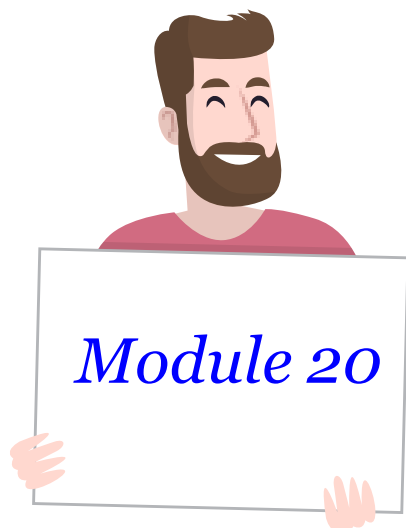




LSTM及GRU



designed by  freepik

Estimated time:
45 min.

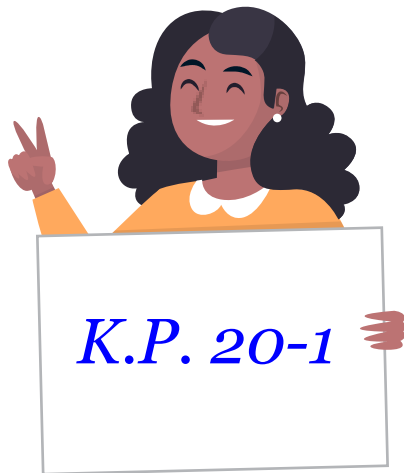
學習目標

- 20-1: RNN Cell
- 20-2: LSTM介紹
- 20-3: GRU介紹



20-1: RNN Cell

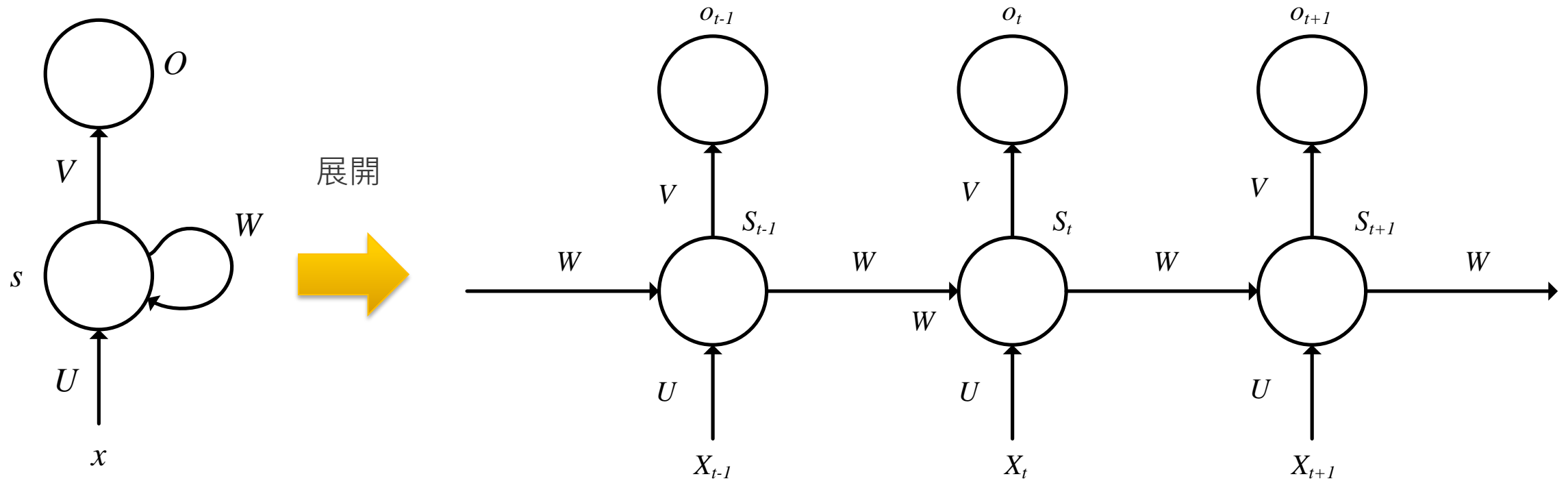
- 傳統RNN cell
- 套用LSTM新型態



designed by freepik

傳統RNN cell

- 最原始RNN的cell如下(如同之前章節所教到的)
 - 但是此種RNN會有梯度消失的問題，所以有人就在想是否能修改一個RNN cell的算法



$$s_t = f(Ux_t + Ws_{t-1})$$

$$o_t = \text{softmax}(Vs_t).$$

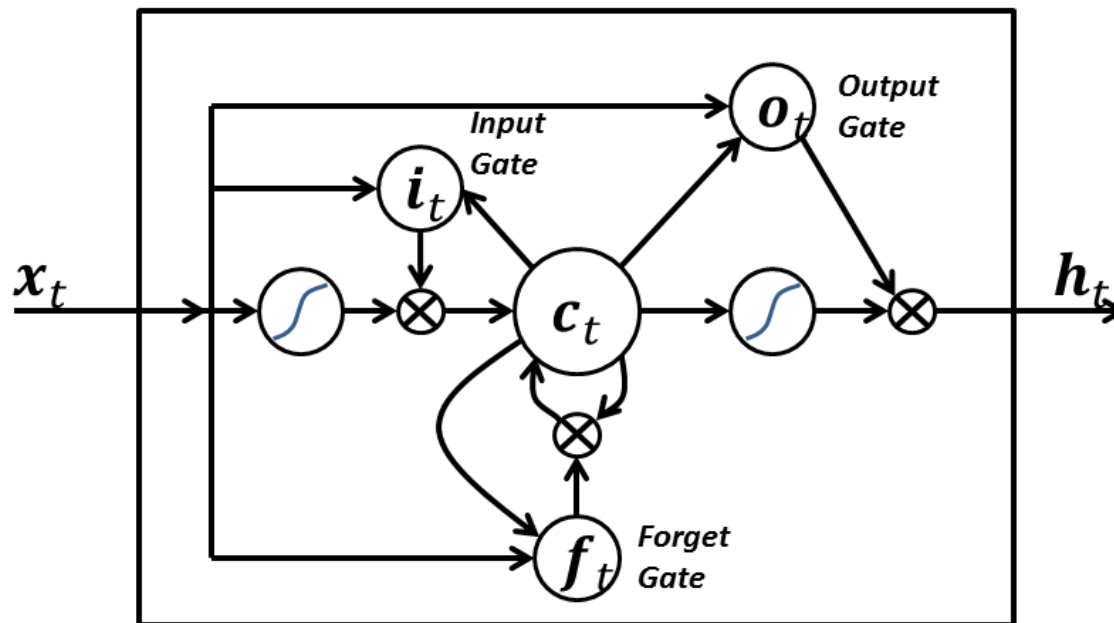
套用LSTM新型態

- 有人就提出了所謂的新RNN cell叫做LSTM
 - 用來取代原來的RNN cell
 - LSTM運算方法我們後面會教到



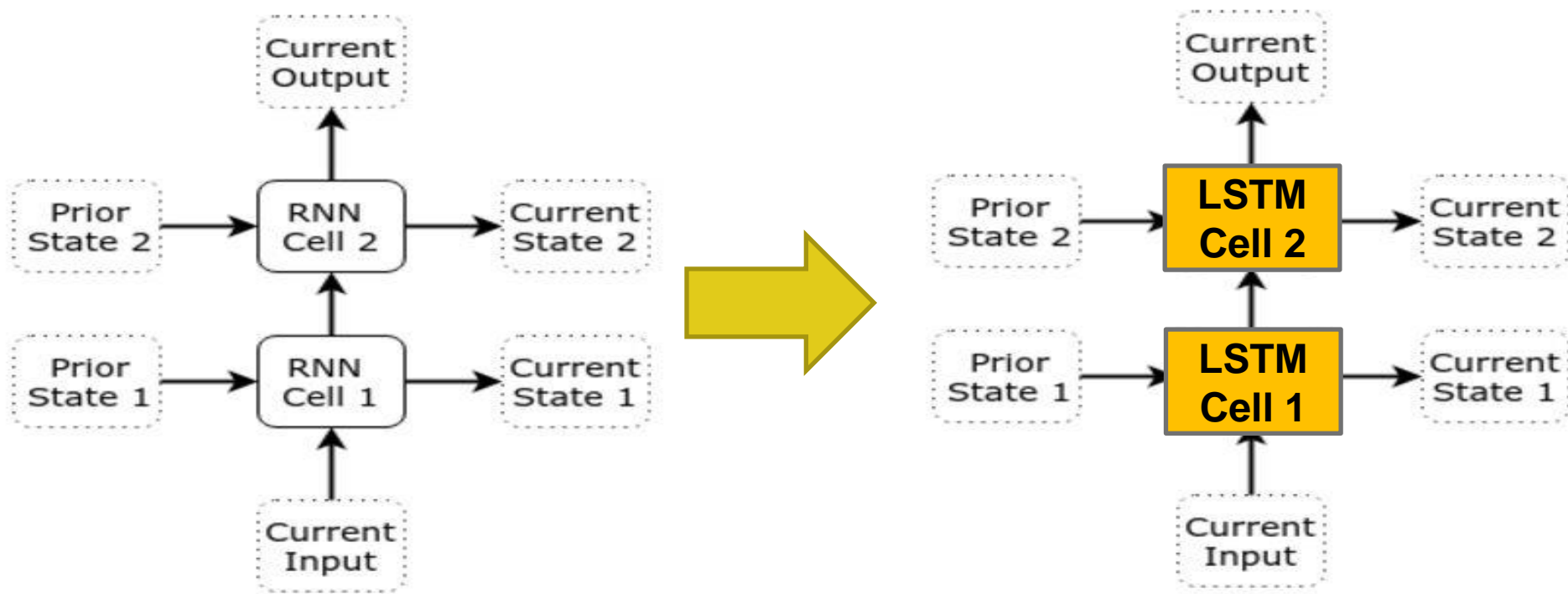
套用LSTM新型態

- LSTM是一種新型態的RNN cell
 - 由於獨特的結構，LSTM適合處理時間軸非常長的事件



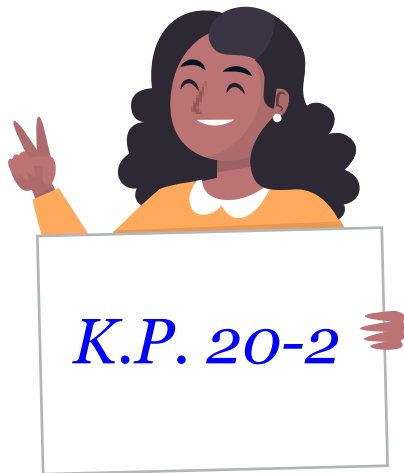
套用LSTM新型態

- LSTM僅只有改變RNN cell内部的算法，許多之前提到RNN的觀念仍然適用
 - 例如建構一個深度RNN，可以改用LSTM cell來建構



20-2: LSTM介紹

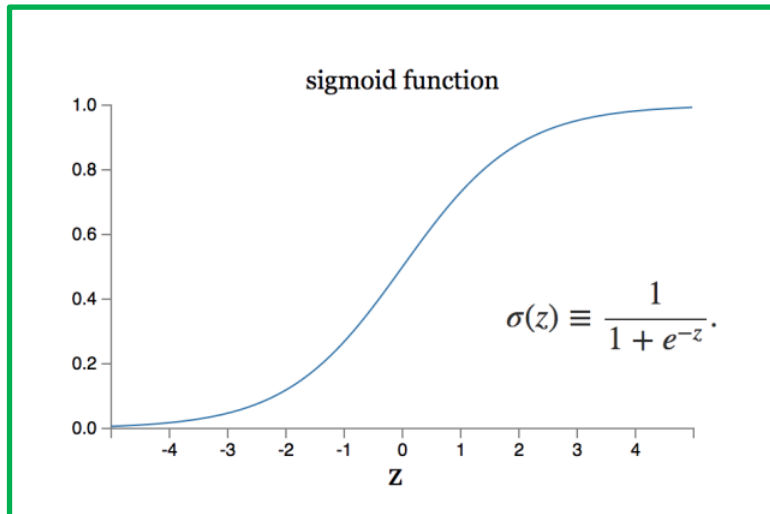
- LSTM運算方法
- LSTM物理意義



designed by freepik

LSTM運算方法

- 一個LSTM運算方法如下
 - LSTM定義了三種閘門(gate)，分別是input gate、output gate、forget gate，分別用 i_t 、 o_t 、 f_t 來表示，t代表時間為t的當下
 - 三種閘門有各自的參數W、U、b
 - σ 代表sigmoid函數，其輸出介於0~1之間



$$\begin{aligned} i_t &= \sigma(W_i h_{t-1} + U_i x_t + b_i) \\ o_t &= \sigma(W_o h_{t-1} + U_o x_t + b_o) \\ f_t &= \sigma(W_f h_{t-1} + U_f x_t + b_f) \end{aligned}$$

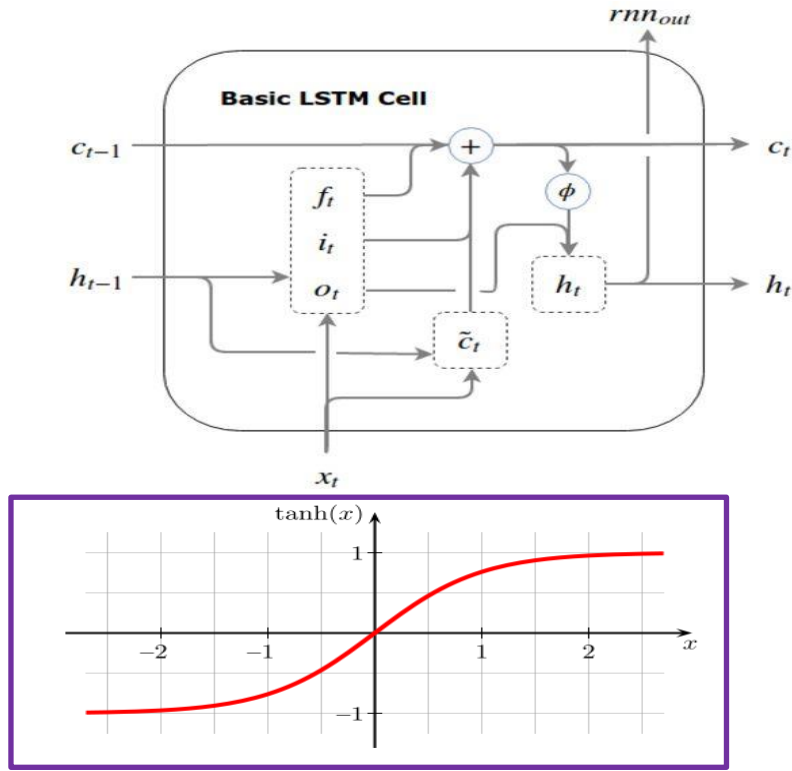
$$\begin{aligned} \tilde{c}_t &= \phi(W h_{t-1} + U x_t + b) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \end{aligned}$$

$$h_t = o_t \odot \phi(c_t)$$

$$\text{rnn}_{out} = h_t$$

LSTM運算方法

- LSTM定義了兩種狀態，分別是c state以及h state
 - \hat{c}_t 表示輸入對於原始狀態 c_t 的改變



$$\begin{aligned}i_t &= \sigma(W_i h_{t-1} + U_i x_t + b_i) \\o_t &= \sigma(W_o h_{t-1} + U_o x_t + b_o) \\f_t &= \sigma(W_f h_{t-1} + U_f x_t + b_f)\end{aligned}$$

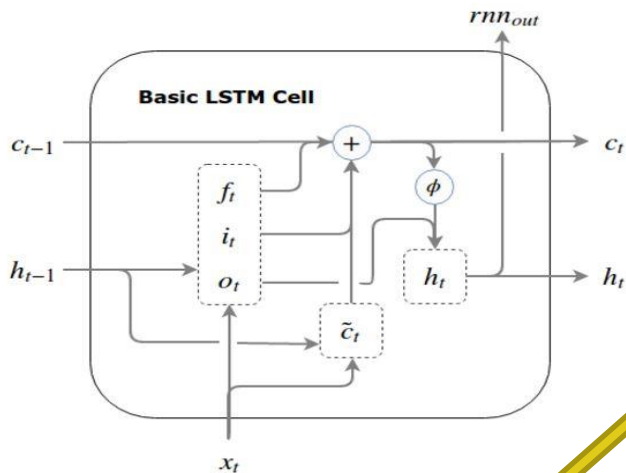
$$\begin{aligned}\tilde{c}_t &= \phi(W h_{t-1} + U x_t + b) \\c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t\end{aligned}$$

$$h_t = o_t \odot \phi(c_t)$$

$$rnn_{out} = h_t$$

LSTM 運算方法

- 接下來這條式子是LSTM最關鍵的式子
 - c state的值等於上一個c state值乘以保留比例加上c state改變乘以輸入比例
 - c state與上一個時間點的c state呈現線性遞迴關係，也因為這條式子，所以才解決梯度消失的問題



$$\begin{aligned}i_t &= \sigma(W_i h_{t-1} + U_i x_t + b_i) \\o_t &= \sigma(W_o h_{t-1} + U_o x_t + b_o) \\f_t &= \sigma(W_f h_{t-1} + U_f x_t + b_f)\end{aligned}$$

$$\tilde{c}_t = \phi(W h_{t-1} + U x_t + b)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

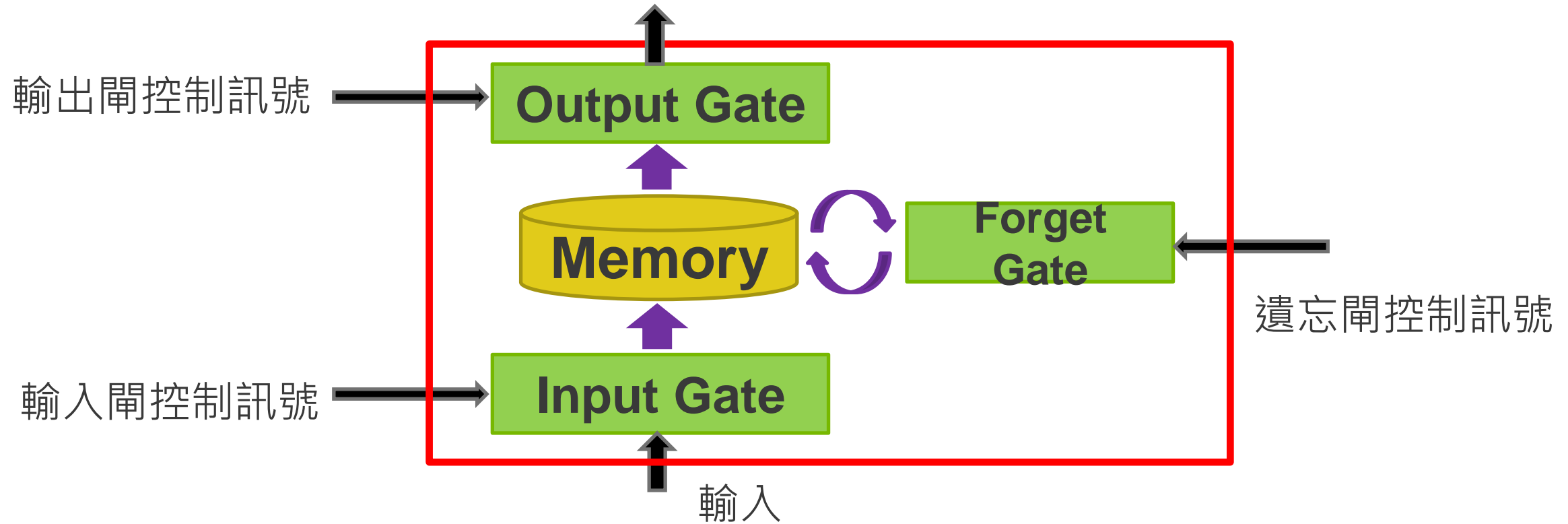
$$h_t = o_t \odot \phi(c_t)$$

$$rnn_{out} = h_t$$

關鍵的式子

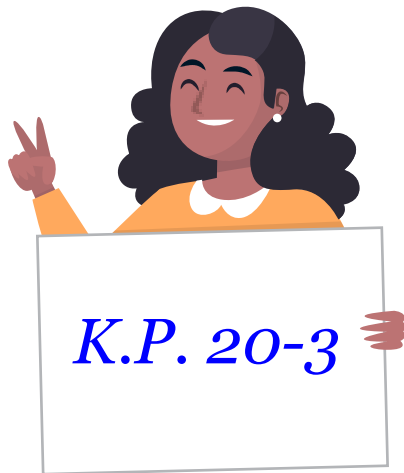
LSTM 物理意義

- input gate、forget gate、output gate就如同控制訊號一樣
 - 例如當input gate為1時，輸入有100%可以進入RNN cell



20-3: GRU 介紹

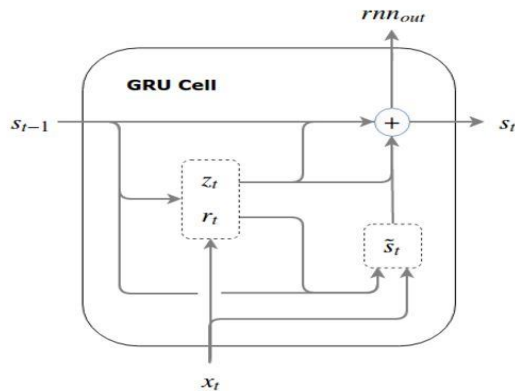
- GRU 介紹
- 其他RNN cell
- 新舊RNN cell之間的差異



designed by freepik

GRU介紹

- GRU是一種新型態的RNN cell
 - 是從LSTM演化過來的
 - 其參數比LSTM少
 - LSTM有兩個閘門，但GRU改為兩個



$$r_t = \sigma(W_r s_{t-1} + U_r x_t + b_r)$$

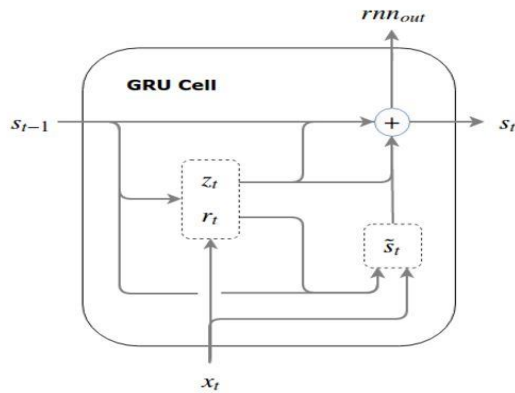
$$z_t = \sigma(W_z s_{t-1} + U_z x_t + b_z)$$

$$\tilde{s}_t = \phi(W(r_t \odot s_{t-1}) + U x_t + b)$$

$$s_t = z_t \odot s_{t-1} + (1 - z_t) \odot \tilde{s}_t$$

GRU介紹

- GRU一樣有相鄰s state之間的遞迴式子
 - 跟LSTM一樣不會有梯度消失的問題



$$r_t = \sigma(W_r s_{t-1} + U_r x_t + b_r)$$

$$z_t = \sigma(W_z s_{t-1} + U_z x_t + b_z)$$

$$\tilde{s}_t = \phi(W(r_t \odot s_{t-1}) + Ux_t + b)$$

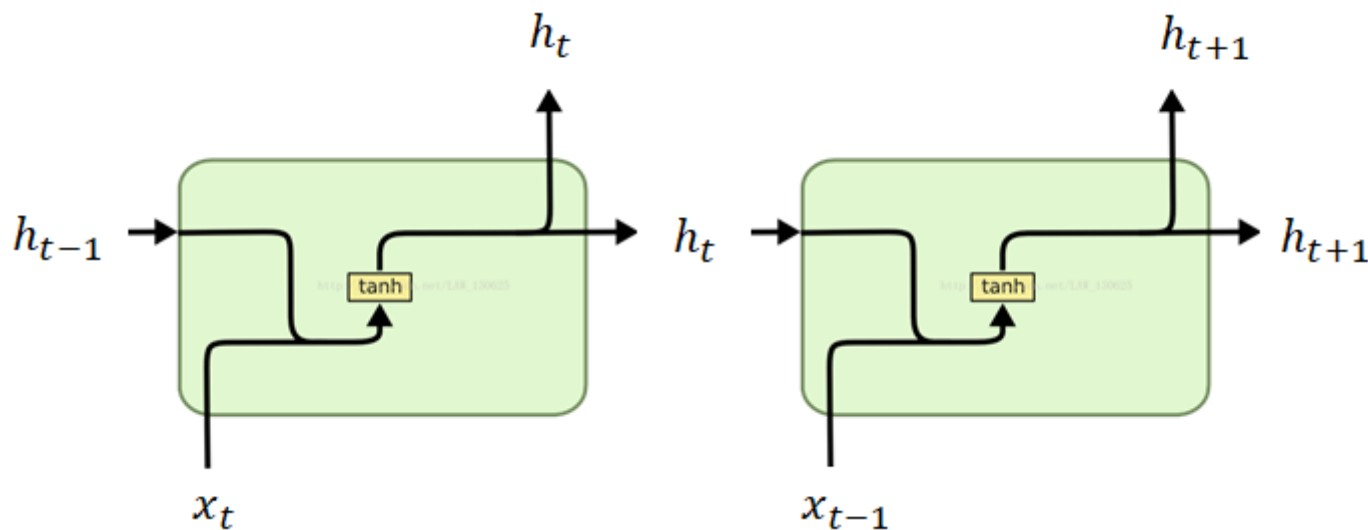
$$s_t = z_t \odot s_{t-1} + (1 - z_t) \odot \tilde{s}_t$$

其他RNN cell

- 自從LSTM被提出以後，因為其效果非常好，有不少論文嘗試著改良它
 - 例如GRU就是其中一個例子
 - 目前LSTM也衍生出一堆不同的變形，同學如果有興趣可以再去深入研究

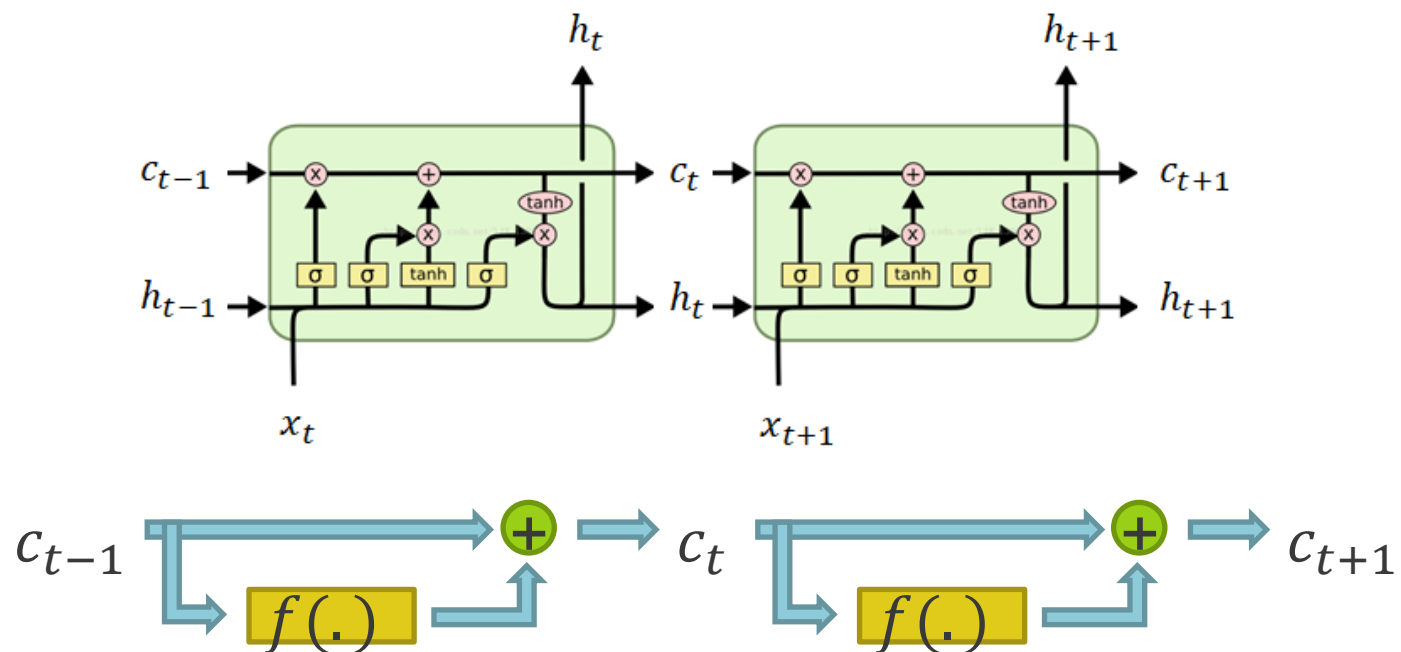
新舊RNN cell之間的差異

- 傳統RNN cell概念如下
 - 假設狀態為 h_t ， h_{t+1} 為 h_t 經由RNN cell所轉換成的



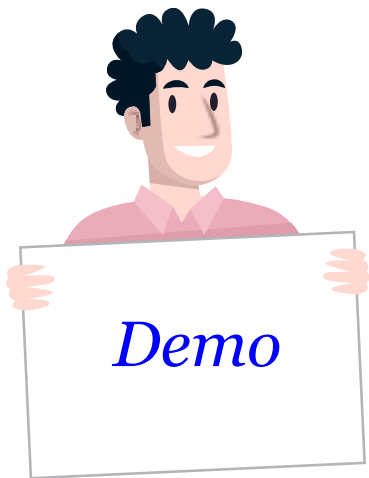
新舊RNN cell之間的差異

- 新RNN cell概念如下
 - 假設狀態為 c_t ， c_{t+1} 為 c_t 加上 c_t 經由RNN cell所轉換的值的總和
 - 這樣的觀念跟Resnet一樣



Demo 20-3

- 使用LSTM做MNIST分類
- 使用GRU做MNIST分類
- 使用LSTM做垃圾郵件分類



designed by freepik

線上Corelab

- **題目1：使用LSTM做MNIST分類**
 - 完成RNN cell程式碼，使用LSTM做MNIST分類
- **題目2：使用GRU做MNIST分類**
 - 完成RNN cell程式碼，使用GRU做MNIST分類
- **題目3：使用LSTM做垃圾郵件分類**
 - 完成RNN cell程式碼，使用LSTM做垃圾郵件分類

本章重點精華回顧

- RNN cell
- LSTM介紹
- GRU介紹



Lab: LSTM/GRU 分類

- **Lab01:使用LSTM做MNIST分類**
- **Lab02:使用GRU做MNIST分類**
- **Lab03:使用LSTM做垃圾郵件分類**

Estimated time:

20 minutes

