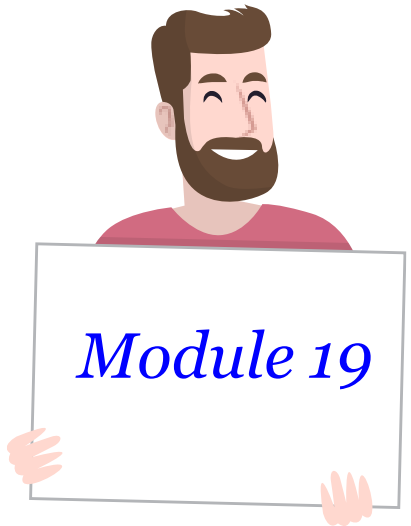
RNN網路建構



designed by freepik

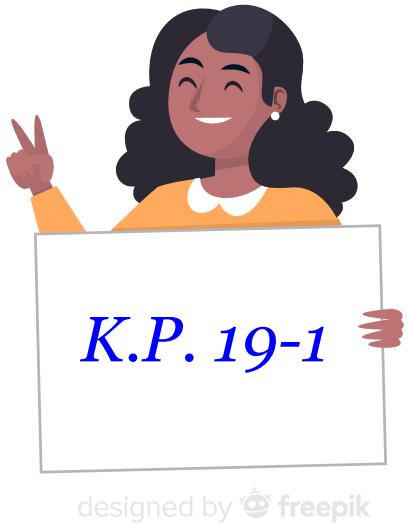
Estimated time:
45 min.

學習目標

- 19-1: RNN網路的種類
- 19-2: BPTT
- 19-3: 梯度消失以及梯度爆炸問題

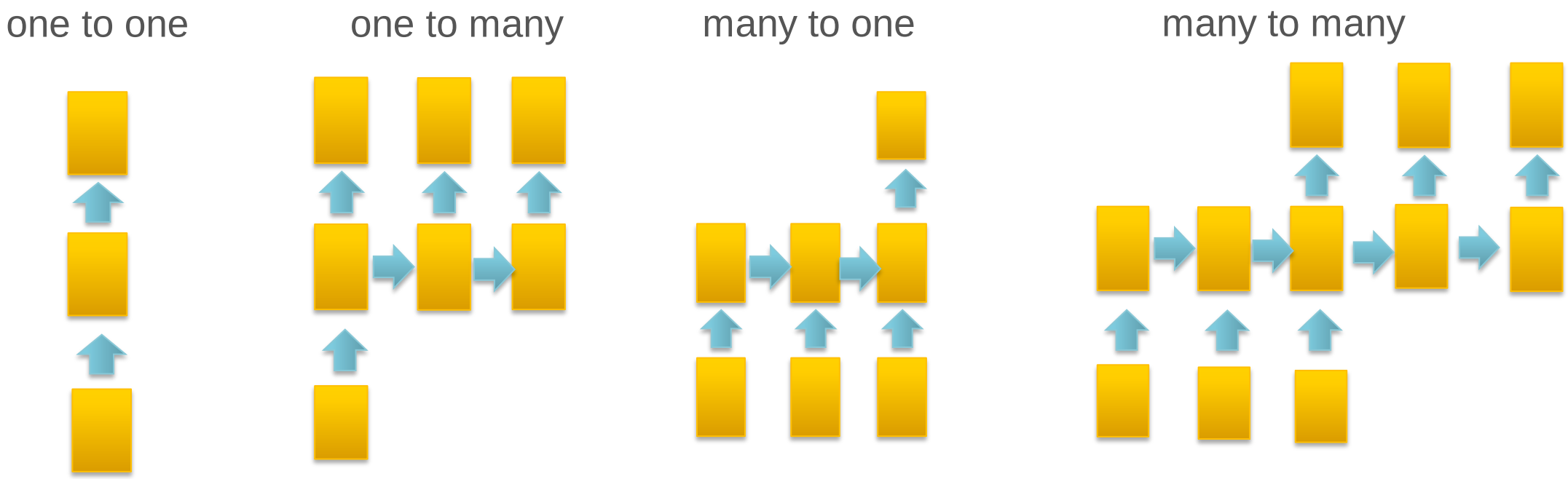
19-1: RNN網路的種類

- RNN網路種類
- RNN網路種類應用
- 深度RNN



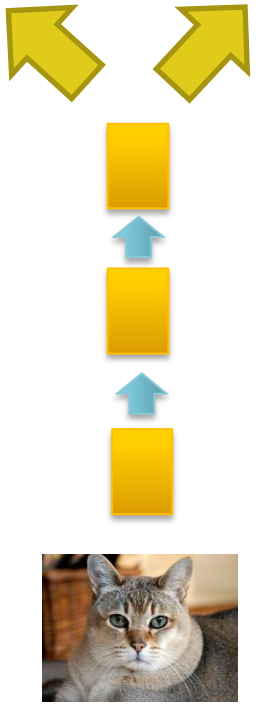
RNN網路種類

- **RNN網路的種類非常多，可以根據輸入以及輸出的數量分成**
  - **one to one、one to many、many to one、many to many**
  - **不同種類的應用所適用的RNN網路也不一樣**



RNN網路種類應用

- **One to One指的是輸入只有一個，輸出也只有一個**
  - **其實就是DNN神經網路，因此在講RNN網路的時候比較少討論它**
  - **例如貓狗影像分類就是一個例子**

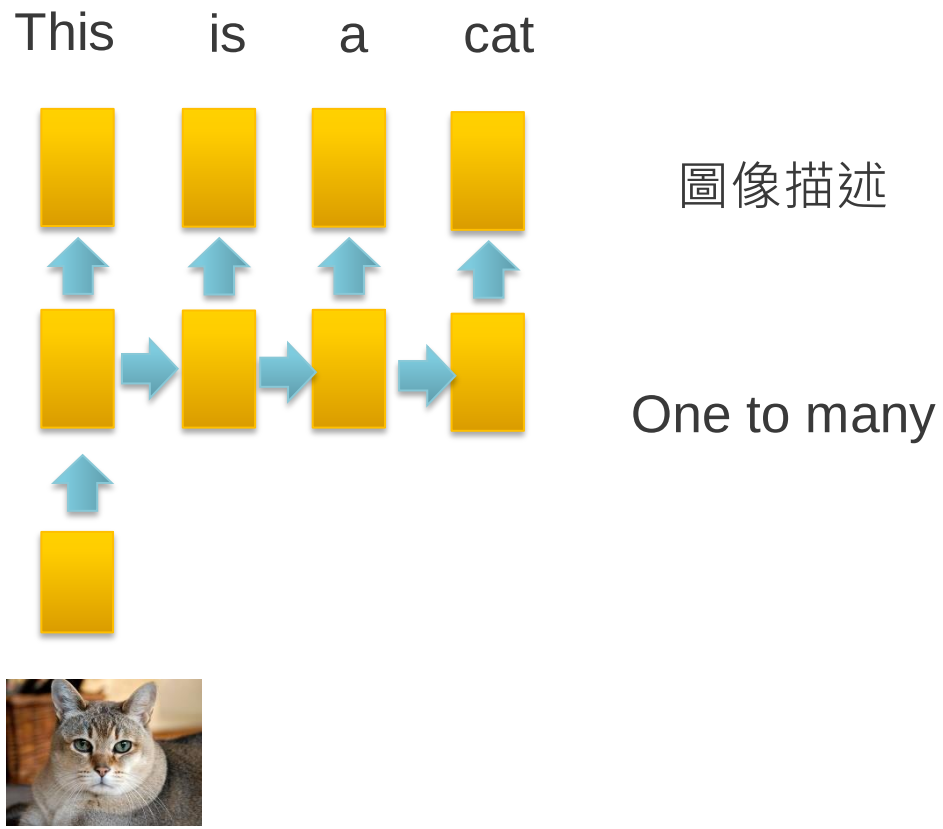


影像分類

One to one (DNN)

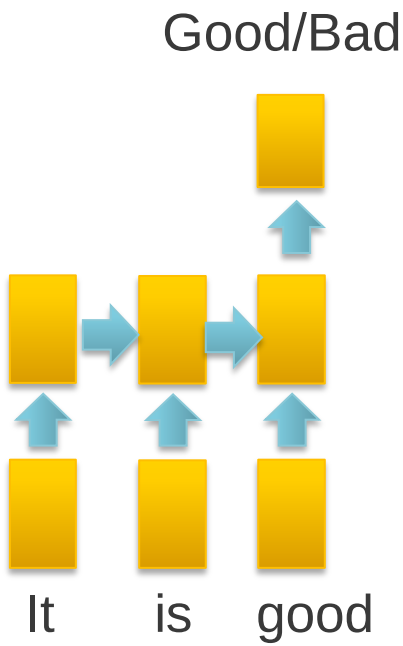
RNN網路種類應用

- **One to Many指的是輸入只有一個，輸出有多個**
  - **最有名的就是圖像描述，給予此種RNN一張照片，RNN會根據此照片輸出一段描述的文字**



RNN網路種類應用

- **Many to One指的是輸入只有多個，輸出只有一個**
- **最有名的就是電影影評分類，給予此種RNN一段影評，RNN會根據此影評來判斷此作者對於這部電影的評價是正向還是負向**

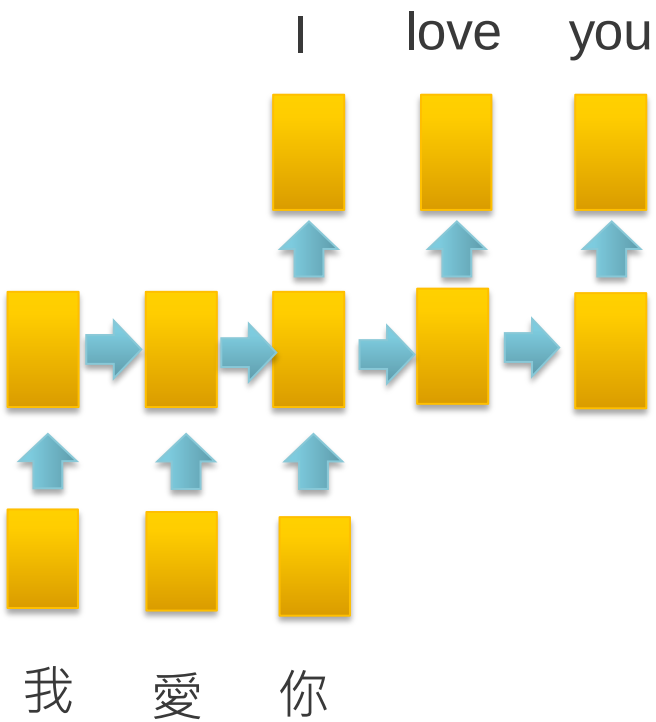


電影影評分類

Many to one

RNN網路種類應用

- **Many to Many指的是輸入只有多個，輸出也有多個**
- **最有名的就是語言翻譯，給予此種RNN一段話，RNN會根據此段話翻譯成不同的語言**

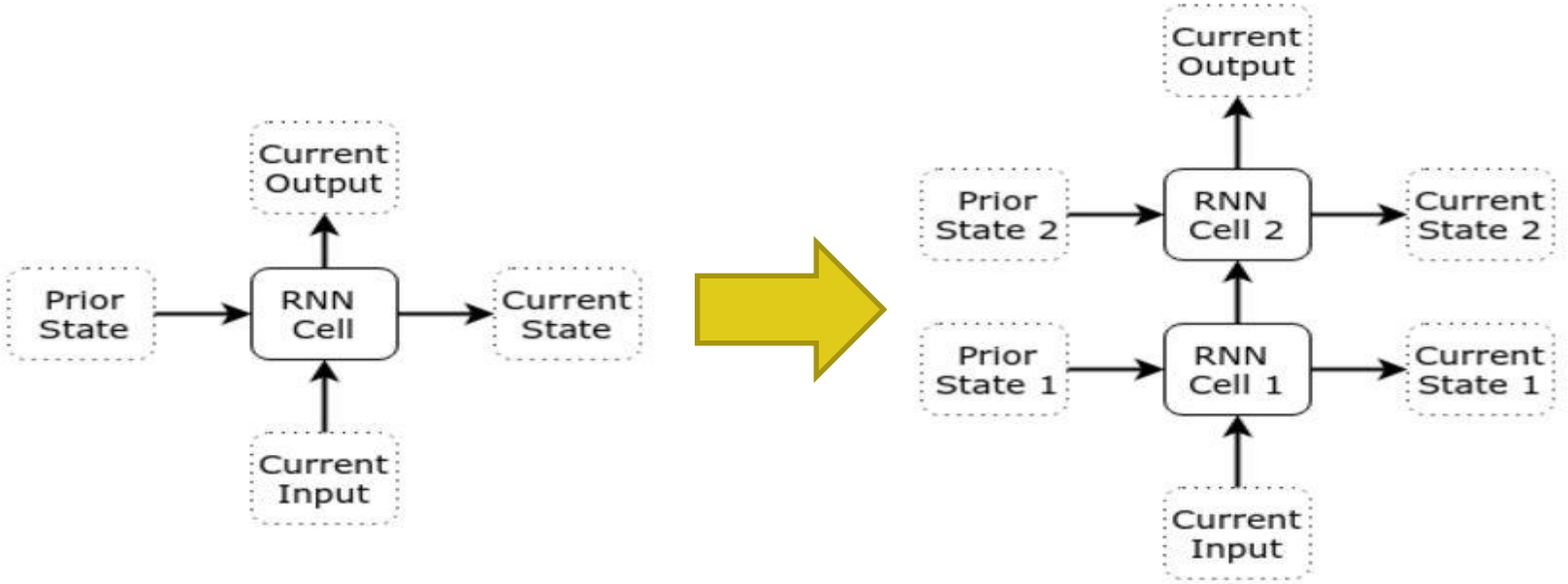


語言翻譯

Many to many

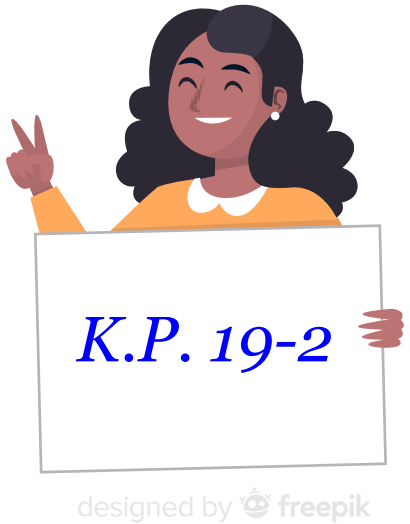
深度RNN

- RNN除了有不同種類外，我們也可以在空間軸上面把RNN疊得更”深”
  - 此RNN可以稱為深度RNN



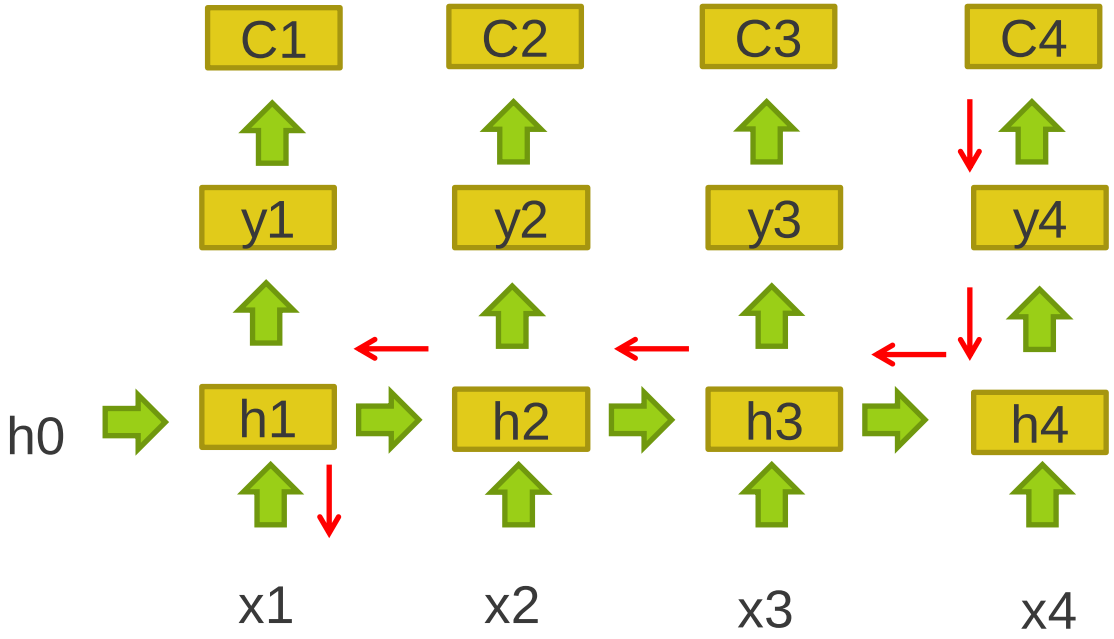
19-2: BPTT

- BPTT
- BPTT的問題
- Truncated BPTT



BPTT

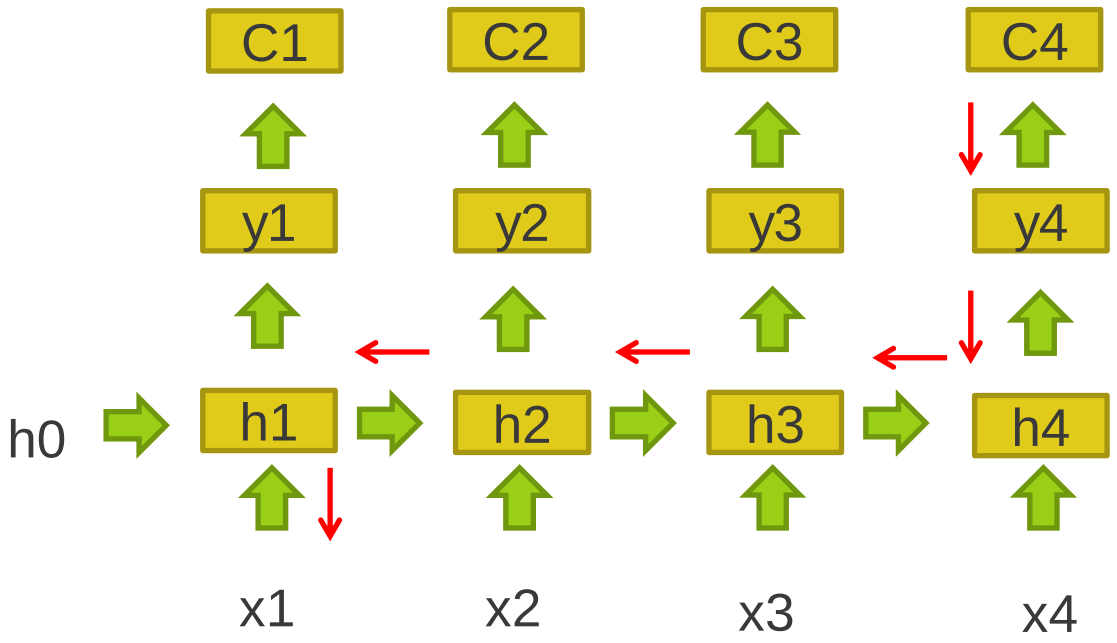
- **BPTT(Backpropagation through time)是一種訓練RNN的方法**
  - **其實就跟DNN、CNN裡面的Backpropagation一樣，只是BPTT是用在RNN這種與時間軸有關的神經網路**



$$\frac{\partial C_4}{\partial w} = \frac{\partial C_4}{\partial y4} * \frac{\partial y_4}{\partial h4} * \frac{\partial h_4}{\partial h3} * \frac{\partial h_3}{\partial h2} * \frac{\partial h_2}{\partial h1} * \frac{\partial h_1}{\partial w}$$

BPTT

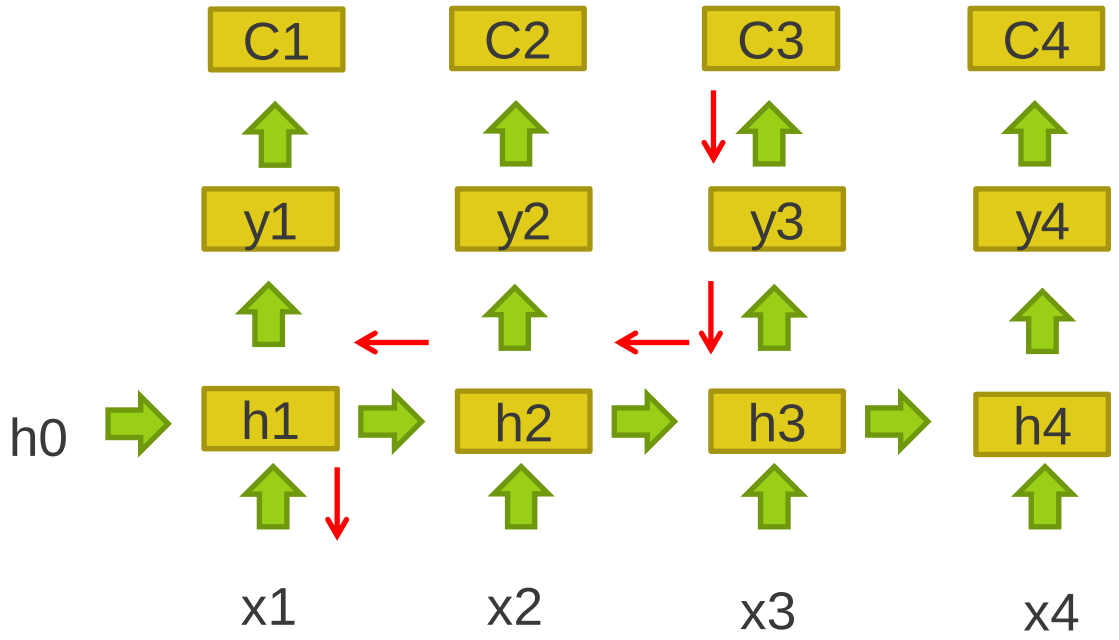
- **BPTT會從最後一個time step依序往回傳做參數上的修正**



$$\frac{\partial C_4}{\partial w} = \frac{\partial C_4}{\partial y4} * \frac{\partial y_4}{\partial h4} * \frac{\partial h_4}{\partial h3} * \frac{\partial h_3}{\partial h2} * \frac{\partial h_2}{\partial h1} * \frac{\partial h_1}{\partial w}$$

BPTT

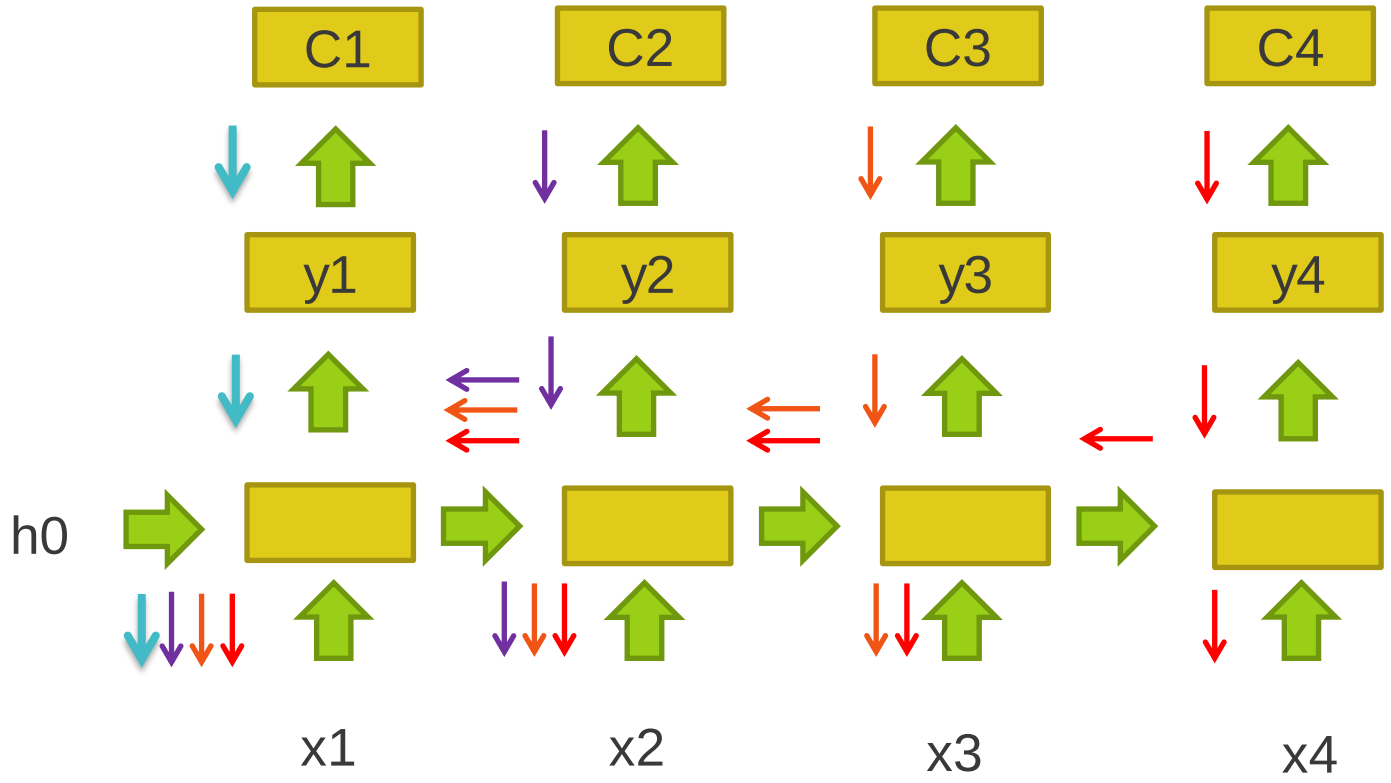
- 最後一個time step修正完後，再從倒數第二個time step回傳做參數修正



$$\frac{\partial C_3}{\partial h1} = \frac{\partial C_3}{\partial y3} * \frac{\partial y_3}{\partial h3} * \frac{\partial h_3}{\partial h2} * \frac{\partial h_2}{\partial h1} * \frac{\partial h_1}{\partial w}$$

BPTT

- 將所有time step回傳做參數修正後，整體示意圖如下

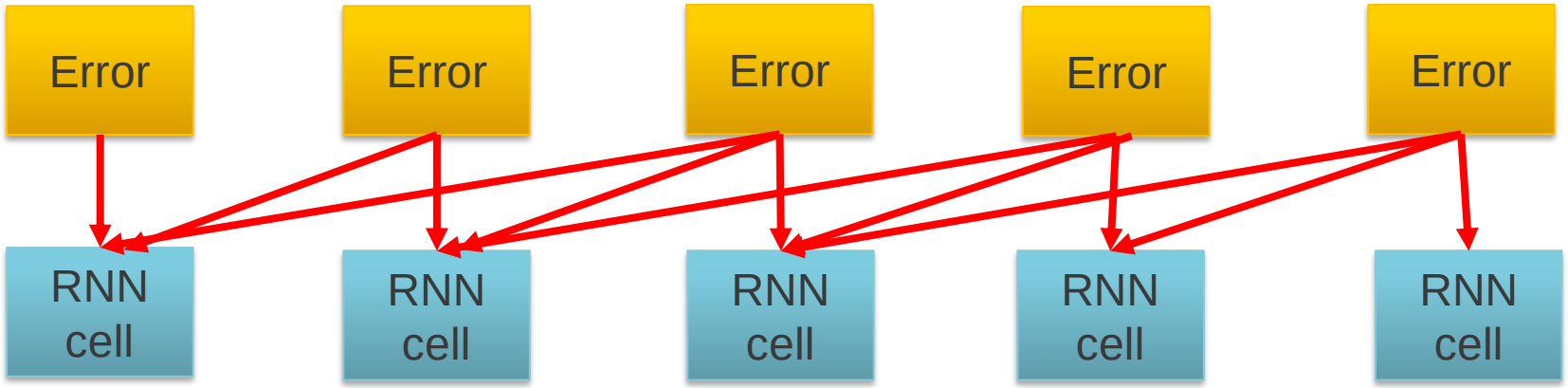


BPTT的問題

- 當RNN的time step很長的時候，BPTT會很沒有效率
  - 因為每個time step都要流回到第一個time step上面，導致很浪費時間
- 為了解決BPTT沒效率的問題，有人就提出了Truncated BPTT

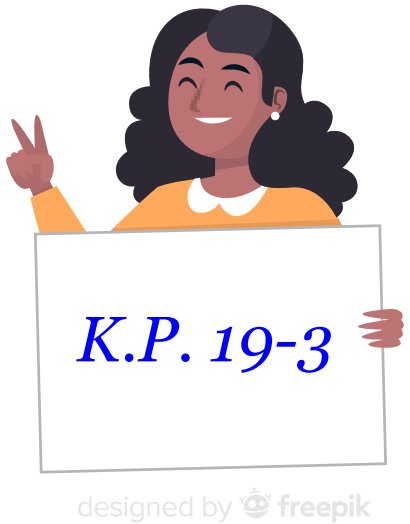
Truncated BPTT

- **Truncated BPTT是修改過後了BPTT，其概念是限制每個time step往前傳的步數**
  - **例如下圖， Truncated BPTT只限制每個time step往回傳兩步**



19-3: 梯度消失以及梯度爆炸問題

- 梯度消失以及梯度爆炸
- 解決梯度消失



梯度消失以及梯度爆炸

- 早期做RNN一直很難成功，因為往往RNN網路的time step很大，很容易產生梯度消失以及梯度爆炸的問題
  - 目前解決的方法是直接換一種RNN cell的算法(後面章節會教到)

梯度消失以及梯度爆炸

- 我們可以觀察到，在做BPTT時，會有很多h微分連乘項，如果這些h微分連乘項大於1，那麼只要time step一長，很容易讓數值非常大並產生overflow
  - 這就是所謂的梯度爆炸
  - 梯度爆炸最簡單的解法是如果數值大於某個門檻，我們就將整個連乘項設定成門檻值

$$\frac{\partial C_t}{\partial h1} = \frac{\partial C_t}{\partial y_t} * \frac{\partial y_t}{\partial h_t} * \frac{\partial h_t}{\partial h_{t-1}} * \frac{\partial h_{t-1}}{\partial h_{t-2}} * \cdots * \frac{\partial h_3}{\partial h_2} * \frac{\partial h_2}{\partial h_1}$$

梯度消失以及梯度爆炸

- 我們可以觀察到，在做BPTT時，會有很多h微分連乘項，如果這些h微分連乘項介於0~1之間，那麼只要time step一長，很容易讓數值變得非常小
  - 這就是所謂的梯度消失
- 梯度消失是RNN難以解決的問題之一

$$\frac{\partial C_t}{\partial h1} = \frac{\partial C_t}{\partial y_t} * \frac{\partial y_t}{\partial h_t} * \frac{\partial h_t}{\partial h_{t-1}} * \frac{\partial h_{t-1}}{\partial h_{t-2}} * \cdots * \frac{\partial h_3}{\partial h_2} * \frac{\partial h_2}{\partial h_1}$$

解決梯度消失

- 梯度消失經過一連串的研究後發現，我們必須要讓狀態h之間呈現線性遞迴的狀態，這樣在做h微分的時候才能控制其數字穩定
  - 於是基於這個概念，專家們提出了新的RNN cell，叫做LSTM及GRU(後面章節會教)

兩個h狀態是線性遞迴

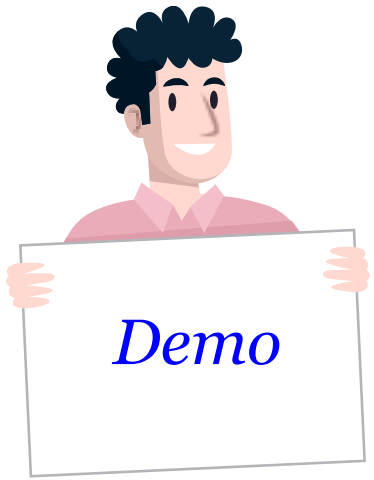
$$h_n = h_{n-1} + \ldots\ldots$$

$$\frac{\partial C_t}{\partial h1} = \frac{\partial C_t}{\partial y_t} * \frac{\partial y_t}{\partial h_t} * \frac{\partial h_t}{\partial h_{t-1}} * \frac{\partial h_{t-1}}{\partial h_{t-2}} * \cdots * \frac{\partial h_3}{\partial h_2} * \frac{\partial h_2}{\partial h_1}$$

$$\frac{\partial h_t}{\partial h_{t-1}} = 1$$

Demo 19-3

- 觀察垃圾郵件資料集
- 將句子轉換成token
- 建立RNN網路做垃圾郵件分類



線上Corelab

- 題目1：觀察垃圾郵件資料集
  - 將垃圾郵件資料集前五筆資料顯示出來
- 題目2：將句子轉換成token
  - 將垃圾郵件資料集每個句子轉成token
- 題目3：使用RNN做垃圾郵件分類
  - 完成RNN cell程式碼做垃圾郵件分類

本章重點精華回顧

- RNN的種類
- BPTT的概念
- 梯度消失以及梯度爆炸

Lab:RNN垃圾郵件分類

- **Lab01: 觀察垃圾郵件資料集**
- **Lab02: 將句子轉換成token**
- **Lab03: 建立RNN網路做垃圾郵件分類**

Estimated time:
20 minutes


THANK YOU