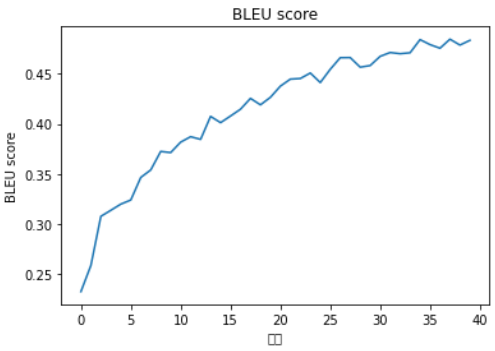
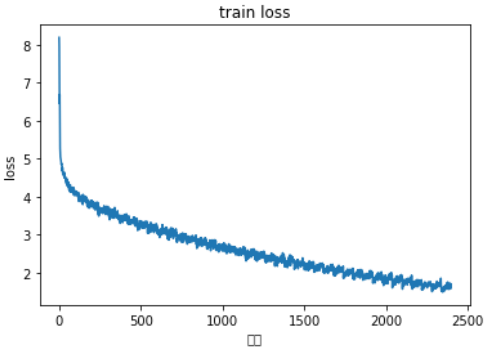
學號：B07701209　系級：電機二　　姓名：王式珩

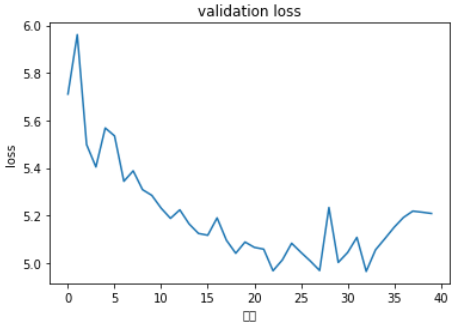
1. (20%) Teacher Forcing:
   a. 請嘗試移除 Teacher Forcing，並分析結果。

With Teacher Forcing:

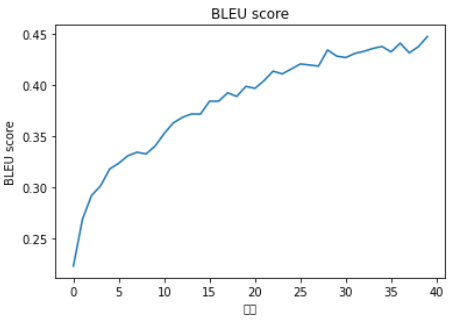
```
val [12000] loss: 5.208, Perplexity: 182.738, blue score: 0.483
test loss: 8.247193674397215, bleu_score: 0.004369565209193104
```

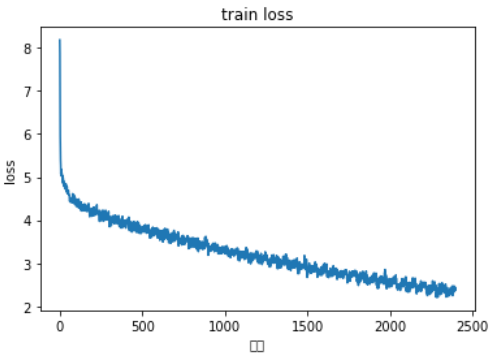


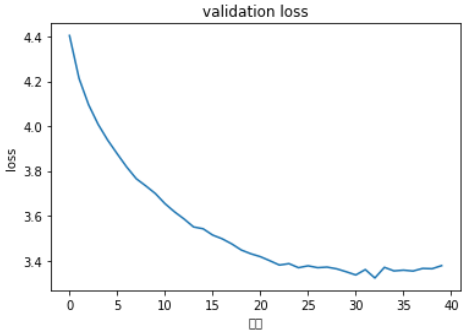





```
Without Teacher Forcing:
val [12000] loss: 3.377, Perplexity: 29.289, blue score: 0.447
test loss: 8.246146486091325, bleu_score: 0.003859525305203452
```







分析結論:

With Teacher Forcing 表現更好(與 Without Teacher Forcing)相比(從 BELU score 看出)，此外兩者的 train loss 在後期都開始劇烈震盪。而 val loss 的部分，With Teacher Forcing 後期有較明顯的震盪，且 loss 值也比較高，Without Teacher Forcing 就較平緩，且 loss 值也比較低。

2. (30%) Attention Mechanism:
   a. 請詳細說明實做 attention mechanism 的計算方式，並分析結果。
      實作方式: 主要參考 https://medium.com/intel-student-ambassadors/implementing-attention-models-in-pytorch-f947034b3e66
      基本上就是把 encoder output、hidden state 在進去 decoder 前再過一層 attention layer，讓 decoder 可以比較知道那些字是比較會影響 Output 的字，如同人類在做翻譯時一樣，會看到特定的字去做當下的翻譯，在這邊有異曲同工之妙。
      那我 attention layer 的實作是參考上方網址，使用 linear+softmax，再經過一些處理，最後再過一次 relu(回到 0~1)，接下來再餵進 decoder。

      沒做 Attention:

```
val [12000] loss: 5.208, Perplexity: 182.738, blue score: 0.483
test loss: 8.247193674397215, bleu_score: 0.004369565209193104
```

      有做 Attention:

```
val [12000] loss: 5.223, Perplexity: 185.483, blue score: 0.475
test loss: 8.243712547516425, bleu_score: 0.004004300513942988
```

3. (30%) Beam Search:
   a. 請詳細說明實做 beam search 的方法及參數設定，並分析結果。
      我這次只做了 topk = 2 的 Beam Search，大致的想法是把過了 decoder 的 output 取 topk(K=2)，然後像程式碼原來做的一樣，把路徑存下來。因為每次吃出發點都是 2 個 node，所以最後會各取一個 topk，一共是 4 個 node，再取最大的兩個，並把其母節點的路徑存下來，以便還原。
      結果分析:

      原本做出來發現 BELU score 沒有變太好的趨勢，所以以為自己做錯，但後來看臉書社團發現，因為 Beam Search 只是找 local minimum，再加上我只找兩個，因此很有可能並沒有效能上的提升，因此之後我會想實作 topk 值較大的模型，希望可以把表現提高。

      沒做 Beam Search:

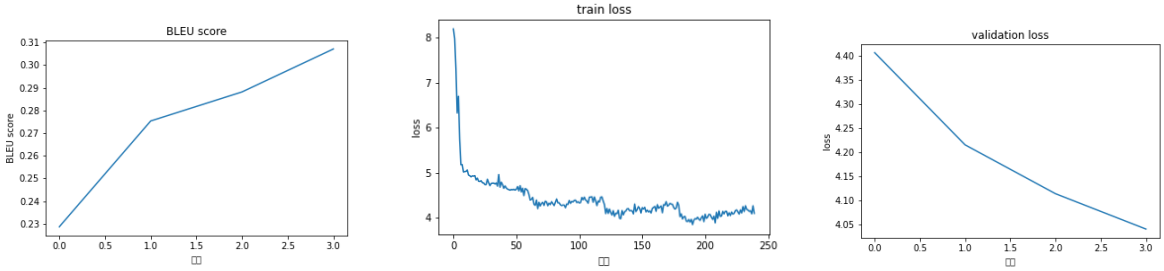
```
val [12000] loss: 5.208, Perplexity: 182.738, blue score: 0.483
test loss: 8.247193674397215, bleu_score: 0.004369565209193104
```

      有做 Beam Search:

```
val [12000] loss: 5.301, Perplexity: 200.505, blue score: 0.488
test loss: 8.241760579154054, bleu_score: 0.0017299276334910105
```

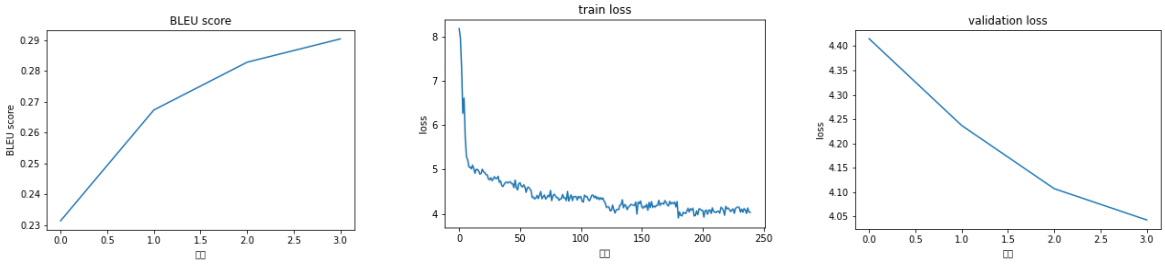
4. (20%) Schedule Sampling:
   a. 請至少實做 3 種 schedule sampling 的函數，並分析結果。

schedule sampling function :cosin
val [1200] loss: 4.040, Perplexity: 56.848, blue score: 0.307



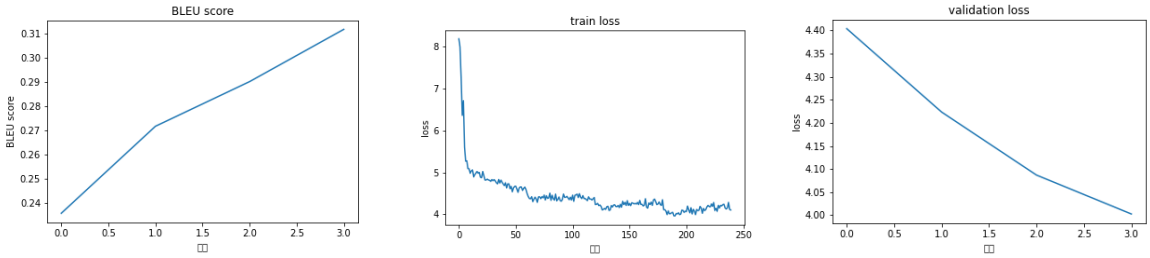
schedule sampling function :exponential

val [1200] loss: 4.078, Perplexity: 59.032, blue score: 0.297



schedule sampling function :linear

val [1200] loss: 4.003, Perplexity: 54.751, blue score: 0.312



結論: 我覺得基本上三種 schedule sampling function 都沒有太出乎意料的表現，彼此間差異也不大，或許原因是我 schedule sampling function 挑的不好，所以我最好的 model 是直接用 teacher forcing 的辦法去做。