學號：B06902006 系級：資工三 姓名：王俊翔

1. Teacher Forcing:
   請嘗試移除 Teacher Forcing，並分析結果。
   Ans : (beam search = 5, dot product, step = 18000)
   Evaluation

Ways	Max Bleu
Teacher Forcing	0.549
Without Teacher Forcing	0.615

   從結果可以知道，不使用 teacher forcing 效果會差很多，僅僅使用結果做預測下個字會導致正解並沒有實際被 train 到，不過並沒有直接壞掉我猜原因是我 step 調到很高，讓他最後還是有學到一點點正解。
2. Attention Mechanism:
   請詳細說明實做 attention mechanism 的計算方式，並分析結果。
   Ans :
   這裡我做了兩種不同的 attention 方式，分別為過一層 neural network 以及 dot product，以下為兩種的詳細說明。
   a. 過 NN
      實作方法：這裡是接在 decoder input 的地方，所以取其 hidden[0] 出來與 encoder output 做運算。這裡我把每個 word 的 output 拆開，每個都接上 hidden[0]在把它全部接起來，接著使用 nn.Conv1d 每個 output 加 hidden[0]的長度為其 filter 大小及 stride 的值，如此就等同於使用同個參數去對每個 word 的 encoder output 過 NN。然後過 softmax，最後再與 encoder output 做矩陣相乘 torch.bmm 即可完成。
      Evaluation :

Ways	Max Bleu
Baseline	0.493
NN	0.483

      之所以會出現這樣的問題，我認為是因為接在前面的關係導致 Bleu Score 沒有變得比較好，因此之後的方法全部改成接在後面，且由於這樣的架構導致 network 變得比較大，所以 train 的時間太久，所以後來都是使用 dot product 在跑。
   b. Dot product (接在 fully connected layer 前面)
      實作方法：這裡的實作就相對簡單，把 decoder output 最後一層拿出來與 encoder output 做 torch.bmm 後可以得到 batchsize * sen_len，之後看要不要過 softmax 然後再與 encoder output 做一次 torch.bmm 即可。

Evaluation :

Ways	Max Bleu
Baseline	0.493
Dot Product with softmax	0.539
Dot Product without softmax	0.605
Dot Product without softmax (dropout 0.5 on fully connect)	0.614

由上面可以知道接再後面的 attention 會使效果變好，其中把 softmax 移掉後會使 Bleu Score 變得再更高且成長幅度極大，加 dropout 後也有不錯的提升。

3. Beam Search:

請詳細說明實做 beam search 的方法及參數設定，並分析結果。

Ans :

這裡定義了一個 class 準備一些 beam search 要用到的參數

```
class beam_node():
    def __init__(self, parent, hidden, input, prob, length, output, pred):
        self.parent = parent
        self.hidden = hidden
        self.input = input
        self.prob = prob
        self.length = length
        self.output = output
        self.pred = pred
```

Parent : 為了做 back propagation。

Hidden, input : 丟到 decoder

Prob : 目前這條路的機率

Length : 目前長度，達 max length 即停止

Output : 紀錄當前預測結果 output (pred 最後沒用到= =)

之後使用 bfs 的方法去做前 n 名的紀錄，主要是使用一個 queue，queue 中存的就是一個個 beam node，直到 queue 中沒有元素，每一輪都取前 n 高 prob 的丟進 queue (這裡取 log 相加值比較大)，所以第一輪有 n 個元素丟進 queue，再來都會有 n*n 個取 n 個丟入，最後再做 back propagation 得到所有的結果就完成 beam search 了。

Evaluation

Ways	Max Bleu
Baseline	0.614
Beam size = 5	0.615
Beam size = 10	0.613

由上表可知其實有沒有做 beam search 對 bleu score 沒有太多的影響，主要影響是收斂的速度，beam size 越大越早可以收斂到約 0.6 左右，加上 bleu score 的計算方式再某些情況沒有那麼的準確，或許使用不同的 evaluation 就會有不同的結果。

4. Schedule Sampling:
   請至少實做 3 種 schedule sampling 的函數，並分析結果。
   Ans : (beam search = 5, dot product, step = 18000)
   這裡使用了三種函數，分別是 linear, exponential decay 和 inverse sigmoid decay。
   Linear : 1 – (current step / total step)
   Exp : exp(-1*current step / (total step / 4))
   Inverse Sigmoid : 1100 / (1100 + exp(current step / 1100))

   Evaluation

Ways	Max Bleu
Teacher Forcing	0.615
Linear	0.585
Exponential	0.566
Inverse Sigmoid	0.569

   從這裡的結果發現 Max Bleu 反而下降了，其實實施 Schedule Sample 後收斂地比原本快，不過我想是因為我讓他機率太早就變成使用原 output 了，所以效果並沒有那麼好，Linear 的下降的程度沒有其他兩者來的多，或許多調整幾個參數後會有不同的結果。