1. **Please answer the following questions about GRU structure (30%)**

   (1) What are its strength and weakness compared to LSTM?

   GRU:

   僅擁有兩個 Gate(update gate, reset gate)，且沒有 memory cell

   Strength:

   一、計算效率快

   二、訓練速度快

   三、許多時候表現與 LSTM 相當

   Weakness:

   一、參數少

   二、長期記憶能力低

   LSTM:

   擁有三個 Gate(input gate, forget gate, output gate)，且有一個 memory cell

   Strength:

   一、參數多

   二、長期記憶能力高

   Weakness:

   一、計算量大

   二、訓練速度慢

   三、比較容易 overfitting

   (2) Can we say GRU is an improvement over LSTM? Give your detailed reasoning

   我認為 GRU 以及 LSTM 不能拿來比較優劣，主要是他們有不同的應用層面，若在有限的 data、算力上，GRU 在綜合考慮時間、硬體成本及結果上可能表現會比 LSTM 佳，然而在需要長期記憶的任務上，LSTM 也無法被 GRU 所替代，因此我認為我們不能夠說" GRU is an improvement over LSTM "。

2. **How are recurrent neural networks different from other deep learning networks?**

RNN 與其他 network 最大的不同就是 RNN 具有記憶的能力，他可以保留上一個步驟的數據，並依照過去的數據計算預測現在的結果，讓其擁有記憶並不被新數據覆蓋的能力，這是一般的神經網路無法達成的。也基於他是一步步根據過去結果計算當前的數據，他比其他的神經網路更適合處理序列的問題，包括語言、語音、等有時間順序的任務。不過一般的 RNN 很容易出現梯度消失很難訓練的問題，因此後續出現 LSTM、GRU 等會與過去的資料直接做計算，讓其出現 shortcut path 使梯度消失或爆炸的問題獲得解決的模型。

3. **What are the limitations of recurrent neural networks?**

如同上題所述，RNN 的輸出會考慮過去所有的輸出數據做計算，也因此輸出必須一個接著一個，無法一次計算全部數據，導致計算效率低下，也無法做平行化計算，這是 RNN 最大的弱點，也因此未來才有出現 self-attention 等可以平行化運算的 network。同樣原因，也會使 RNN 在訓練上耗時更久，且較難保留久遠的訊息，在處理序列中距離較遠的數據時較果不佳。另外，一般的 RNN 也很容易出現梯度消失或爆炸問題，因此後續才出現 LSTM、GRU 等可以解決梯度消失/爆炸的模型，有些有可以解決長期記憶衰減的問題。

4. **Please introduce a subtask of NLP**

subtask of NLP: **Text Summarization**

(1) What is its goal?

從輸入之較長的文本輸出簡短的重點整理，並保留文本原意及想法，可以是從原文擷取文句的方式，也可以從頭自己生成新的文章。

(2) What common dataset does it use?

1. DUC (Document Understanding Conferences): NIST 提供的標準數據庫。
2. CNN/Daily Mail: 由 CNN、Daily Mail 收集之新聞數據及摘要。
3. Gigaword: 由 Linguistic Data Consortium（LDC）提供，包含來自美聯社、紐約時報、華盛頓郵報等新聞文章。

(3) How to calculate its metric?

常見指標: ROUGE（Recall-Oriented Understudy for Gisting Evaluation）

又主要分成 ROUGE-N、ROUGE-L、ROUGE-S

1. **ROUGE-N**：計算 n-gram(單詞、詞組等) recall 的數量比例。

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \quad (1)$$

Example:

N = 1

Reference：（人工標注）

R1: police killed the gunman.

R2: the gunman was shot down by police.

自動摘要：（程式生成）

C1: police ended the gunman.

C2: the gunman murdered police.

$$\boldsymbol{ROUGE-1(C1) = \frac{3+3}{4+7} = \frac{6}{11}}$$

分子：C1 對應 R1 有 3 字重複、C1 對應 R2 有 3 字重複

分母：R1、R2 的字數相加

$$\boldsymbol{ROUGE-1(C2)=\frac{3+3}{4+7}=\frac{6}{11}}$$

分子：C2 對應 R1 有 3 字重複、C2 對應 R2 有 3 字重複

分母：R1、R2 的字數相加

N = 2

Reference：（人工標注）

R1: police killed the gunman.

R2: the gunman was shot down by police.

自動摘要：（程式生成）

C1: police ended the gunman.

C2: the gunman murdered police.

$$\boldsymbol{ROUGE-2(C1)=\frac{1+1}{3+6}=\frac{2}{9}}$$

分子：C1 對應 R1 有 1 組 2 字重複、C1 對應 R2 有 1 組 2 字重複

分母：R1、R2 的 2 字詞組數相加

$$\boldsymbol{ROUGE-2(C2)=\frac{1+1}{3+6}=\frac{2}{9}}$$

分子：C2 對應 R1 有 1 組 2 字重複、C2 對應 R2 有 1 組 2 字重複

分母：R1、R2 的 2 字詞組數相加

2. **ROUGE-L**：LCS (Longest Common Subsequence)，利用最大共同子序列來評估生成的摘要與參考摘要之間的匹配度。

Example:

References (人工標註):

R1: police killed the gunman

Summary (程式生成):

S1: police kill the gunman.

S2: the gunman kill police.

$$ROUGE-L(S1) = \frac{3}{4} = 0.75(police\ the\ gunman)$$

$$ROUGE-L(S2) = \frac{2}{4} = 0.5\ (the\ gunman)$$

原論文公式如下:

X → Reference（人工給的摘要），長度為 m。

Y → 自動摘要（程式生成），長度為 n。

β：控制 P 和 R 的相對重要性。

$$R_{lcs} = \frac{LCS(X,Y)}{m} \quad (2)$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \quad (3)$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \quad (4)$$

公式看似很複雜，但實際上主要考慮的只是 R_{lcs}。

這種算法的優點是可以用單字順序篩選出最正確的答案，若同樣這個例子我們使用上一個算法: ROUGE-2，得到結果為:

$$ROUGE-2(S1) = \frac{1}{4} = 0.25\ (the\ gunman)$$

$$ROUGE-2(S2) = \frac{1}{4} = 0.25\ (the\ gunman)$$

兩者明明語義完全相反，但使用 ROUGE-2 卻得到同樣的分數，這就是原先 ROUGE-N 的劣勢。

但同樣的，若我們將輸出結果的動詞改變，如下:

S1: police save the gunman.

S2: the gunman was killed by police.

也會使算法有錯誤的判斷。或是我們有一個新的輸出為

S3: the gunman policed killed

S3 明明是最符合原語句的結果，但使用 ROUGE-L 仍只能得到與 S2 相同的分數，因為最長匹配的單字序列僅有 2，這些就是 ROUGE-L 的劣勢。

3. **ROUGE-S**：測量具有間距的 skip-gram 序列的匹配度，也就是詞組可以不用連續出現。

Example:

References (人工標註):

R1: police killed the gunman

Summary (程式生成):

S1: police save the gunman.

S2: the gunman was killed by police.

S3: the gunman policed killed

S1 和 R1 有 3 個 skip-bigram: police the, police gunman, the gunman

S2 和 R1 有 1 個 skip-bigram: the gunman

S3 和 R1 有 2 個 skip-bigram: police killed, the gunman

原論文公式如下:

X → Reference（人工給的摘要），長度為 m。

Y → 自動摘要（程式生成），長度為 n。

SKIP2(X, Y): skip-bigrams 的數量。

β：控制 P 和 R 的相對重要性。

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)} \qquad (16)$$

$$P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)} \qquad (17)$$

$$F_{skip2} = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2}+\beta^2 P_{skip2}} \qquad (18)$$

令 β = 1

$$ROUGE - S(S1) = \frac{3}{C(4,2)} = 0.5$$

$$ROUGE - S(S2) = \frac{1}{C(4,2)} = 0.167$$

$$ROUGE - S(S3) = \frac{2}{C(4,2)} = 0.3$$

以 ROUGE-S 計算的結果優至劣排列為:

$$S1 > S3 > S2$$

結論比較三種計算方式，大致來說 ROUGE-S 比另外兩種計算方式好，然而對於特定任務如極短的摘要來說，ROUGE-1 及 ROUGE-L 表現優異。

(4) What are its practical applications in real-life?

當想要快速了結一篇長篇文章的重點時，現在人就常使用大型語言模型幫忙完成 Text Summarization 的任務，包含學術文章、新聞報導、公司財報、著作文章等，可以大幅降低使用者的時間成本。

Reference:

https://en.wikipedia.org/wiki/ROUGE_(metric)

https://mycollegenotebook.medium.com/rouge-%E8%A9%95%E4%BC%B0%E6%96%B9%E6%B3%95-%E8%87%AA%E5%8B%95%E6%96%87%E6%9C%AC%E6%91%98%E8%A6%81-8d9e9516698b