Lab5 Conditional Sequence-to-sequence VAE

0516054 劉雨恩

1. Introduction

本次實驗目的為架構 Conditional Sequence-to-sequence VAE 神經網路來得到一個單字的不同時態(現在式、第三人稱、現在進行式、過去式)。需要完成的地方為：建立自訂的 Dataloader、架構 Conditional Sequence-to-sequence VAE 神經網路(Encoder 和 Decoder)以及以 loss、KL loss 和 BLEU-4 score 評估神經網路表現。其目標有二：一為測試 testing data，達到平均 BLEU-4 score 有 0.7 以上的表現；二為從 Gaussian distribution sample 得到一個字的四種時態。

2. Implementation details

A. Describe how you implement your model.

- Dataloader
  - 處理data file

    將train.txt從一行四個時態重新處理成四行一種單字加上它的時態index，並重複一次。(0: simple present, 1: third person, 2: present progressive, 3: simple past)

    ex. abandon abandons abandoning abandoned

    → abandon 0 abandon 0

    abandons 1 abandons 1

    abandoning 2 abandoning 2

    abandoned 3 abandoned 3

    將test.txt處理成每個字後面都加上它的時態index。

    ex. abandon abandoned → abandon 0 abandoned 3

  - load data

    在每一個iteration都以list讀入一行data，並且index 0和index 1的元素為encoder的input data和condition，index 2和index 3的元素為decoder的input data和condition。

    而為input data的單字(index 0, 2)都會經過以下處理：

    1. 將單字依每個字母切開

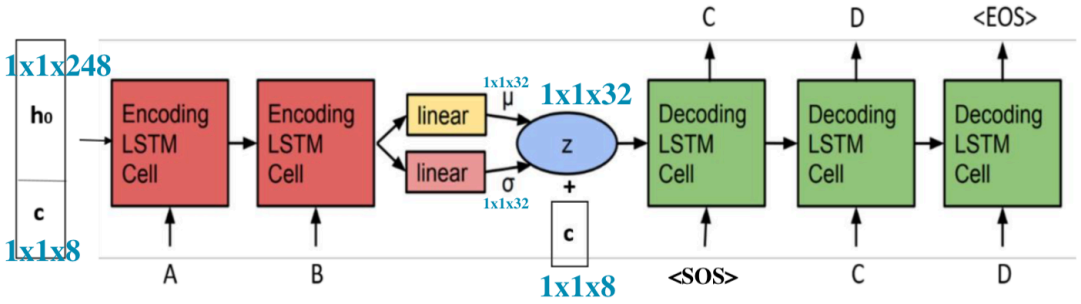
2. 將字母轉為token存入同一個tensor

   (0: SOS token, 1-26: a-z, 27: EOS token)

3. 加上EOS token

training的每一個epoch，單字讀進來的排序是random的。

- CVAE Structure



- Encoder

```
for di in range(input_len):
    en_hidden = encoder(input_tensor[di], en_hidden)
```

將一個字的字母一個個放入，每次都會經過encoder network，其出來的hidden unit為下一個字母的hidden unit。而最後一個字的hidden unit經過Reparameterization Trick處理會成為decoder的latent input。

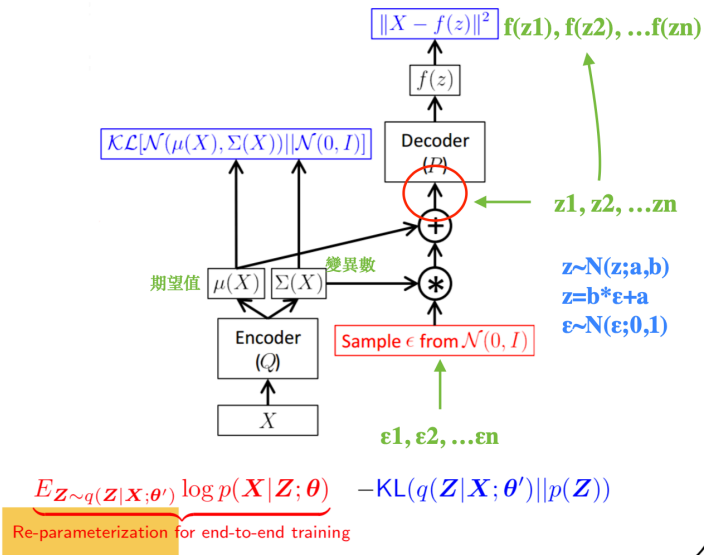
■ Input data

input有二：hidden units(with condition)和input data。hidden unit會先initial成一個1x1x248且數值皆為0的矩陣，然後會和經過condition的nn.Embedding(4,8)的condition concat。input data的方式為將字母一個個放入，而後會經過nn.Embedding(28,256)。其RNN的hidden size為256。

■ Forward

input進來的hidden units和data會經過encoder的nn.GRU，並得到output和下一個input的hidden units。

■ Reparameterization Trick



為了能夠成功計算gradient，encoder ouput出來的latant code需要經過Reparameterization Trick。

encoder的mean和log variance分別是由將最後一個字母出來的hidden unit input到不同的nn.Linear(256, 32)網路生成而來。而後在N(0,I) sample出ε(1x1x32)，將它進行exp(log variance/2)+mean的運算，變為decoder的hidden unit input。

- Decoder

將目標字的字母一個個放入，每次都會經過decoder network，下一個字母的hidden unit依照是否有teacher forcing，來決定是target的字母還是hidden unit。而最後出來的ouput是每個token(SOS, a-z, or EOS)的機率(size: 28)。

- Forward

  處理input的方式和經過的network種類相同，只是每個單字第一次forward時要經過nn.Linear(40,256)網路，和ouput的時候要經過nn.Linear(256,28)網路。

- Compute Loss

  Loss = Reconstruction loss + KL loss

  - Reconstruction Loss

    為和input字元相似度的loss，和第一個test將一個時態轉變成另一個時態有關。在這裏使用nn.CrossEntropyLoss()。

  - KL Loss

    為和 latent 分佈相似度的 loss，和第二個 test 將隨意從 N(0,I)的 sample 生成四種時態有關。因為分佈分別為 diagonal multivariate normal distribution 和 standard normal distribution 是特殊案例，在這裏的計算方法為

$$D_{\mathrm{KL}}\left(\mathcal{N}\left((\mu_1,\ldots,\mu_k)^\mathsf{T},\mathrm{diag}(\sigma_1^2,\ldots,\sigma_k^2)\right)\parallel\mathcal{N}\left(\mathbf{0},\mathbf{I}\right)\right)=\frac{1}{2}\sum_{i=1}^{k}(\sigma_i^2+\mu_i^2-\ln(\sigma_i^2)-1).$$

- RNN Type

  - nn.Embedding

    encoder input、decoder input和condition分別是三個不同的nn.Embedding。而Embedding layer的作用主要是學習單字的distribution representation，能將原本的one-hot編碼降維。

  - nn.GRU

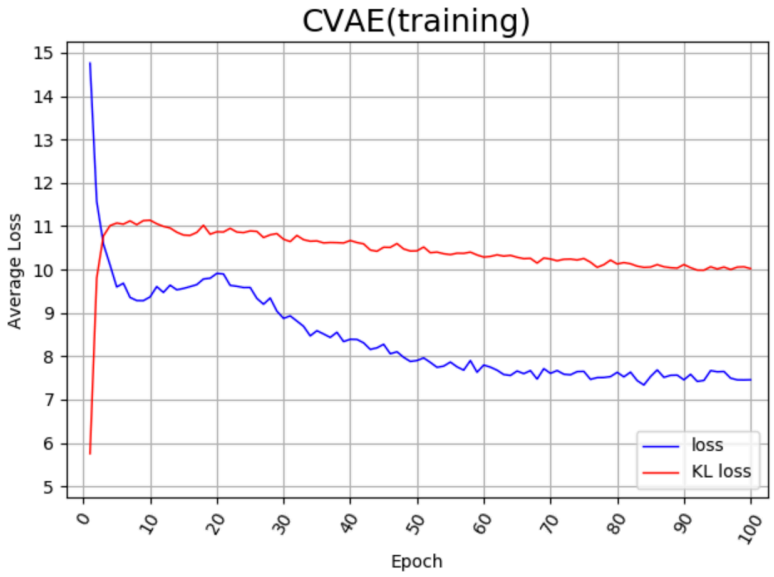
GRU通過gate的機制控制RNN中的資訊流動，用來緩解梯度消失問題；其核心思想是有選擇性的處理輸入，只保留相關資訊進行預測。

B. Specify the hyperparameters

- Epoch= 100
- Condition Embedding Size= 8
- Character Kinds= 28
- Network Hidden Size= 256
- Network Latent Size= 32
- Learning Rate= 0.001 (每epoch乘以0.95)
- Teacher Forcing Ratio= 1.0 (每epoch減0.025直到其值為0.5)
- KL Weight= 0.0 (每iteration加0.00025直到其值為0.5)
- Reconstraction Loss Function= nn.CrossEntropyLoss()

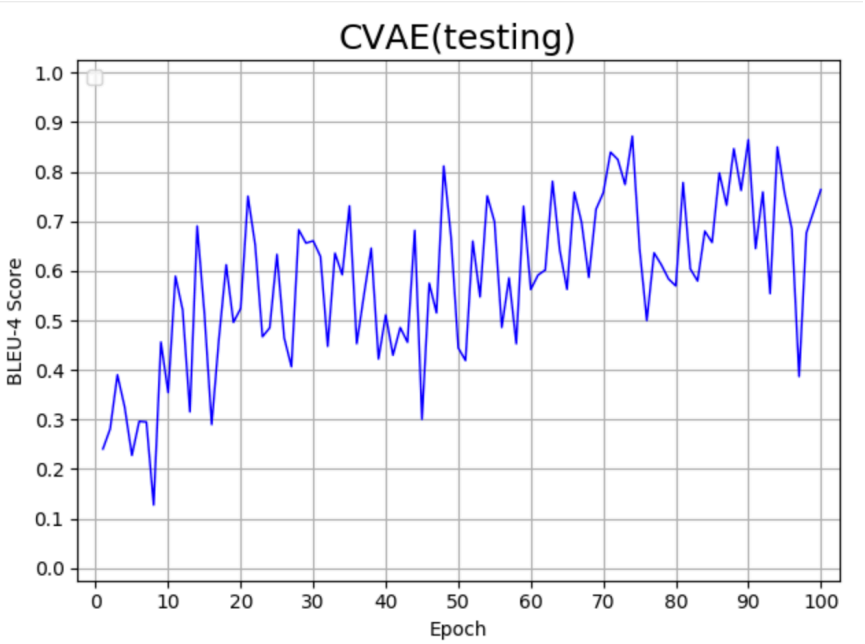
3. Results and discussion

A. Plot the loss and KL loss curve during training and discuss the results according to your setting of teacher forcing ratio, KL weight, and learning rate.



由於KL的weight一直都沒有設很高，所以神經網路較傾向學習reconstuction的loss，也因此KL的loss偏高。

B. Plot the BLEU-4 score of your testing data while training and discuss the result.



BLEU-4 score的值雖然非常震盪，但整體來看是有在穩定成長，直到到差不多0.8左右便不再往上。