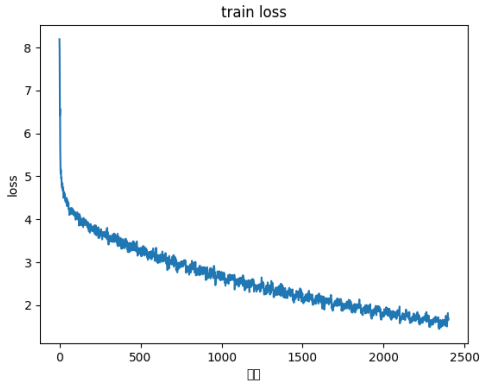
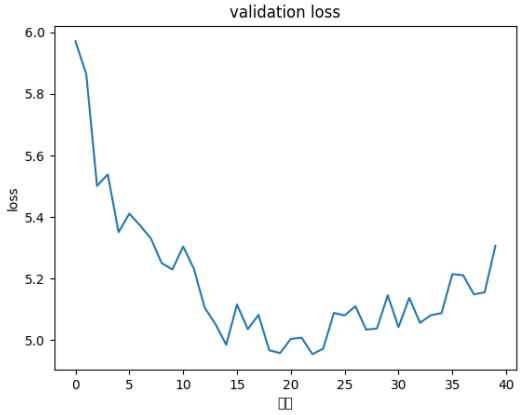
學號：B05901168　　系級：電機四　　姓名：陳冠豪

以下回答對照組為助教所提供之原始 model (teacher Forcing, no attention, no beam search)，該模型的訓練結果如下:

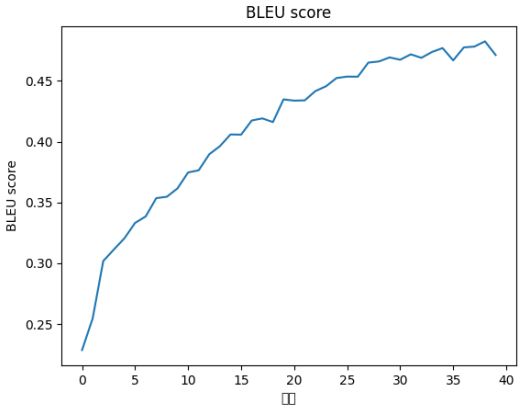
Training loss: 最後結果約可以將 training loss 降到 1 左右



Validation loss : 最後的 testing loss 大約為 5 左右，可以看到跟 training loss 差蠻多的，也有 overfitting 的問題



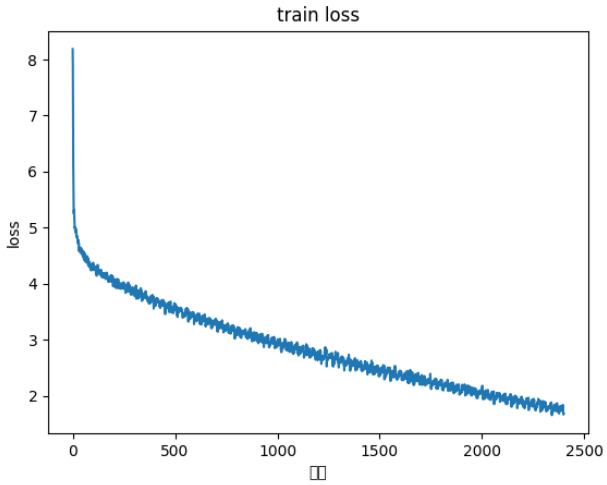
BLEU score: 最後的結果約在 0.45 左右



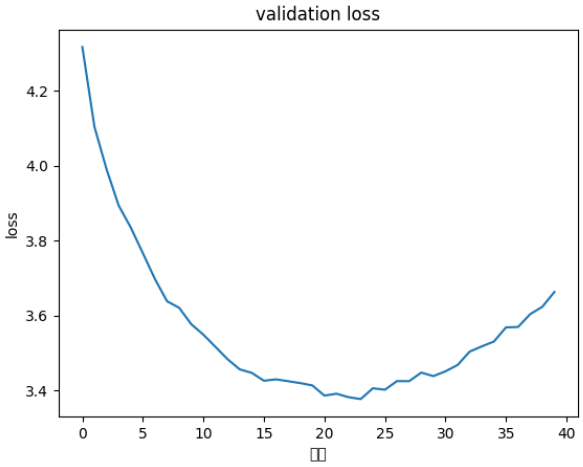
1. (20%) Teacher Forcing:
   a. 請嘗試移除 Teacher Forcing，並分析結果。

將 Teacher Forcing 拿掉之後結果如下:

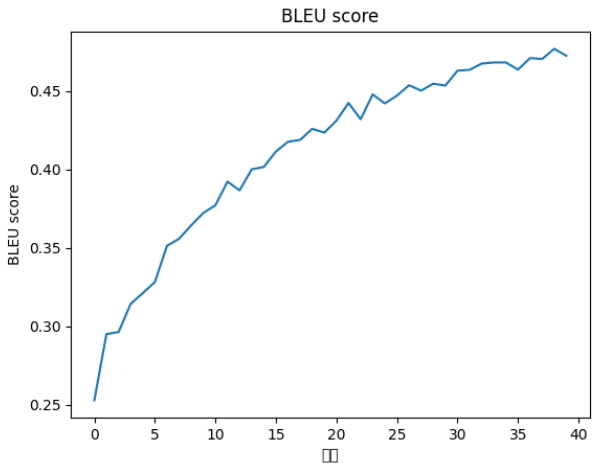
Training loss: 最後結果約可以將 training loss 降到 1 左右，其曲線與對照組相差不大



Validation loss: 最後的 testing loss 大約為 3.6 左右，最低約為 3.4,跟 training loss 還是差蠻多的，也有 overfitting 的問題，不過跟對照組相比算是有進步



BLEU score：最後的結果約在 0.45 左右，跟對照組相差不大

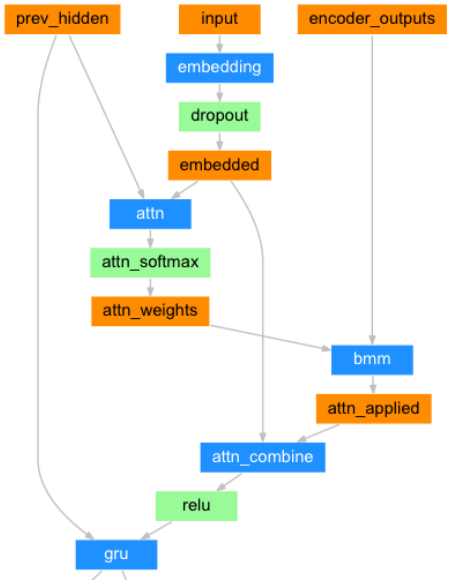


總結: 可以看到移除 teacher forcing 對於整體的影響其實不大，最後的 BLUE score 差不多，但是 validation loss 有降低一些.

2. (30%) Attention Mechanism:

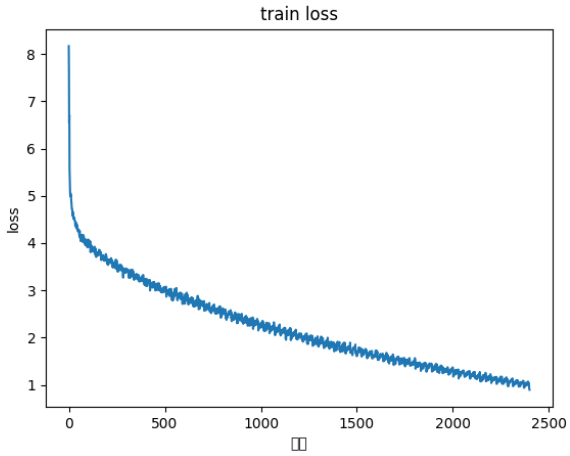
a. 請詳細說明實做 attention mechanism 的計算方式，並分析結果。

Attention 架構參考圖如下，首先將 input 做 embedding 後,與 encoder 最後一層 hidden layer 做 nn.Linear 轉換後再經過 softmax 得到 attention weights，在與 encoder_outputs 做 bmm 得到最後的 attention 並將其串接在 input embedded 後.

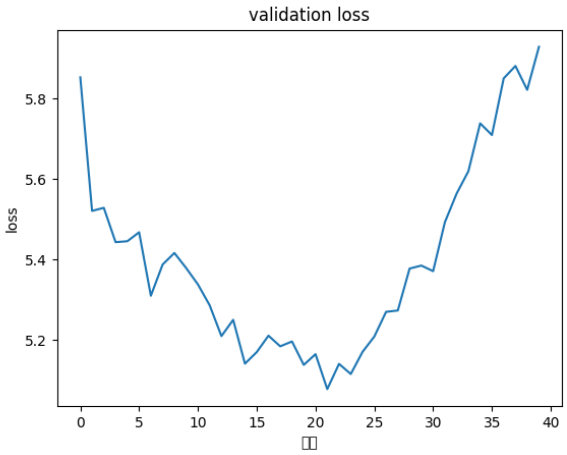


以下為 training 結果

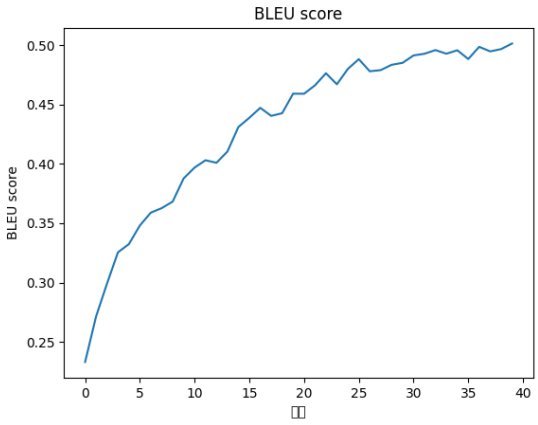
Training loss: 最後結果約可以將 training loss 降到小於 1, 與對照組相比可以得到更小的 loss.



Validation loss: 最後的 testing loss 大約為 5.0 左右, 跟對照組相比 overfitting 的情況非常嚴重



BLEU score : 最後的結果可以接近 50%左右, 跟對照組相比上升約 5%

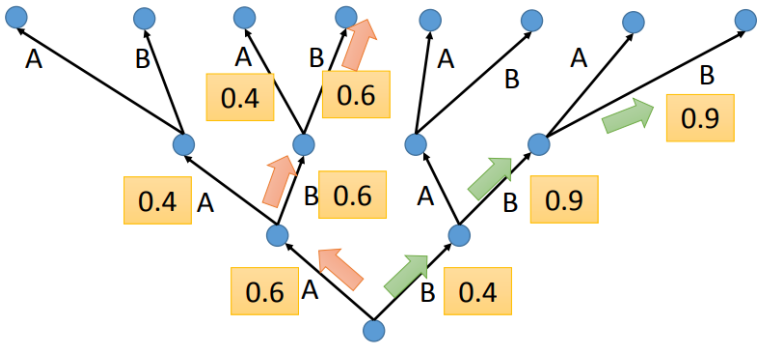


結論: 使用 attention 能夠讓 model 有夠大的能力，讓 training loss 以及 BLUE 得到很好的結果，但是同時 overfitting 的問題較嚴重，可能要增加 dropout 或是調整 model 的架構來避免 overfitting.

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。

Beam search 示意圖如下，首先將 input 餵進 decoder 得到各個詞的分數，接著要將這個分數經過 log_softmax 轉換成機率，每次取機率最大的前 k 個 (根據 beamsize)，接著依分數去 sort (分數計算為取算術平均，由於有經過 log_softmax，所以取算術平均會與各個機率相乘成正相關) 最後取分數最大的當作答案



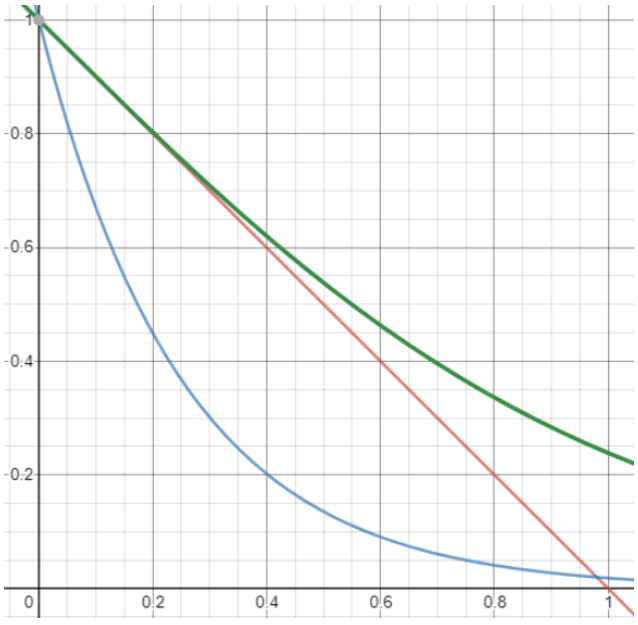
以下結果分別為在對照組 model 以及加入 attention model 拿去做 validation 的 BLEU 結果

	對照組 model	Attention model
K = 1	0.422	0.495
K = 5	0.424	0.505
K = 10	0.418	0.510
K = 20	0.410	0.489

在對照組的結果，加入 beam search 對 BLEU 的正確率影響不大，而對 attention 的結果有較大的提升，我想是因為對照組的正確率本來就較低，所以即使 beam search 開很大，還是有可能答案不在 beam search 的範圍內. 另外還有一個現象是當 beam size 開很大的時候反而會讓 BLEU 分數下降，這個現象應該是因為其實在做 beam search 的時候目標不是去 maximize BLEU，所以 BLEU 的分數會浮動應該是正常的.

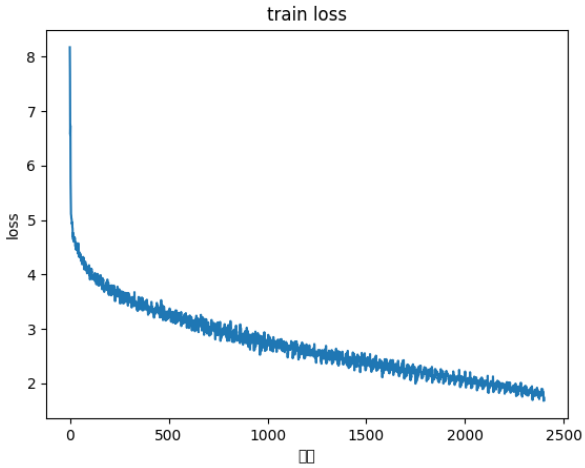
4. (20%) Schedule Sampling:
   a. 請至少實做 3 種 schedule sampling 的函數，並分析結果。
   總共使用了三種方法，分別為
   1 : y = 1 - x
   2 : y = 1- tanh(x)
   3 : y = exp(-4x)

   函數圖形如下: 1 為紅色, 2 為綠色, 3 為藍色, x 軸為對 total step normalize 後的結果(x = current step / total step)

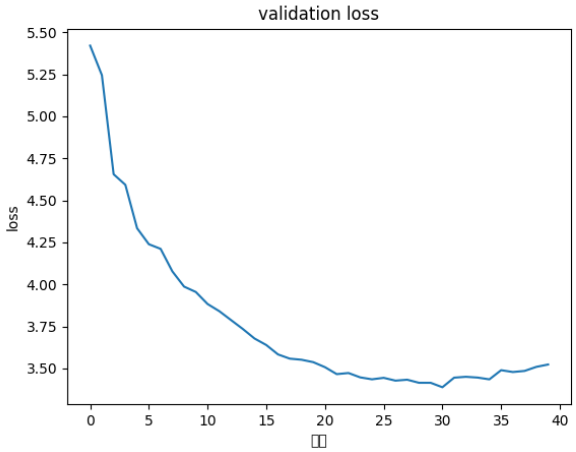


1 結果 (函數為 y = 1 - x)

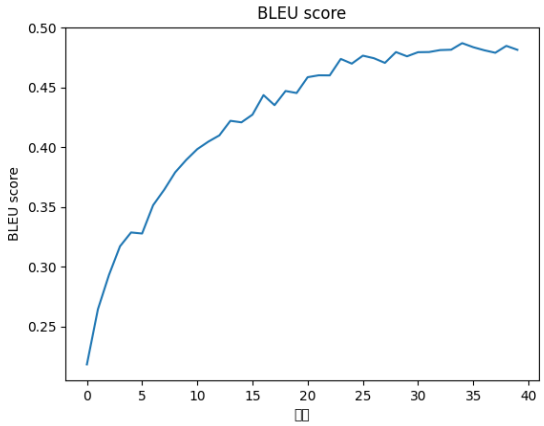
Training loss: 最後結果約可以將 training loss 降到 1 左右，其曲線與對照組相差不大



Validation loss: 最後的 testing loss 大約為 3.6 左右，最低約為 3.4,跟 training loss 還是差蠻多的, overfitting 的問題相較於對照組較不嚴重

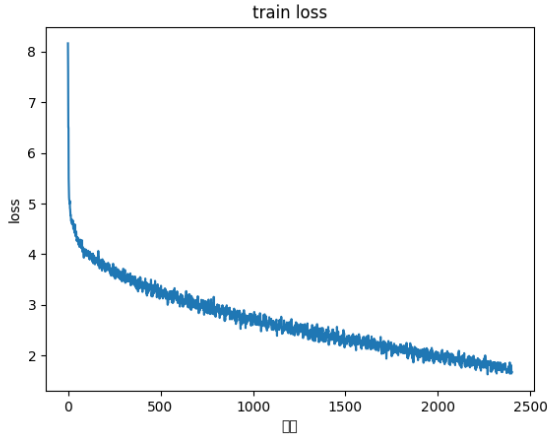


BLEU score： 最後的結果約在 0.48 左右，相較於對照組上升 3%

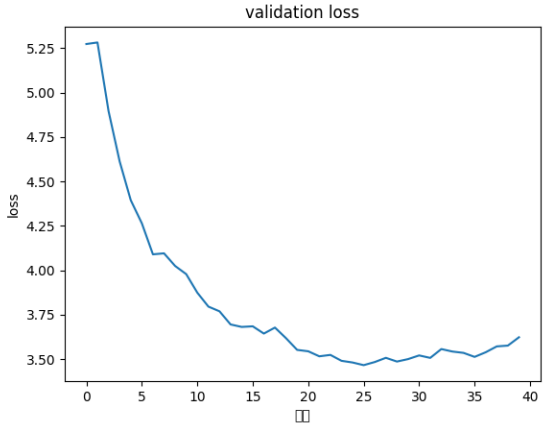


2 結果 (函數為 y = 1- tanh(x))

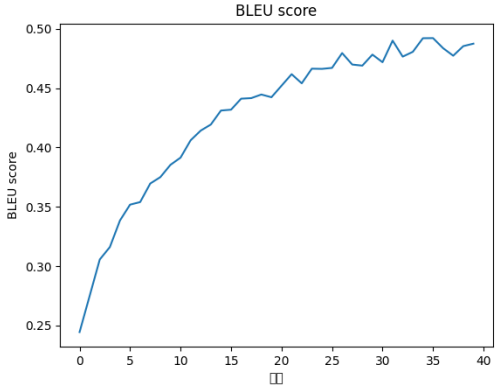
Training loss: 最後結果約可以將 training loss 降到 1 左右，其曲線與對照組相差不大



Validation loss: 最後的 testing loss 大約為 3.5 左右，跟 training loss 還是差蠻多的, overfitting 的問題相較於對照組較不嚴重，而相較於 linear 函數,其收斂的速度較快

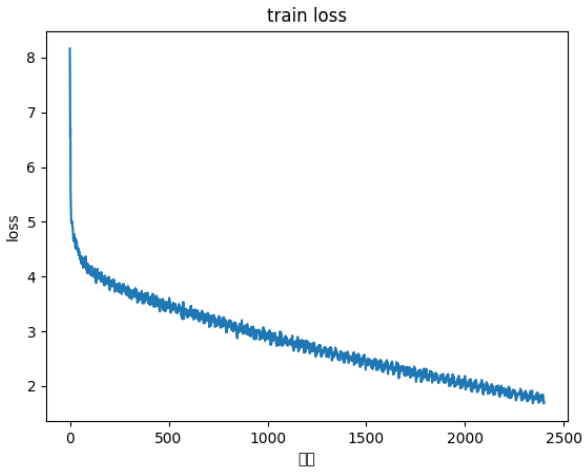


BLEU score： 最後的結果約在 0.48 左右，相較於對照組上升 3%，結果與使用 linear 函數差不多

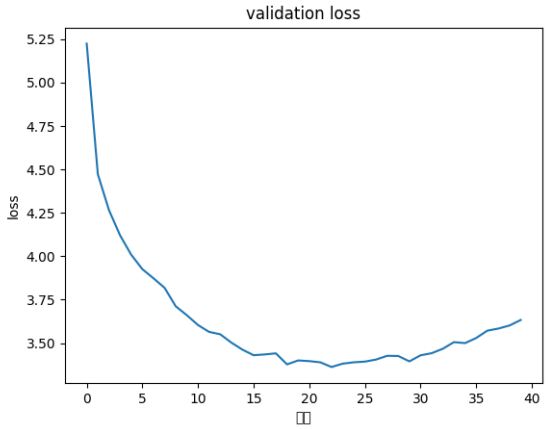


3 結果 (函數為 y = exp(-4x))

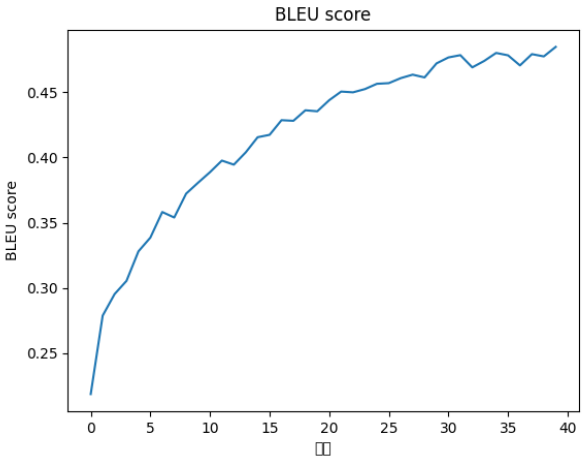
Training loss: 最後結果約可以將 training loss 降到 1 左右, 其曲線與對照組相差不大



Validation loss: 最後的 testing loss 大約為 3.6 左右, 最低約為 3.4,跟 training loss 還是差蠻多的, overfitting 的問題相較於對照組較不嚴重, 但跟前兩個函數相比,其 overfitting 的情況較嚴重



BLEU score：最後的結果約在 0.46 左右，相較於對照組上升 1%，但是跟前兩者相比效果較不好



結論：使用這三者都能使 BLEU 的結果上升，其中使用 linear 與 tanh 函數效果較顯著，並能減少 overfitting.