第七周

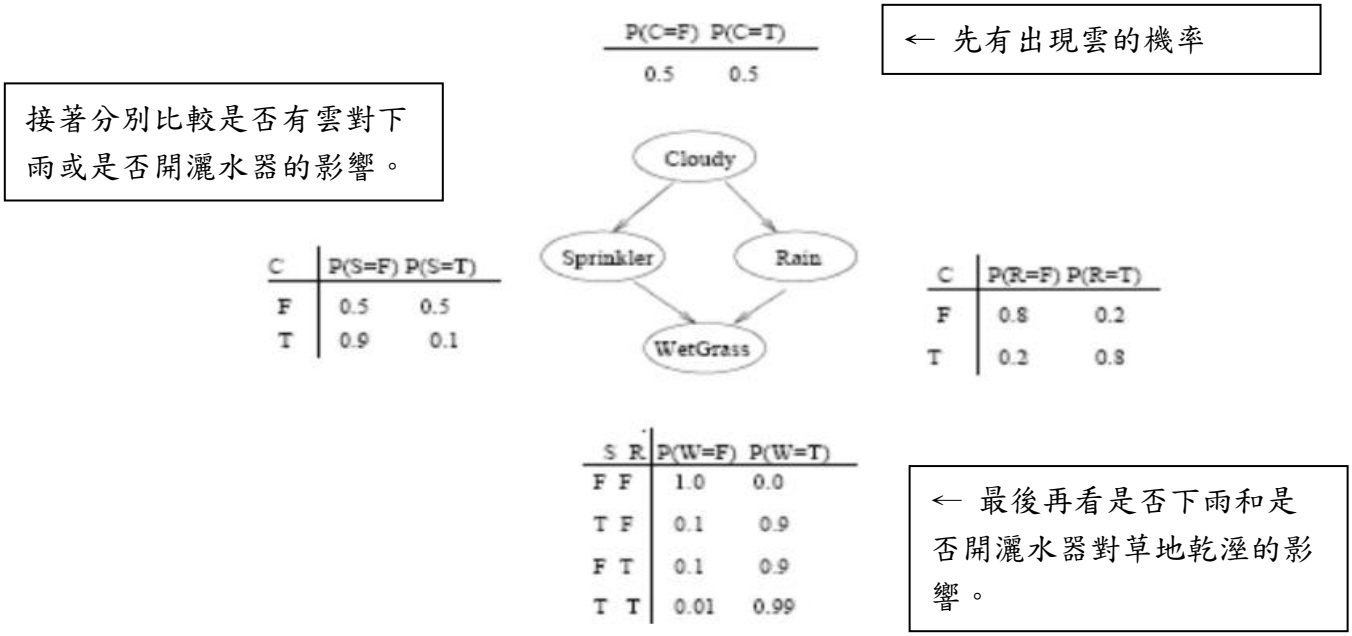
第七周的課程上的是遞迴式神經網路(Recurrent Neural Network,RNN)。

1. 圖機率模型和隱藏馬可夫模型(Hidden Markov Model, HMM)

無論是隱藏的馬可夫模型，還是遞迴式神經網路，都是使用序列資料來學習。序列資料是一組和時間先後有關的資料，例如：股票、語音等。因此，無論是遞迴式神經網路還是隱藏馬可夫模型，目的是為了找出資料間的關係，以及預測未來走向。

①圖機率模型

圖機率模型是一種結合圖和機率的學習方法。如下圖：



藉由上圖我們可以發現，圖機率模型可以很清楚的表達狀態或事物之間的關係，也因此我們可以用圖機率模型找出更複雜的機率關係表(如圖中最下表，有兩個或以上的變因)

一個圖機率模型包含三個部分：

1. Representation

圖機率模型的 Representation，指的是如何去表達隨機變數之間的關係(即是如何畫圖)。不同的畫圖方式可能表達的意思會不一樣，學習效果也會不同，因此，Representation 是很重要的。

2. inference

給定一些已知狀態，估計其他變數。例如：上圖中若給定草地狀態，則往回估計灑水器、下雨和雲的狀態。

3. learning

給定資料，學習變數之間的關係(圖形結構)或參數。

圖機率模型有以下優點：

1. 可彈性加入先驗知識。
2. 可建立變數之間的關係。
3. 獨立性可以減少計算量。

圖機率模型和深度學習的相似之處：

1. 皆用圖形結構表示
2. RBM(Restricted Boltzmann Machines)試圖機率模型和深度學習結合的產物。

②隱藏馬可夫模型(Hidden Markov Model, HMM)

隱藏馬可夫模型本質上即屬於一種序列圖機率模型。但隱藏馬可夫模型含有隱藏狀態，意思是含有無法觀察到的隱藏過程。例如：語音轉文字時，文字即為隱藏過程。

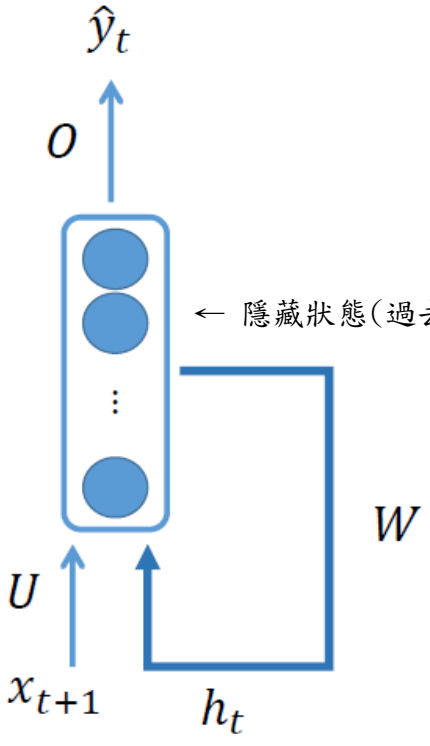
性質：

$P(x_t|x_{t-1})$，意即現在的狀態只受前一個時間的狀態(t-1)影響，和其他時間點(t-2, t-3, …)無關。換言之，現在狀態和歷史路徑是獨立的。常用隱藏馬可夫模型和遞迴式神經網路做對比。

2. 遞迴式神經網路

遞迴式神經網路是利用神經網路來處利序列資料。對比隱藏馬可夫模型，遞迴式神經網路具有記憶性，即 $P(x_t|x_0, x_1, \dots, x_{t-1})$。過去的整段歷史，皆會影響現在。導致雖然可以得到更精確的結果，但卻也會使模型過於複雜。

①遞迴式神經網路模式



← 隱藏狀態(過去的資料，可理解為記憶)，為連續向量。

← 將原本資料乘上一個權重值 W，可理解為篩選過去記憶，使資料不要太多，限縮資料複雜度。

↑新的輸入資料

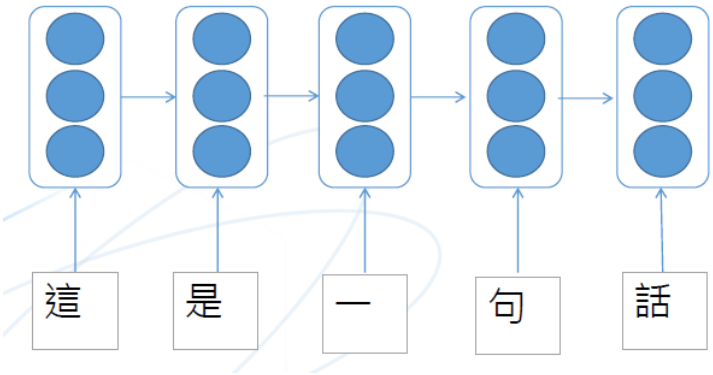
以上的過程可表示成如下算式：

$h_{t+1} = g(Wh_t + Ux_{t+1})$ ，將過去記憶和新資訊放在一起計算。

$\hat{y}_t = Oh_{t+1}$

Δ和 HMM 比較，遞迴式神經網路的隱藏狀態是一個連續多維向量，可以容納更多訊息。

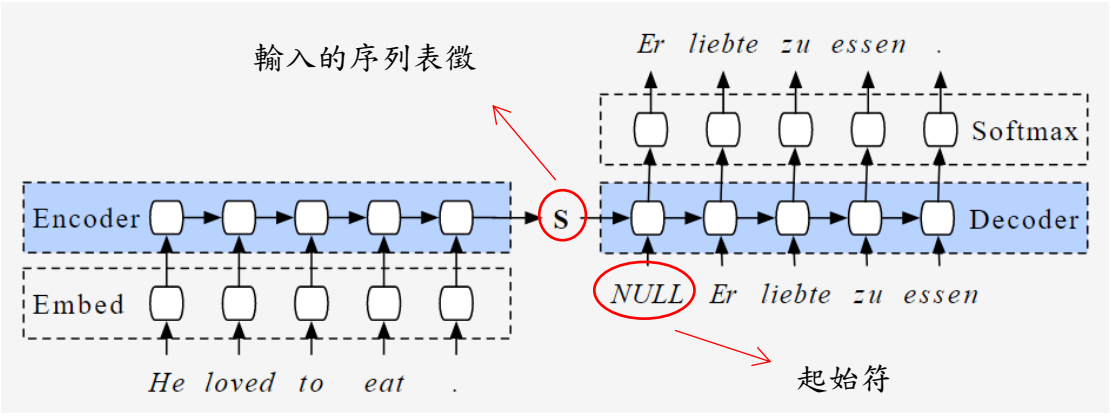
②序列表徵



如例圖，序列表徵即是將序列資料壓縮至隱藏層向量之內。遞迴式神經網路也是利用序列表徵來學習，如以下的 Sequence to sequence(seq2seq)。

1. Sequence to sequence learning

由一個 encoder 和一個 decoder(可能都是 RNN 神經網路，或 CNN 和 RNN 配)組成，常會應用在翻譯的機器學習，如下圖：



Encoder 負責處理和接收輸入的資料，並壓縮成表徵。該表徵會成為 decoder 的初始輸入之後便更新狀態向量。狀態向量經過 Softmax 函數之後，根據機率分布決定第一個字，之後再將第一個字當作第二個神經元的輸入，重複以上過程直到遇到終止符號。

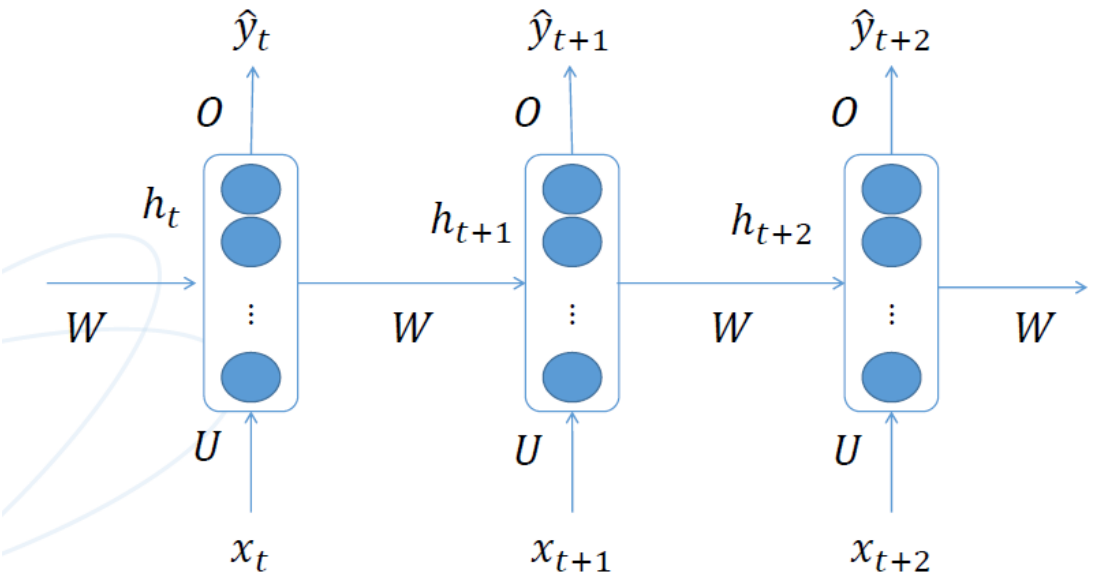
另外，也可以讓 CNN 當作 encoder，輸入影像表徵到 decoder。相關的應用有自動生成圖片標題等。

③遞迴式神經網路的計算過程

在看到遞迴式神經網路的形式後，接下來需要解決的問題是：因為序列的長

度不一，如何以計算圖計算遞迴式神經網路？而參數 W、O、U 重複參與計算過程，如何計算梯度？

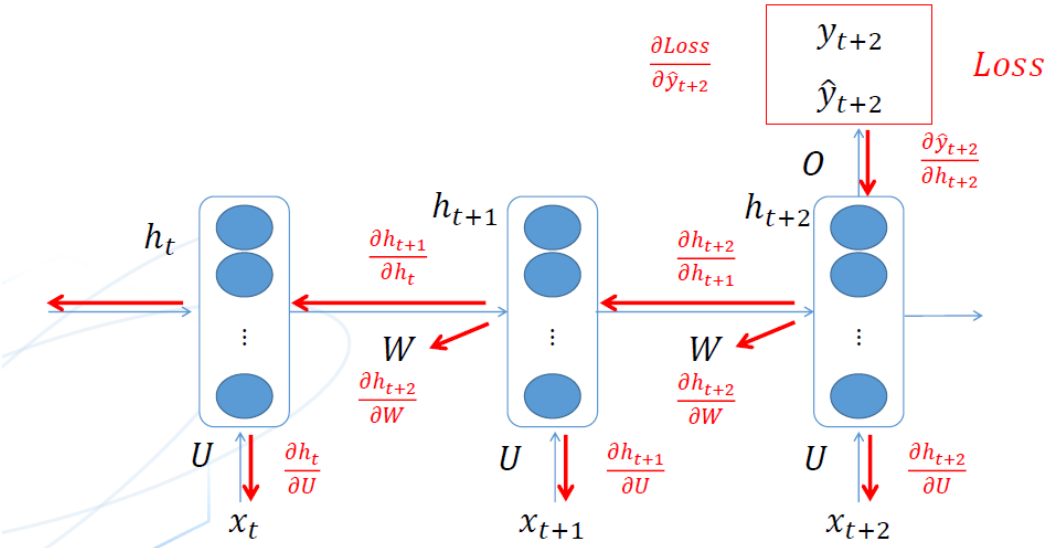
1. 計算圖與正向傳遞



如上圖，因為每次的序列長度不同，所以需要依照序列的長度來調整計算圖的規模，所以每次的計算圖的長度都不一定會相同。

2. 反向傳遞

正向傳遞和神經網路大同小異，主要的問題會出現在反向傳遞，如下圖：



在 RNN 中，每層的同一參數算出的梯度是不同的，舉 t+2 和 t+1 層的參數 U 為例：

$$\frac{\partial \text{Loss}}{\partial U_{t+2}} = \frac{\partial \text{Loss}}{\partial \hat{y}_{t+2}} \frac{\partial \hat{y}_{t+2}}{\partial h_{t+2}} \frac{\partial h_{t+2}}{\partial U_{t+2}}$$

$$\frac{\partial \text{Loss}}{\partial U_{t+1}} = \frac{\partial \text{Loss}}{\partial \hat{y}_{t+2}} \frac{\partial \hat{y}_{t+2}}{\partial h_{t+2}} \frac{\partial h_{t+2}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial U_{t+1}}$$

因為每一個時刻的參數梯度不盡相同，因此，當我們在算損失函數對該參數的總梯度時，需要把每個時刻的梯度相加：

$$\frac{\partial \text{Loss}}{\partial \text{U}} = \sum_{s=0}^{t+2} \frac{\partial Loss}{\partial U_s}$$

以上的梯度計算方式，讓 RNN 在計算梯度時，常會遇到梯度衰減或爆炸問題：

$$\prod_{i=l}^{L} \frac{\partial h_{i+1}}{\partial h_i} = U^n$$

因為每個參數都相同，梯度在向後傳遞時是一路累乘的，因此，當參數出現了糟糕的特性(例如$\text{U} \gg 1$ or $\text{U} < 1$)，則越往深層，梯度值就無法保持和原本接近，導致只有靠近輸出層的位置會被正確改變。如此，原本預設的長期記憶性也難以發揮。

解決梯度衰減或爆炸：短期記憶性模型(Long Short Term Memory)

由於梯度爆炸或消失的問題會導致時間點近的記憶的影響較大，使較久遠的訊息難以被捕捉，一些重要的信息可能也會因此而難以加入學習。為了解決這個問題，短期記憶性模型嘗試去調控每個資料(包括過去和新的)參與學習的比例，讓重要的資料參與多一點，不重要的資料則調低比例甚至遺忘它。這樣的機制有效的解決梯度的問題，增加了學習的精確度。

短期記憶性模型增加了調控門來控制資料的比例，各調控門的公式如下：

※σ_g是用來調控資料比例的 Sigmoid 函式，值域是 0~1。

Input gate： $i_t = \sigma_g(W_i x_i + U_i h_{t-1} + b_i)$

調控輸入資料的比例。

Forget gate： $f_t = \sigma_g(W_f x_i + U_f h_{t-1} + b_f)$

選擇要丟棄什麼資料。

Cell state：$\text{C}_t = f_t \circ C_{t-1} + i_t \circ \sigma_C(W_C x_i + U_C h_{t-1} + b_C)$

更新神經元的狀態。使用經過篩選的過去資料和新加入的資料(同樣被篩選過)放入一個神經元之中。

Output gate： $O_t = \sigma_g(W_o x_i + U_o h_{t-1} + b_o)$

再次過濾資料，決定誰要輸出。

Output： $h_t = O_t \circ tanh(C_t)$

將細胞狀態通過 tanh 函式，輸出-1~1 的值。在和 O_t 相乘，最後輸出數值。

参考資料

1. Introduction to Seq2Seq Models

   https://www.analyticsvidhya.com/blog/2020/08/a-simple-introduction-to-sequence-to-sequence-models/