寫成有著對mean和standard deviation相關的線性關係。如此一來，從latent space 抽出的vector z中的隨機性就可以轉移至normal distribution的變量也就是mean和standard deviation，模型就可以學習如何對高斯分佈的平均和標準差做出調整。具體實作如下圖六。



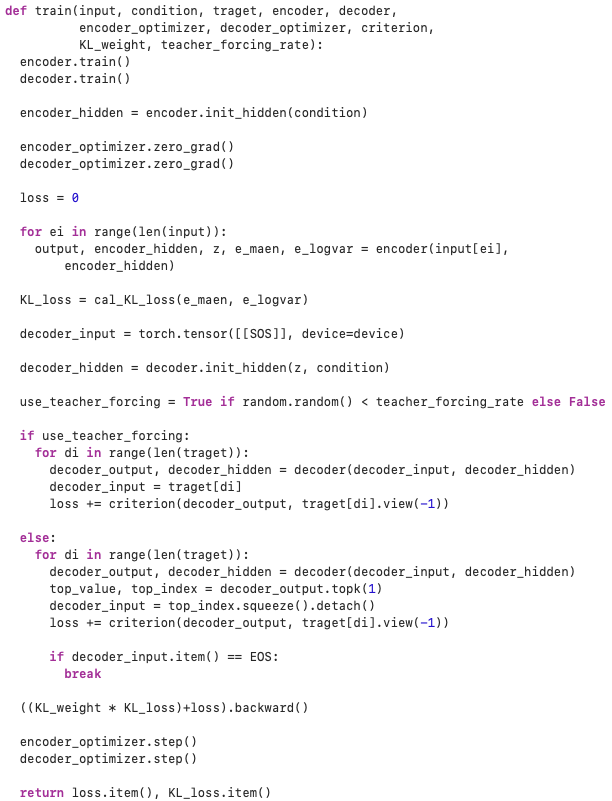
圖六，reparameterization trick。

在做text generation 時，我們需要先對latent space中隨幾抽出一個向量z，在實際的物理意義中抽出來的z可能就代表著一個英文單詞被encode（編碼）至latent space的一個位置中。所以我們就可以利用這隨機抽出的向量（等同於隨機抽出一個單詞），加上condition條件用來生成我們所需要的單詞。Text generation 實作如下圖七。



圖七，text generation。

在train的部分做基本的train單位就是送入一個單詞和他對應的時態詞性，一樣先針對encoder做初始化hidden順便將condition嵌入進去，在依序單詞的每一個token送入至encoder LSTM中，輸出z。decoder再將z與condition初始化，此詞decoder的token輸出可以選擇是否要使用teacher forcing就是先利用標準答案作為decoder的輸入，因爲當decoder還沒訓練好時，decoder會亂猜不利於訓練。Teacher forcing可以改善順練的速度及品質。Train的實作如下圖八。



圖八，train。

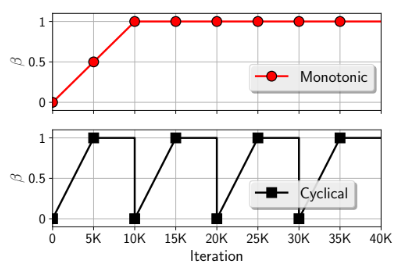
在先前的證明中，我們可以知道KL divergence分別為p和q的分佈的距離，其中p為encoder所生成出的分佈，相關的參數為mean和standard deviation。而q為normal distribution為Ｎ(0,1)其中，我們想要另encoder所生成的向量是符合normal distribution的，所以我們才需要有個loss來評估兩者的差距，由上述證明可以得知加上q的N(0,1)帶入，我們就可以得到證明出的式子。就如同下實作如圖九。



圖九，KL divergence loss。

* + Specify the hyperparameters (KL weight, learning rate, teacher forcing ratio, epochs, etc.)

KL weight:如上所述，我們需要計算p和q的距離，但一開始模型還沒訓練好時，兩者的分佈的距離很大，如果完整個KL divergence loss 合併至loss一同計算會使得loss失焦不利於訓練，因此我們可以乘上一個權重weight做調整，而常見得作法分為monotonic和cyclical，如下圖十所示。



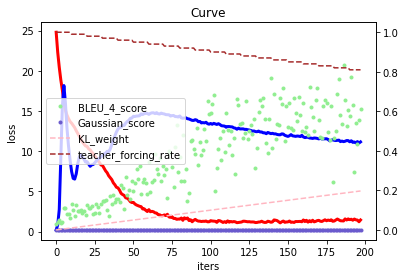
圖十，KL weight annealing。

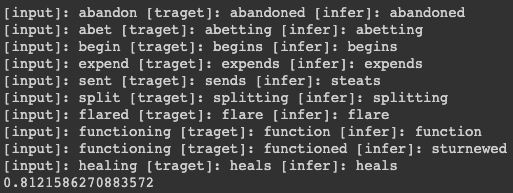
Learning rate:1e-3。

Teacher forcing ratio:如前所說，我們需要一個Teacher forcing得機制來幫助decoder用來學習，而這個Teacher forcing ratio 可以和iters做掛鉤，這樣就可以挑整在當下的iters我們需要多少比例的英文單詞是用正確答案來幫助decoder做訓練。

Epochs:200。

* Results and discussion
  + Show your results of tense conversion and generation and Plot the Crossentropy loss, KL loss and BLEU-4 score curves during training



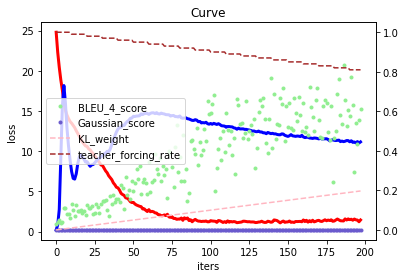




Gaussian\_score = 0.0

其中在實作text generation的時候不知道是哪個部分做錯了，抽出來的單詞有時候，不是同一個詞且不會根據所給予的condition做相應的變化。

* + Discuss the results according to your settings of teacher forcing ratio, KL weight, and learning rate.



在此圖中紅色實線為loss，藍色實線則為KL divergence loss，在這張圖可以觀察到，當KL\_weight緩慢成長時，BLEU\_4\_score的整體趨勢似乎有跟著向上成長，而KLD\_loss在一開始前幾個epoch會突然爆高，後來再慢慢下降，再往上提升，這邊可以猜測應該是一開始模型的encoder的輸出的分佈與normal distirbution差距非常大加上還沒有整握encoder的輸出是mean 和standard deviation和normal distribution的關係。

而整體得loss還也是隨這訓練過程緩面下降，雖說KLD\_loss的數值很誇張，但因為我們有KL weight的機制，所以相乘之後也不至於會令loss有太高的飆升。

在實驗過程中，我有助益到倘若沒有Teacher forcing ratio 的機制，decoder的訓練效果會非常差，而且也很難訓練成功，單靠encoder所編碼的z也無法生成對應的sequence。

這邊因為我是用epoch作為訓練及繪圖得單位，換句話說我要看過4096的單詞與其對應的condition還是一個完整的epoch，而在這個一次訓練單位為一個單詞加condition的情況之下，難免還是不夠細緻的，因此很多情況，可能無法很順利的觀察出來，這倒是我的疏忽，在最後繪圖時才想清這個關係，可惜因為現實的時間壓力沒辦法重新調整及重新訓練。