

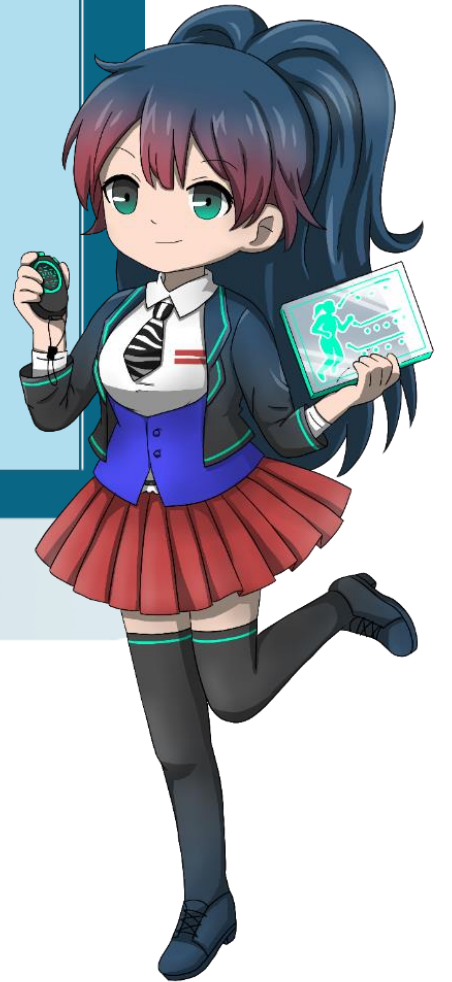
解密AI黑盒子



主題六：遞迴神經網路(RNN)

遞迴神經網路 (RNN)

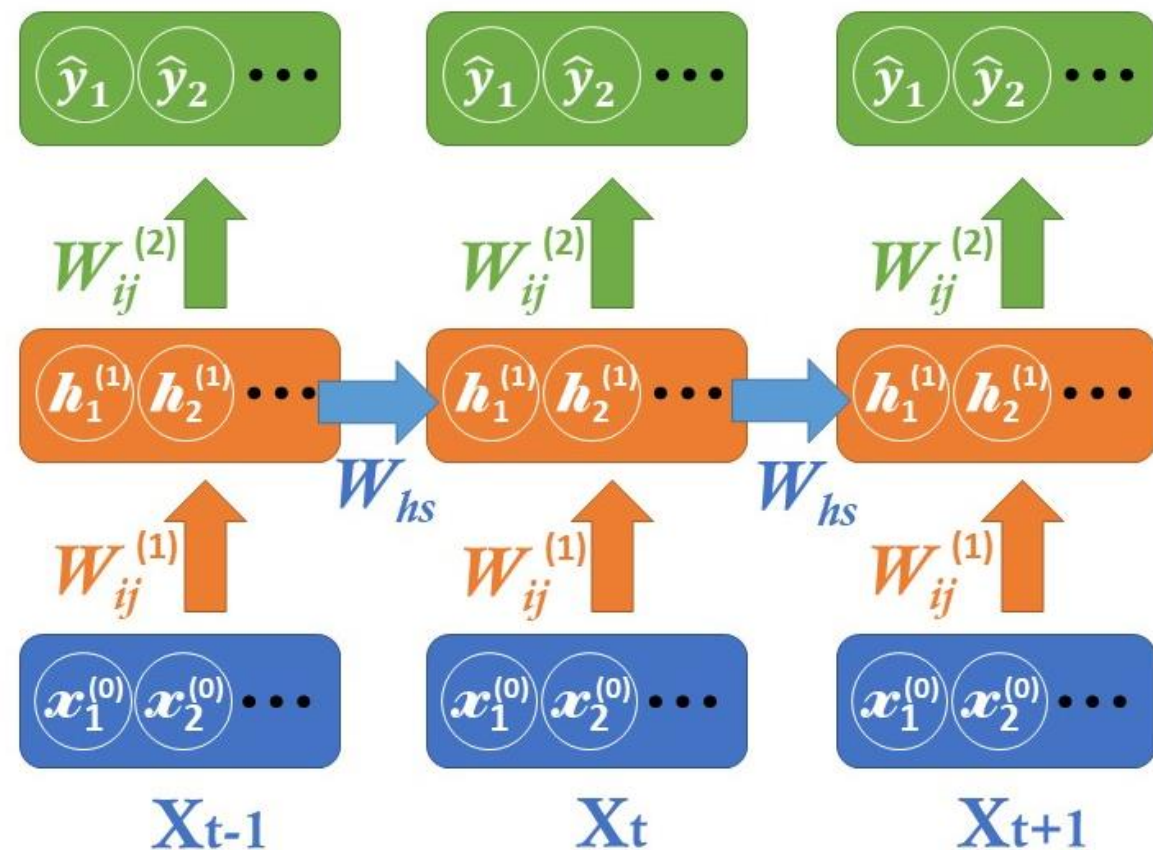
單元1



遞迴神經網路RNN

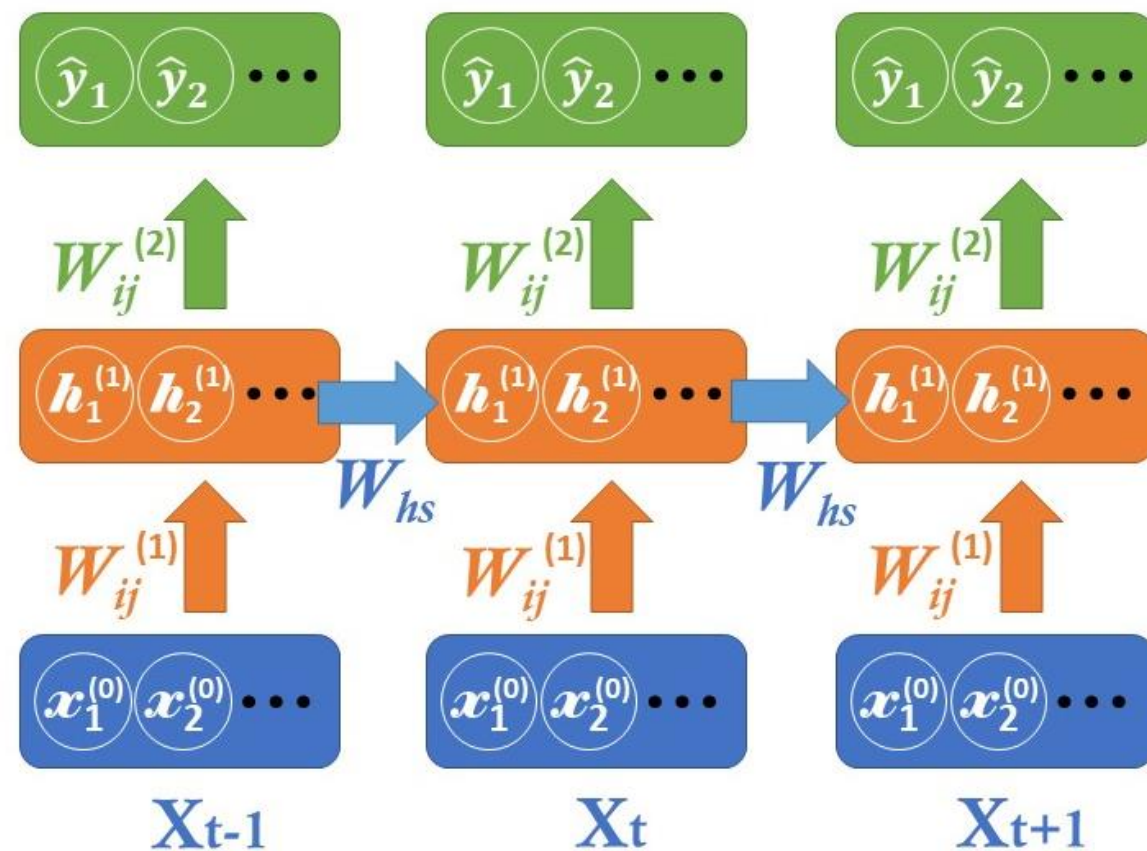
■ 遞迴神經網路(Recurrent Neural Network, RNN)

- RNN主要用於輸入資料間是有序列關聯性的，利用其序列關聯性可以增加特徵擷取的強度，若對調其序列先後，會影響其輸出結果。
- 輸入資料間的序列關聯性不一定是時間關係，空間關係也是可以的。
例：圖片的像素資料也可以將其由上而下，每一列都是一筆輸入資料，他們是有序列關聯性的。



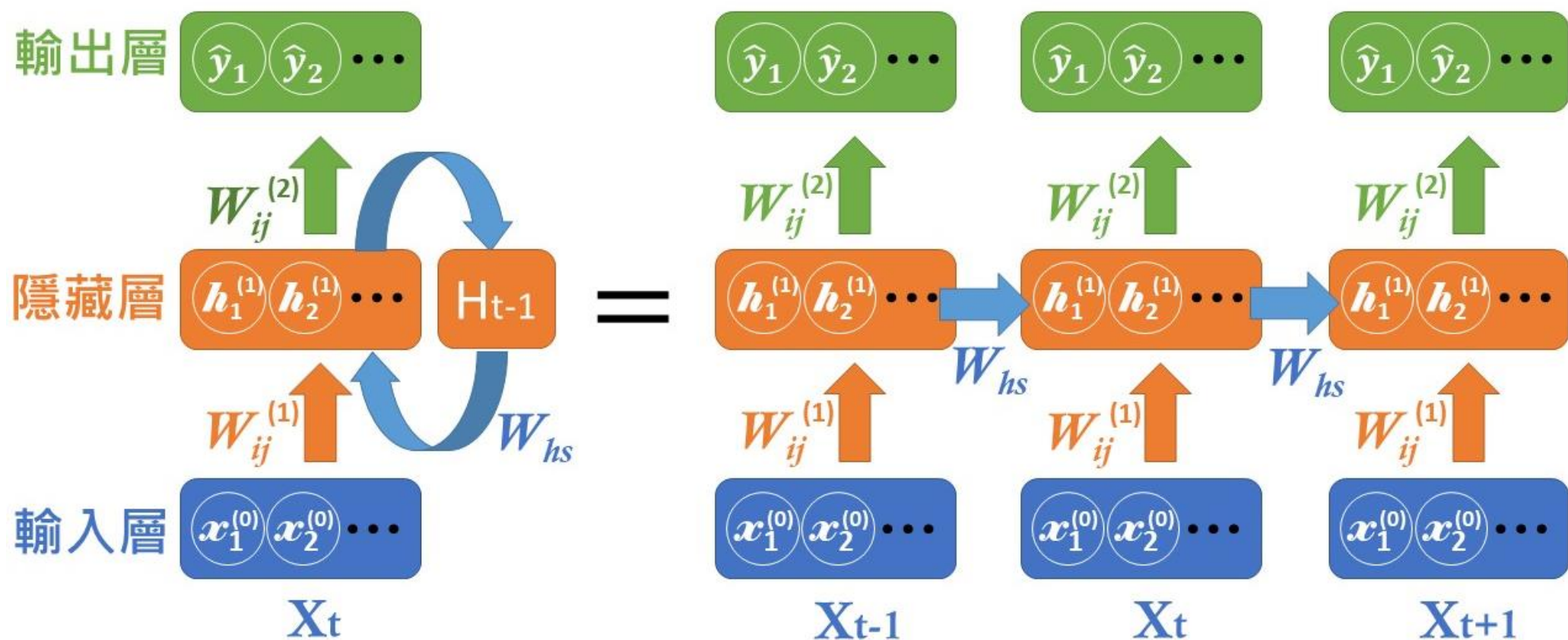
遞迴神經網路的架構

- RNN和前面所介紹的神經網路最大的差異就是引進了具有記憶功能的「隱藏層狀態 (Hidden State)」。
- RNN記憶的狀態通常是上一個序列樣本的隱藏層輸出資料H矩陣。
- RNN神經網路在計算輸出時不是只有計算當下輸入的樣本資料，而是會把前一個序列樣本的「隱藏層狀態 (Hidden State)」也當作輸入來計算。
- 這個記憶功能，可以影響到下一個、下兩個、、、或是下n個State，讓不同的State之間產生關聯性，幫忙RNN記憶之前看過樣本的訊息，並結合當前的樣本訊息得到輸出。



遞迴神經網路的架構

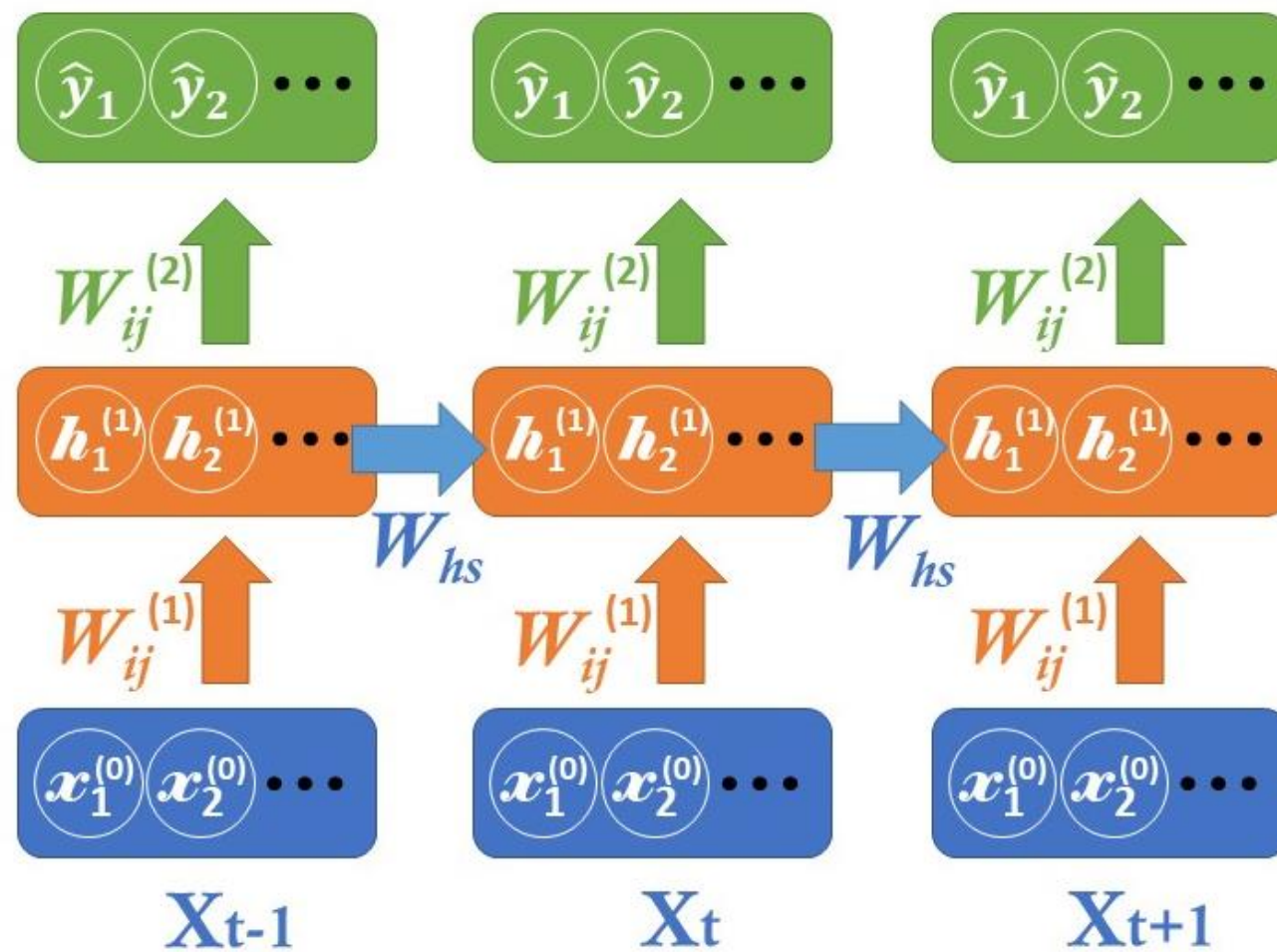
- RNN的模型：左邊是簡化版，右邊是詳細版，從圖可以看到每一個時間點的輸出都會受到前一次State的影響(記憶)。



遞迴神經網路的架構

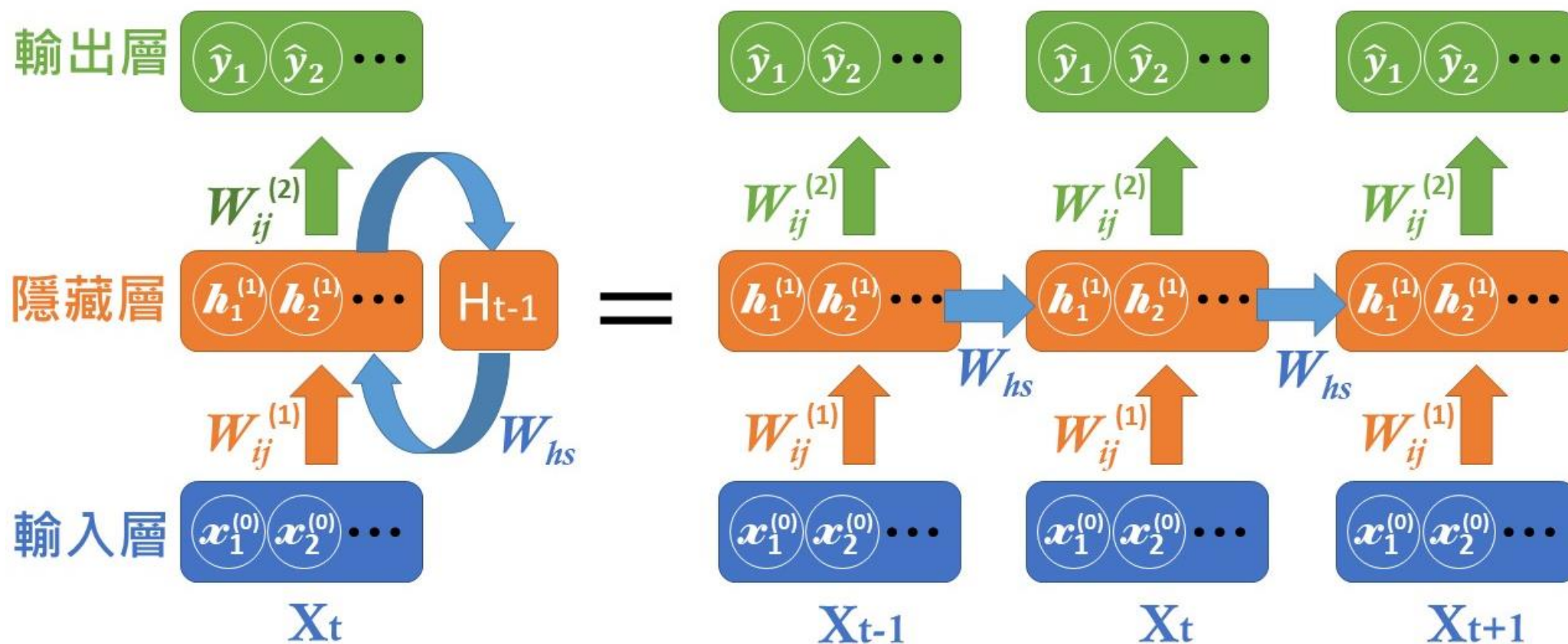
■ RNN每個序列點都會有3個權重，且不同序列的權重都是相同的，即權重共享：

- 輸入層到隱藏層的權重 $w_{ij}^{(1)}$
- 隱藏層到輸出層的權重 $w_{ij}^{(2)}$
- 當前序列隱藏層 H_t 到下一個序列隱藏層 H_{t+1} 的權重 w_{hs}



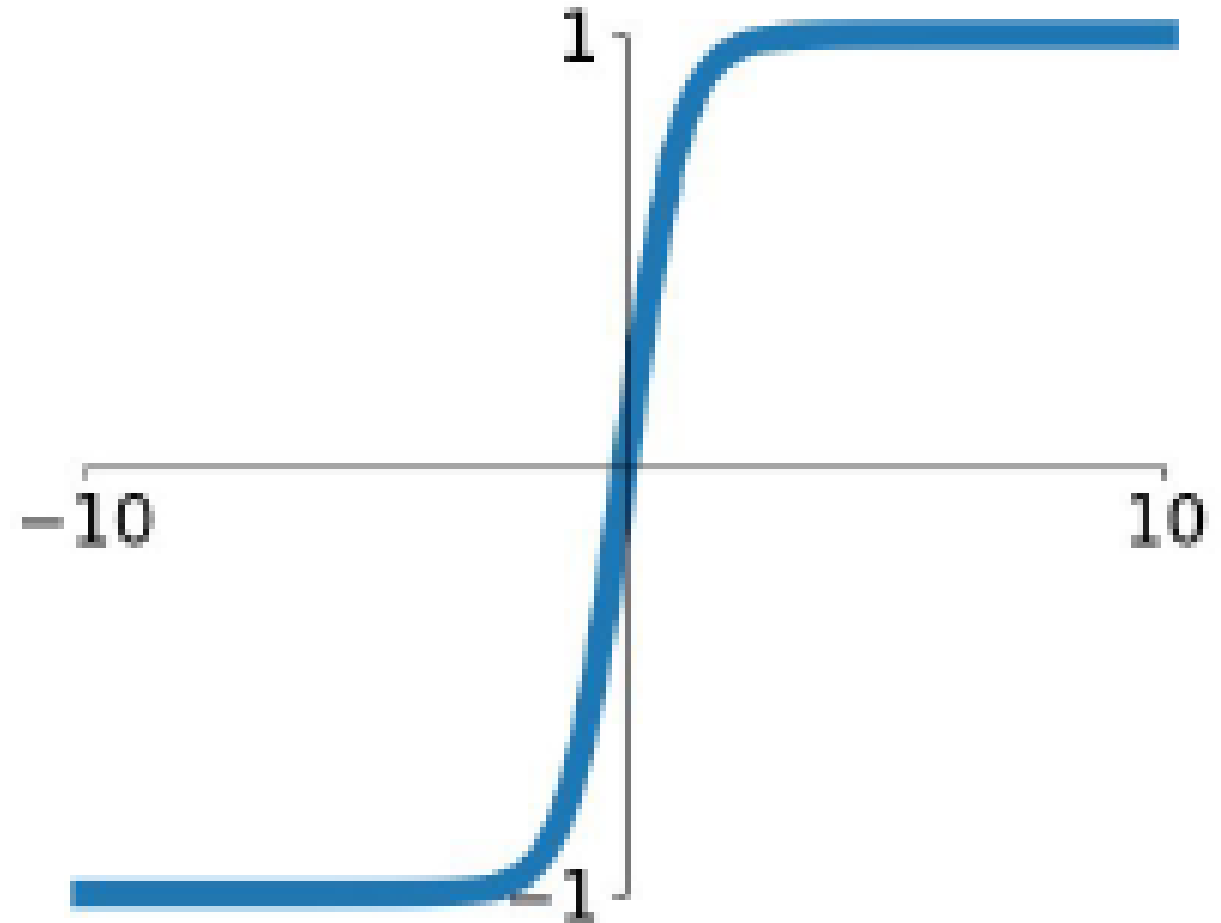
遞迴神經網路的架構

- 時序長短可變：可以處理任意長度的時間序列。
- 上圖中的RNN架構是一個最簡化的架構，較複雜的可以使用深度學習神經網路，也就是三層以上的隱藏層。



- 遞迴神經網隱藏層的激活函數通常使用 tanh。

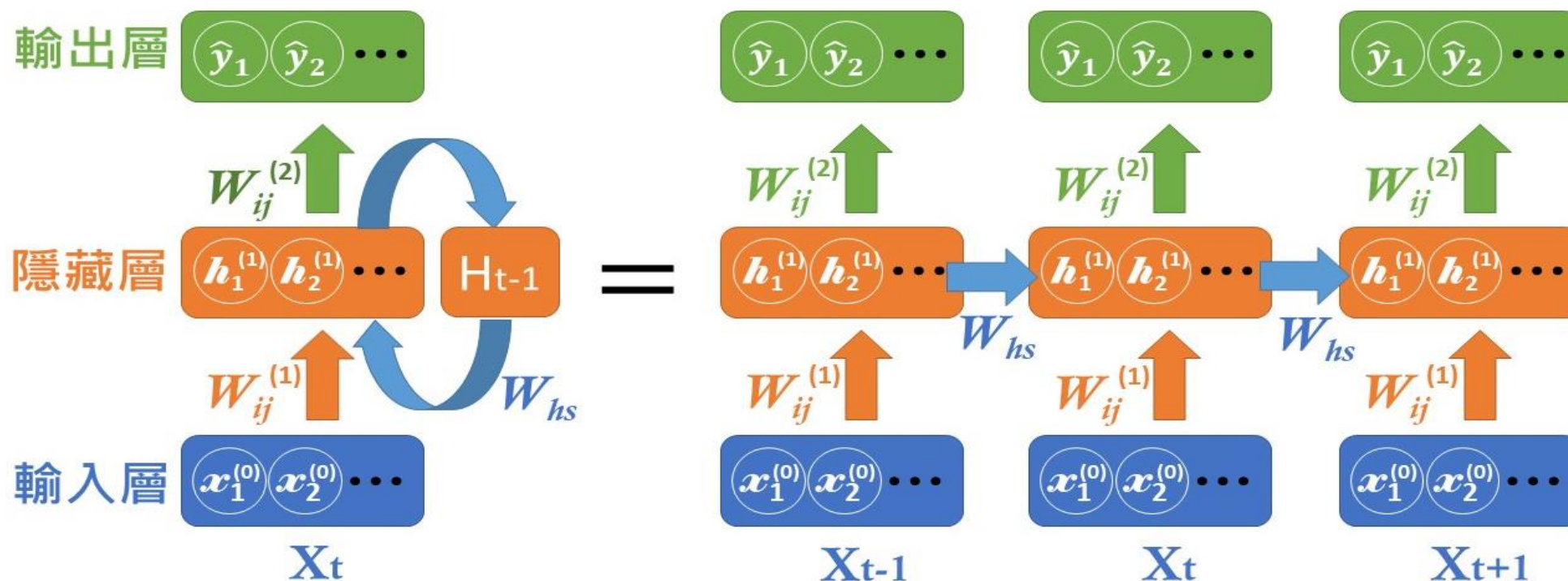
- $\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$ ，函數圖如右圖。
- 其微分為 $\frac{df}{dx} = 1 - f(x)^2$



遞迴神經網路的前向傳播

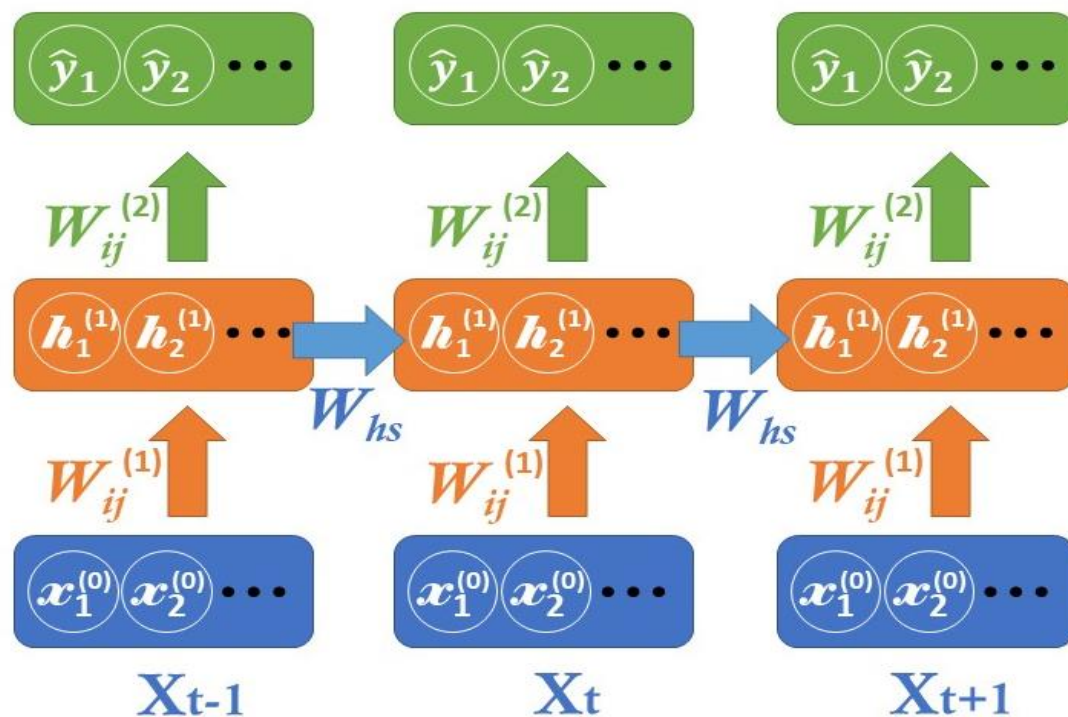
- 第 t 時序的隱藏層輸出 H_t ，除了當下時序的輸入 x_t 外，還必須考慮前一時序的隱藏層狀態 H_{t-1} 。

- $H_t = \tanh([輸入x_t] \times [權重W^{(1)}] + [狀態H_{t-1}] \times [權重W_{hs}] + [偏值B^{(1)}])$
- H_0 初始化為全為零的矩陣，故 $H_1 = \tanh([輸入x_0] \times [權重W^{(1)}] + [偏值B^{(1)}])$



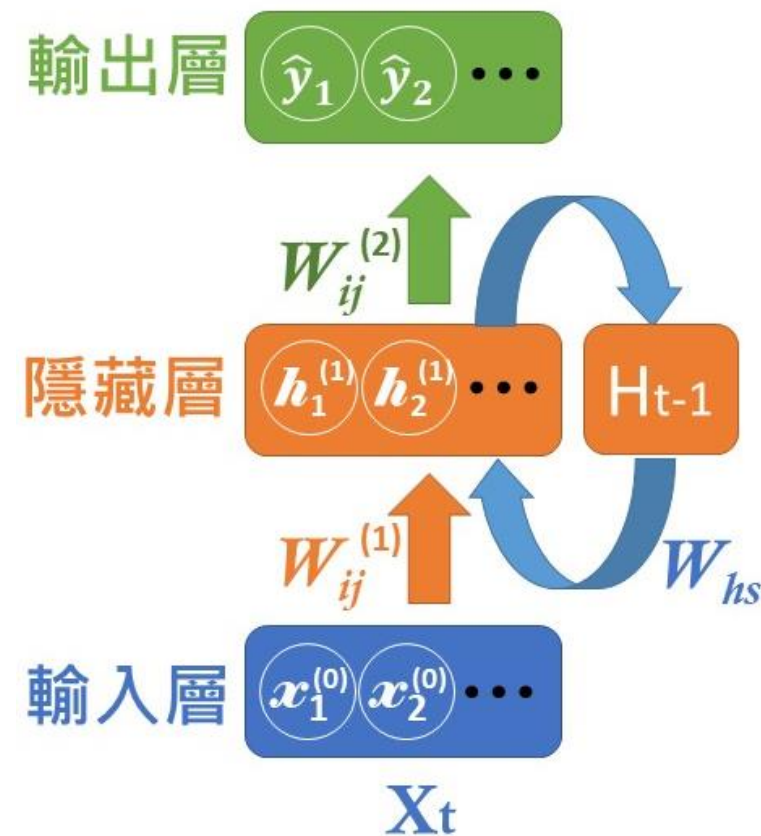
遞迴神經網路的前向傳播

- 假設最後的時序為 $t = n$ ，其隱藏層到輸出層：
 - 輸入 $[Z_j^{(2)}] = [\text{隱藏層輸出} H_n] \times [\text{權重} W^{(2)}] + [\text{偏值} B^{(2)}]$
 - 輸出根據用途(迴歸、二元分類、多元分類)



遞迴神經網路的反向傳播

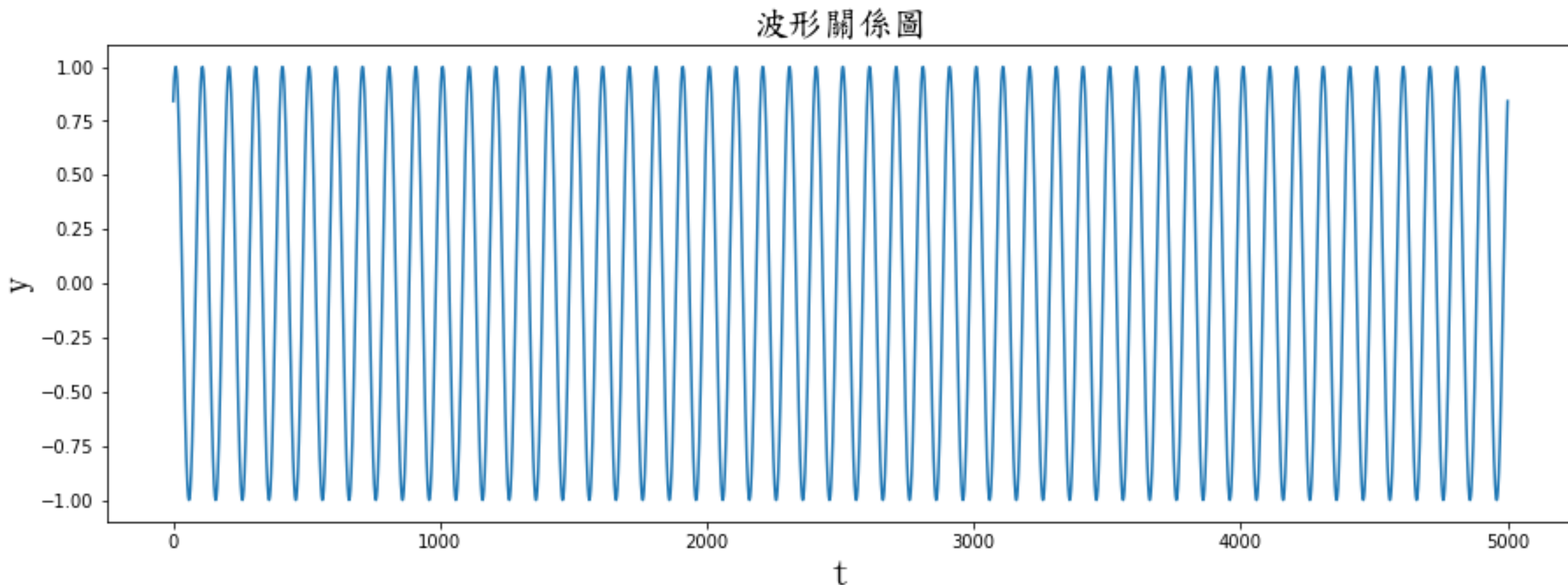
- 遞迴神經網路(RNN)的記憶時序反向傳播BPTT全名為 back-propagation through time
- 要考慮由最後時序 $t=n \Rightarrow t=n-1 \Rightarrow t=n-2$ 、 $\dots \Rightarrow t=1 \Rightarrow t=0$ ，2的時間序列反向傳播。
 - 求出Loss值對隱藏層記憶狀態權重 W_{hs} 的梯度 $\frac{\partial L}{\partial W_{hs}}$
- 如同一般多層神經網路，每一時序中由輸出層 \Rightarrow 隱藏層 \Rightarrow 輸入層的反向傳播。
 - 求出Loss值對隱藏層權重 $W_{ij}^{(2)}$ 的梯度： $\frac{\partial L}{\partial W_{ij}^{(2)}}$
 - 求出Loss值對輸入層權重 $W_{ij}^{(1)}$ 的梯度： $\frac{\partial L}{\partial W_{ij}^{(1)}}$
- 由於各時序的 W_{hs} 、 $W_{ij}^{(2)}$ 、 $W_{ij}^{(1)}$ 權重共享，故各時序的梯度，取平均值作為更新的依據。



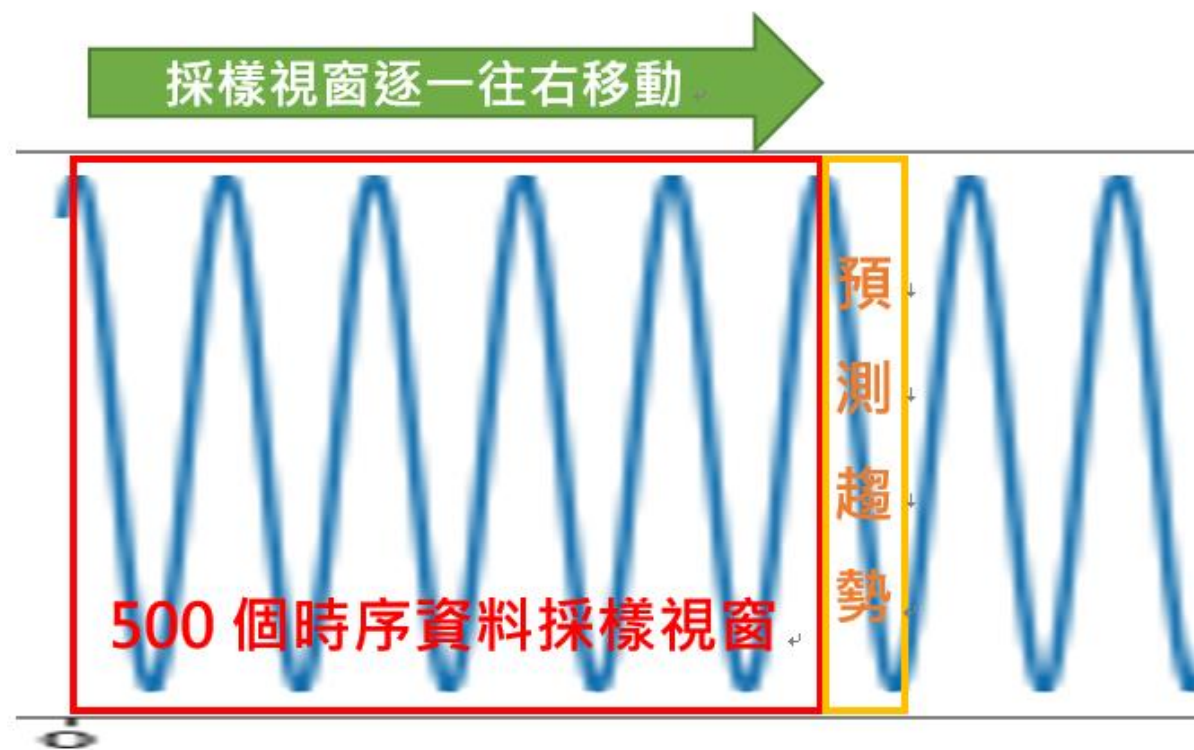
遞迴神經網路的兩大缺失

- 遞迴神經網路從理論上應該可以顧及所有過去時序的依賴，然而實際卻不理想，原因在於梯度消失(vanishing gradient)和梯度爆炸(exploding gradient)問題。
- RNN在做時序的反向傳播時，若時序非常長，從後面時序反向推回前面時序來求梯度時，因為連鎖規則，每一時序的梯度連續相乘非常多次。
 - 梯度消失：若時序的梯度有大量介於1和-1之間(即梯度絕對值 <1)，則推回前面的梯度非常小。
 - 梯度爆炸：若時序的梯度有大量大於1或小於-1(即梯度絕對值 >1)，則推回前面的梯度非常大。
- 梯度爆炸可以用梯度剪裁的方法來避免，將梯度大於1或小於-1的，剪裁為1或-1。
- 梯度消失就無法用剪裁的方法來處理。

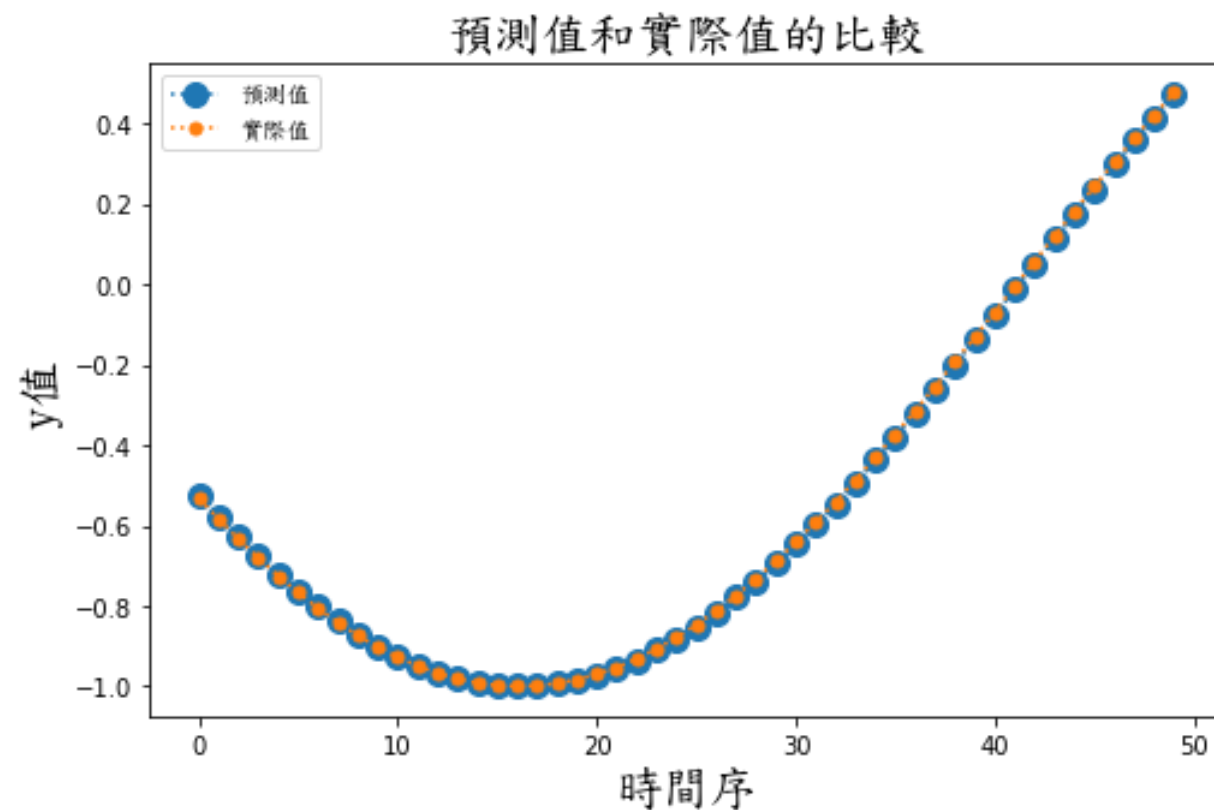
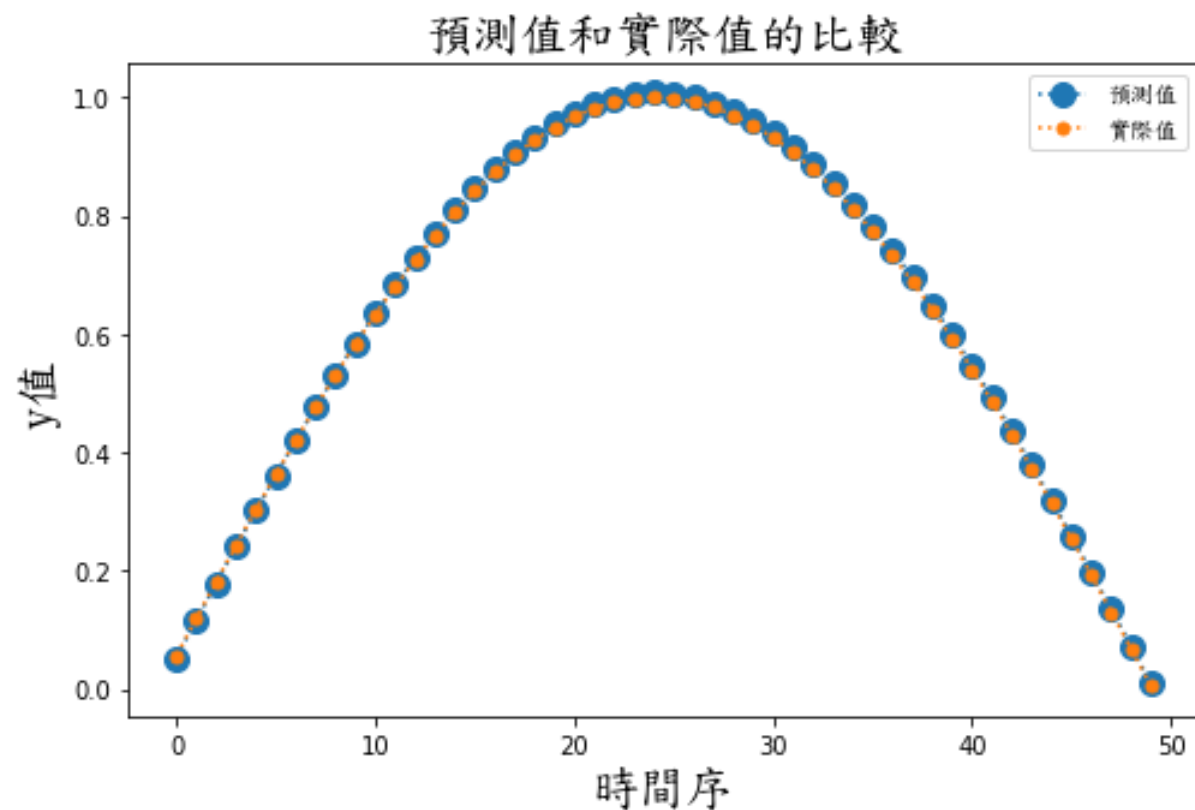
■ 如下圖為一個正弦波動的圖，由5001個點的數據 y 所組成。



- 如右圖，將數據每500個點(稱為採樣視窗)切成一個時序輸入特徵，緊接後面50點數據為預測值，總共可以產生(5001-550)筆的訓練資料。
- 搭建一個RNN神經網路，輸入訓練資料來學習波動趨勢：
 - 每個樣本有500個時序
 - 輸入層有1個節點
 - 隱藏層有64個節點
 - 輸出層有50個節點

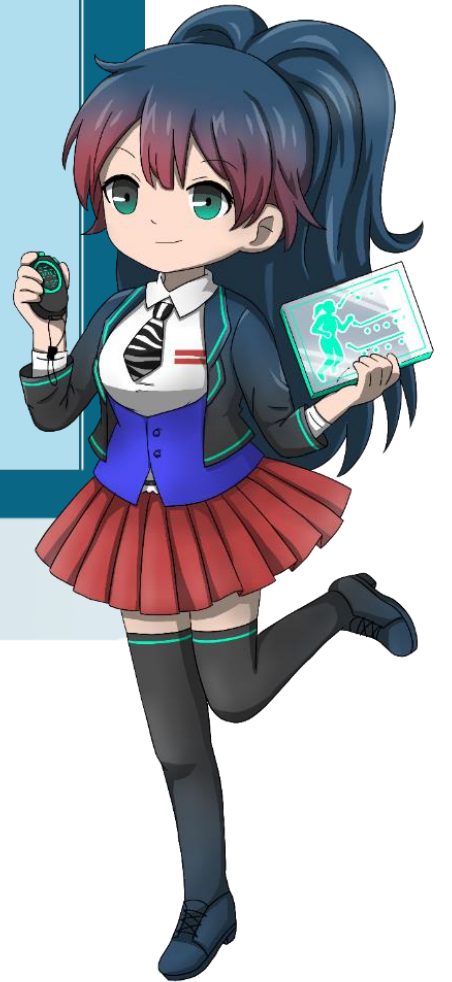


- 以動畫顯示連續100個時序資料經模型預測的50個輸出值和實際值間的差異，如右圖。

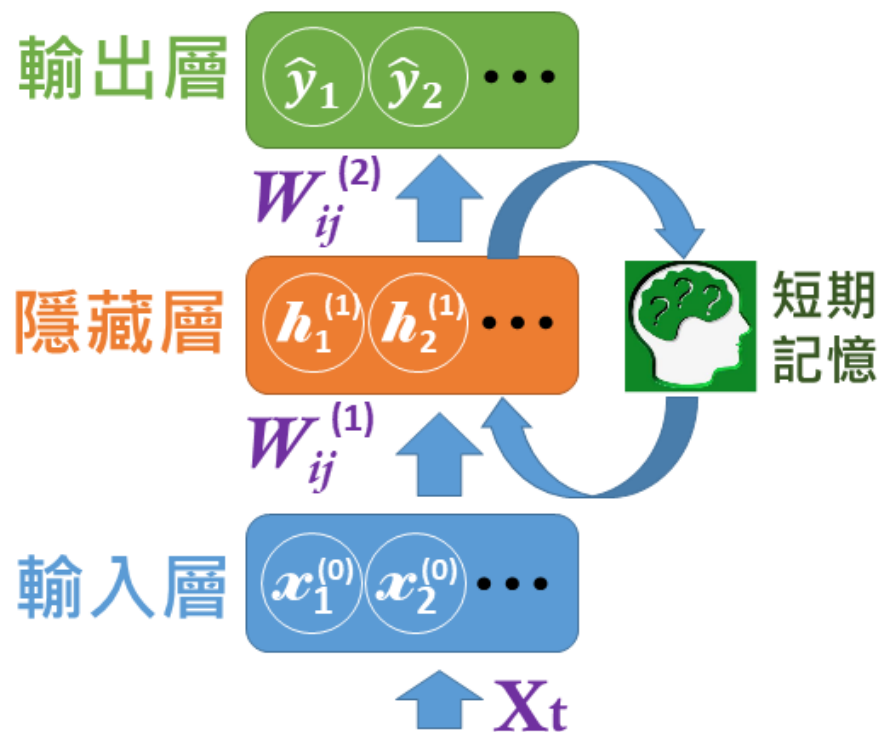


門控迴圈單元 (GRU)

單元2

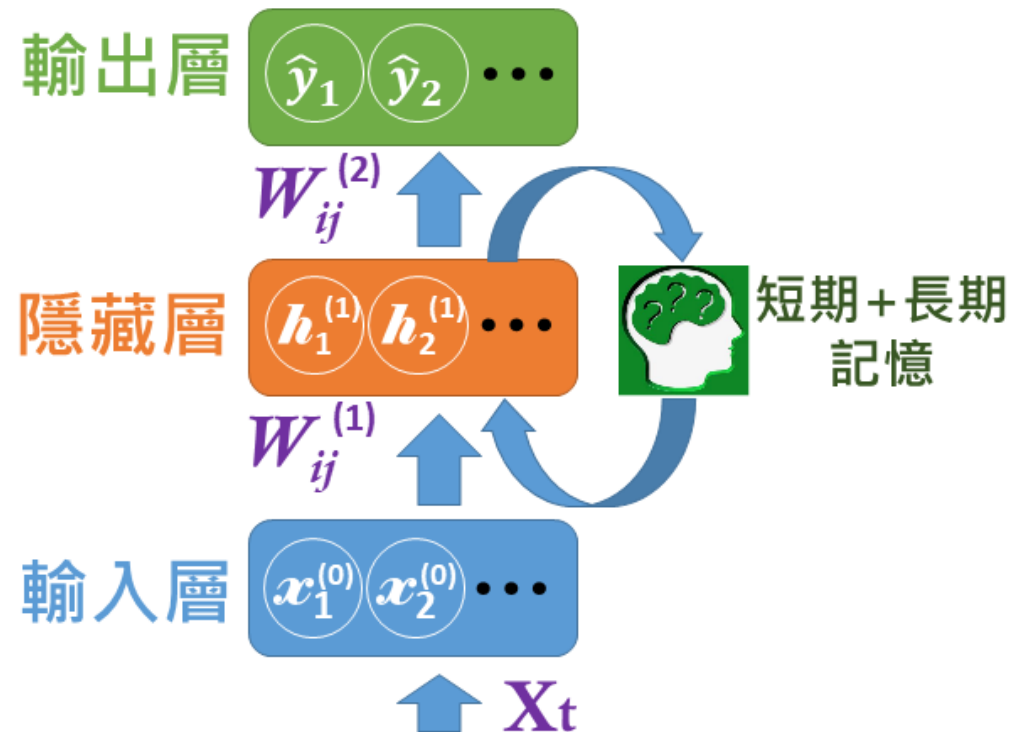


簡單RNN



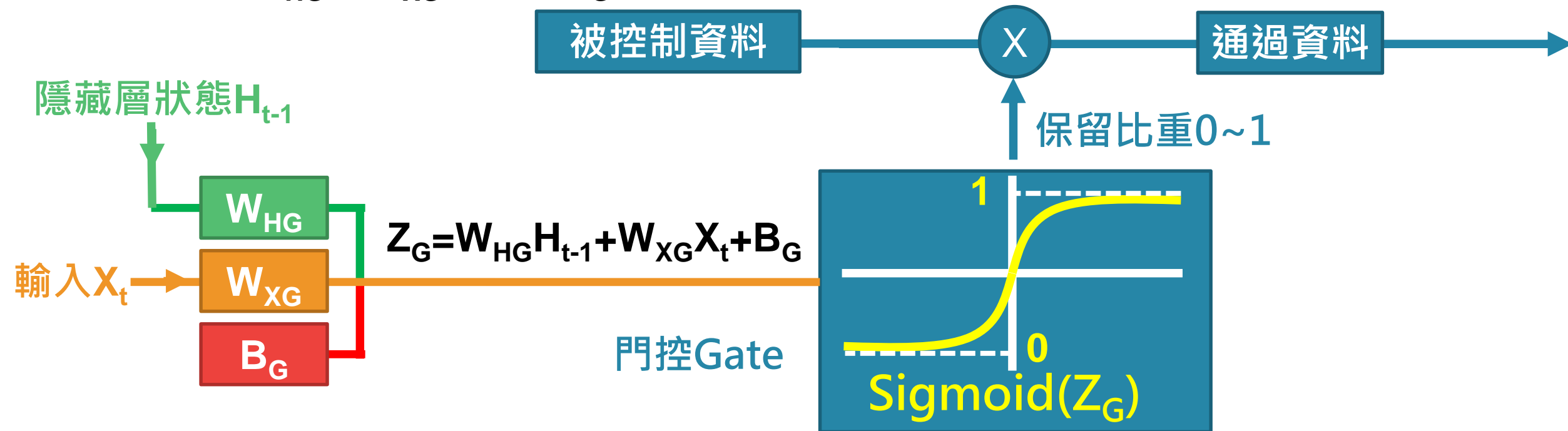
隱藏層狀態，隨時間序列推移，只能存放短期記憶。

GRU



隱藏層狀態，同時存放短期和長期記憶。利用門控機制來確定何時更新記憶，以及時重置記憶狀態。

- 門控機制：參酌要考慮的記憶資料 H_{t-1} 和當前資料 X_t 來產生一個數值，再經過sigmoid函數轉換數值為0~1之間，做為門控輸出。
- 將要控制的資料去乘以這個數值，藉以調整資料要保留(通過)的比重。
 - 若門控輸出為0，則要控制的資料會被清除。
 - 若門控輸出為1，則要控制的資料會被完全保留。
- 門控權重 W_{HG} 、 W_{XG} 和偏值 B_G 在訓練過程中透過優化法做調整。



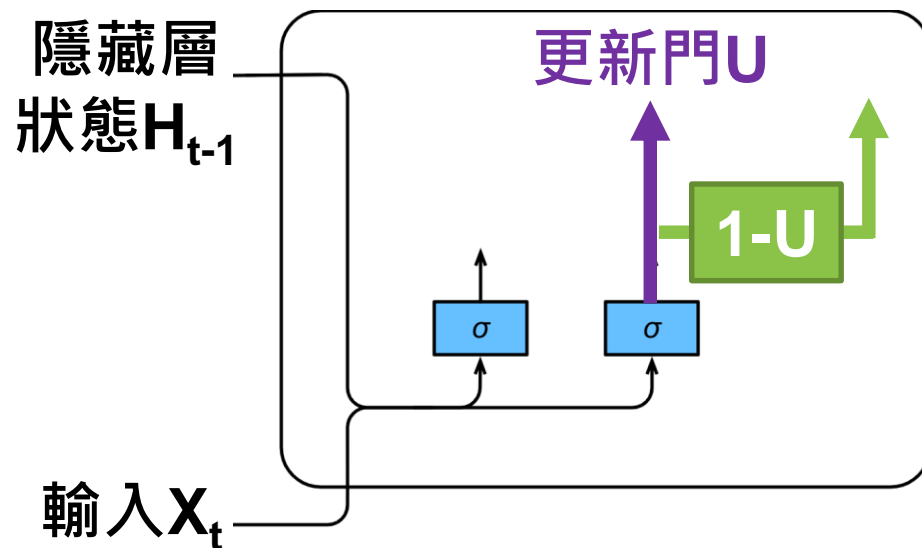
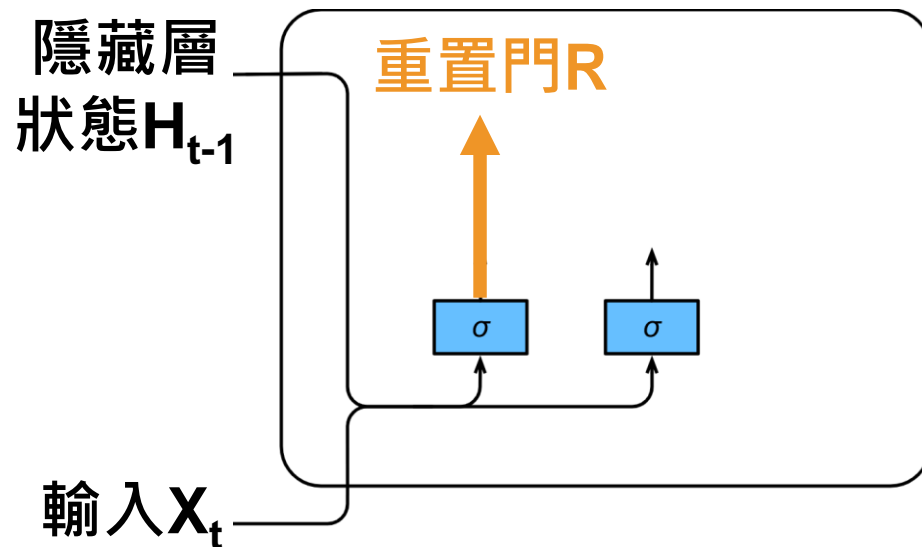
- 重置門(Reset Gate)：控制過去隱藏層狀態 H_{t-1} 要考慮多少比重用來計算隱藏層暫態 \tilde{H}_t ，即舊記憶的影響程度。

$$R = \text{Sigmoid}(W_{HR}H_{t-1} + W_{XR}X_t + B_R)$$

- 更新門(Update Gate)：控制過去隱藏層狀態 H_{t-1} 要考慮多少比重用來計算新隱藏層 H_t 輸出，即舊記憶的保留程度。

$$U = \text{Sigmoid}(W_{HU}H_{t-1} + W_{XU}X_t + B_U)$$

- 更新門的剩餘值(1-U)：用來控制隱藏層暫態 \tilde{H}_t 要考慮多少比重用來計算新隱藏層 H_t 輸出，即要記住多少新記憶。

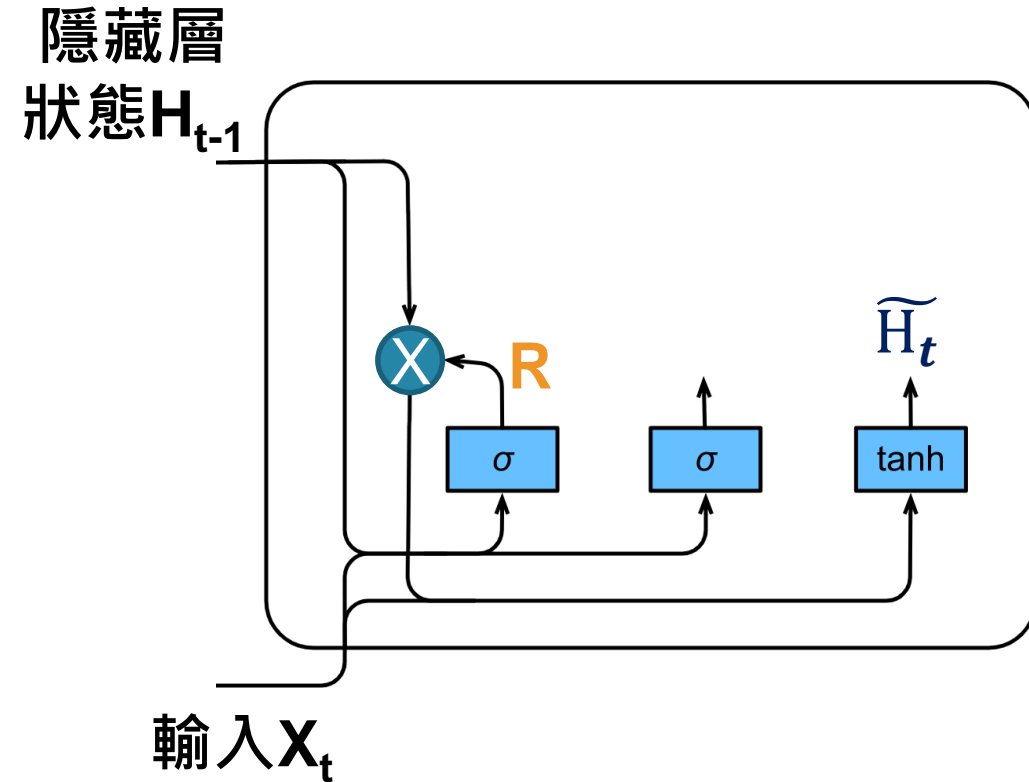


- 隱藏層暫態 \widetilde{H}_t ，同時考慮當下事件資料 X_t 及過去隱藏層狀態 H_{t-1} 。

$$\widetilde{H}_t = \tanh(X_t W_{XH} + (R \otimes H_{t-1}) W_{HH} + B_H)$$

- 過去隱藏層狀態 H_{t-1} 經過重置門 R 控制多少比重參與。

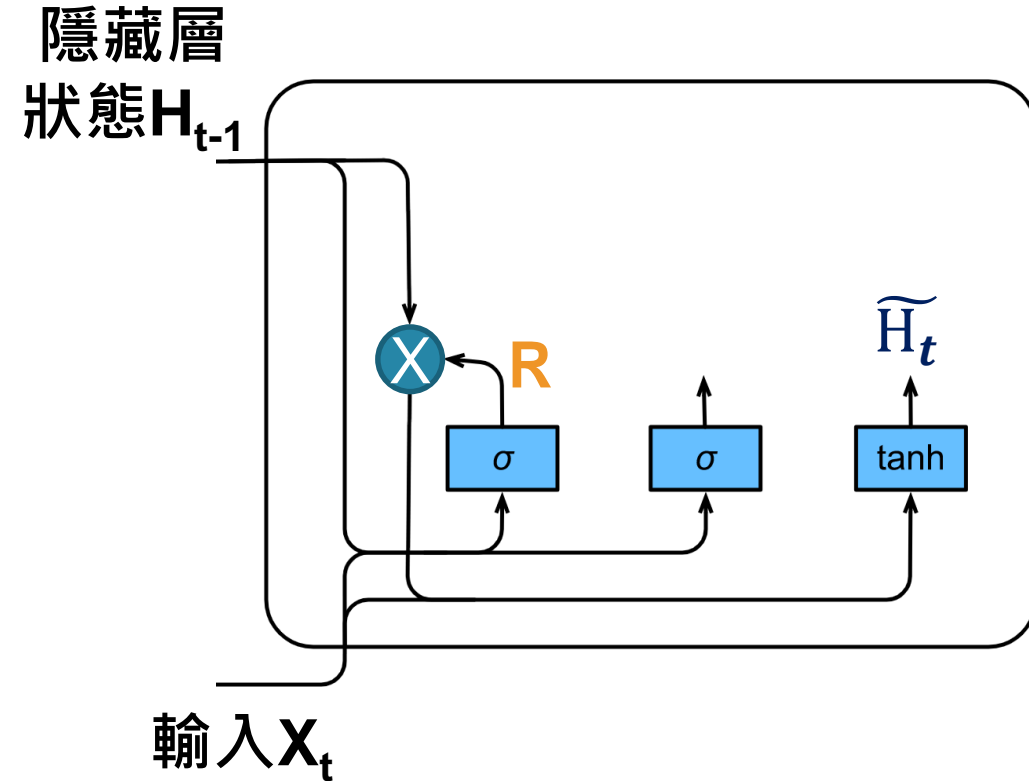
- 若 $R=1$ ：完整保留記憶狀態，即為簡單RNN的運作方式。
- 若 $R=0$ ：記憶狀態完全歸零，即為一般MLP的運作方式。



- 隱藏層暫態 \widetilde{H}_t ，同時考慮當下事件資料 X_t 及過去隱藏層狀態 H_{t-1} 。

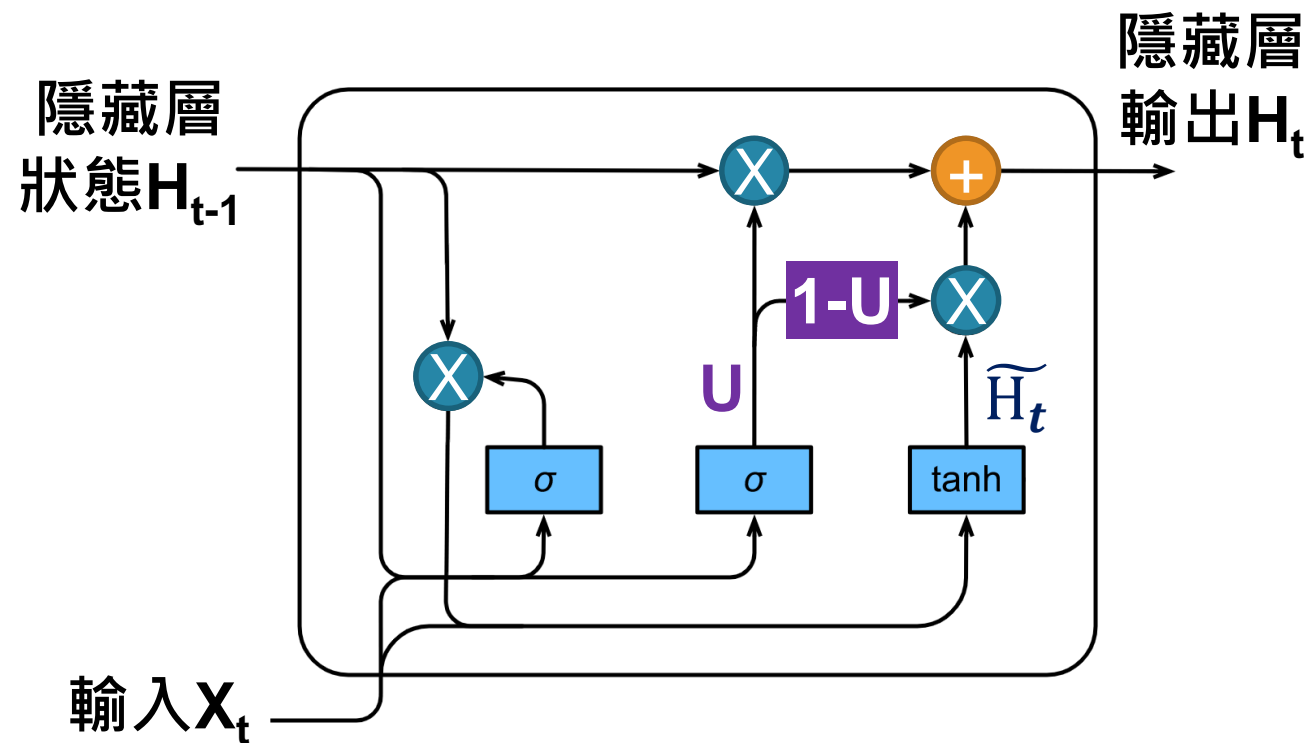
$$\widetilde{H}_t = \tanh(X_t W_{XH} + (R \otimes H_{t-1}) W_{HH} + B_H)$$

- 過去隱藏層狀態 H_{t-1} 經過重置門 R 控制多少比重參與。
 - 若 $R=1$ ：完整保留記憶狀態，即為簡單RNN的運作方式。
 - 若 $R=0$ ：記憶狀態完全歸零，即為一般MLP的運作方式。

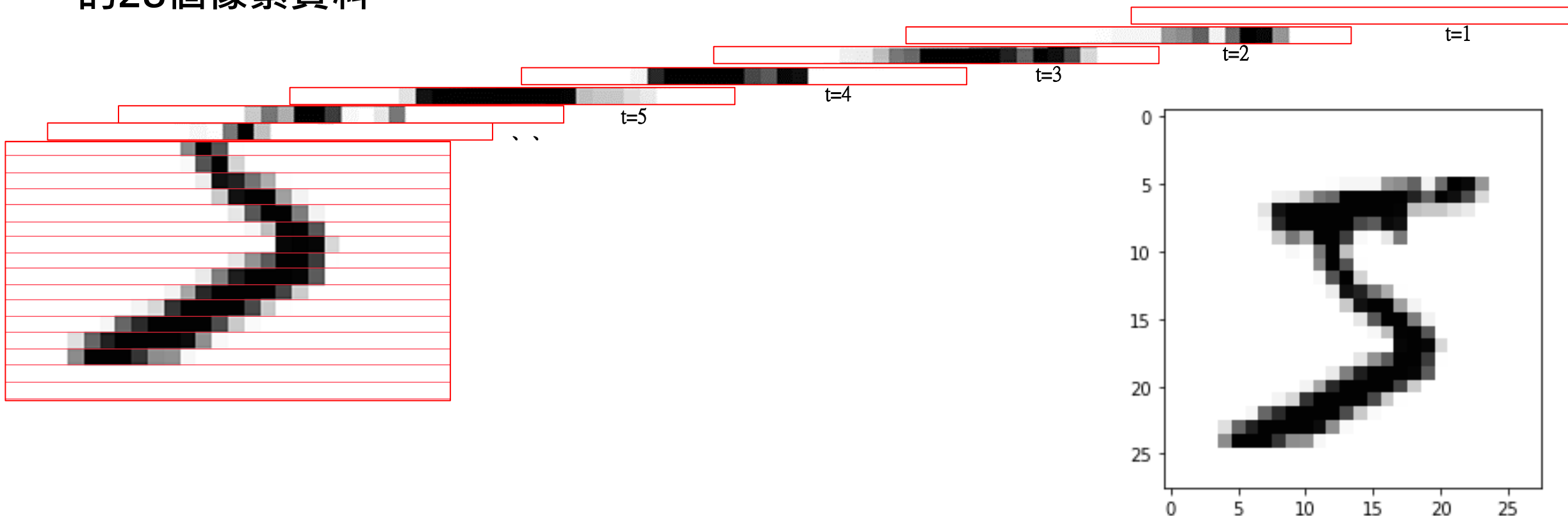


- 隱藏層輸出狀態 H_t ，同時考慮隱藏層暫態 \widetilde{H}_t 及過去隱藏層狀態 H_{t-1} 。

$$H_t = U \otimes H_{t-1} + (1-U) \otimes \widetilde{H}_t$$
- 過去隱藏層狀態 H_{t-1} 經過更新門 U 控制多少比重參與；隱藏層暫態 \widetilde{H}_t 經過 $1-U$ 控制多少比重參與。
 - 若 $U=1$ ：只保留記憶狀態，忽略當下的事件訊息 X_t ，不會加入記憶中。
 - 若 $U=0$ ：將記憶狀態完全清除，用目前的隱藏層暫態取代。

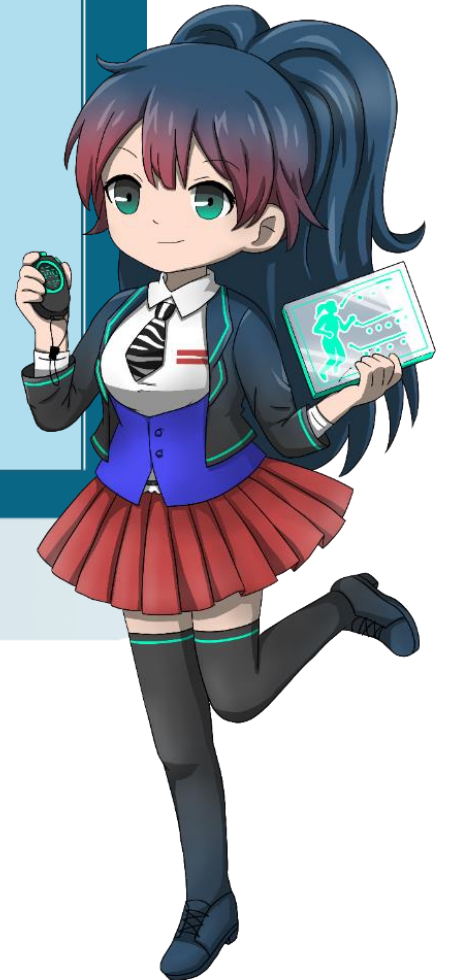


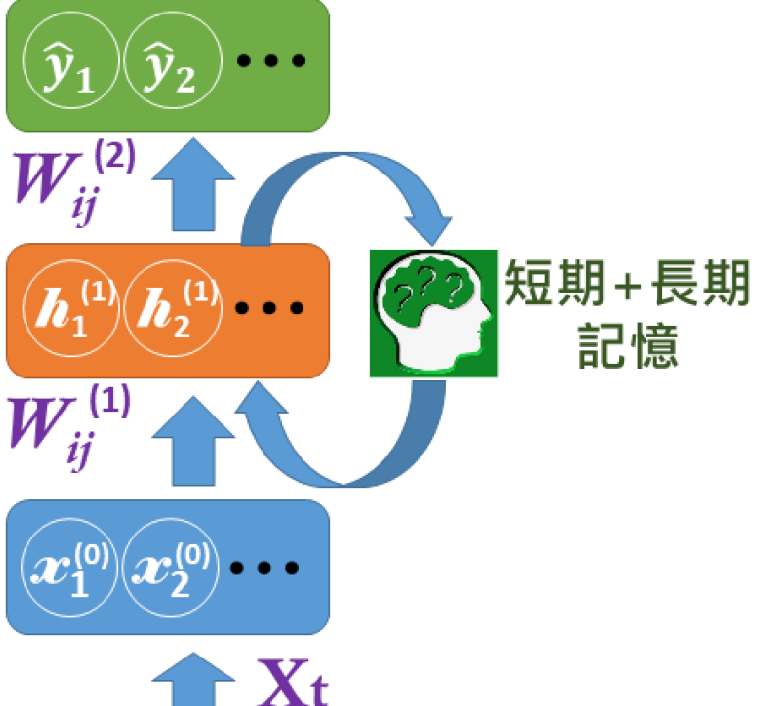
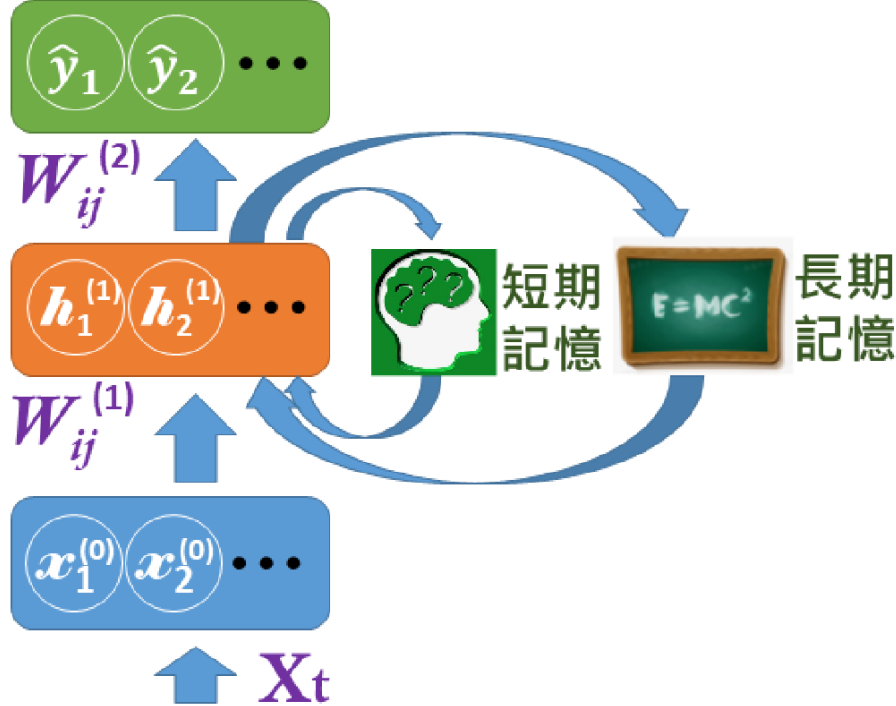
- 圖片可視為有空間序列關係的資料，故亦可以透過RNN來訓練模型，來做圖片的辨識。
- 如下圖將Mnist圖片，由上而下切成 $t=1 \sim t=28$ 個序列資料，每一個序列為一整列的28個像素資料。



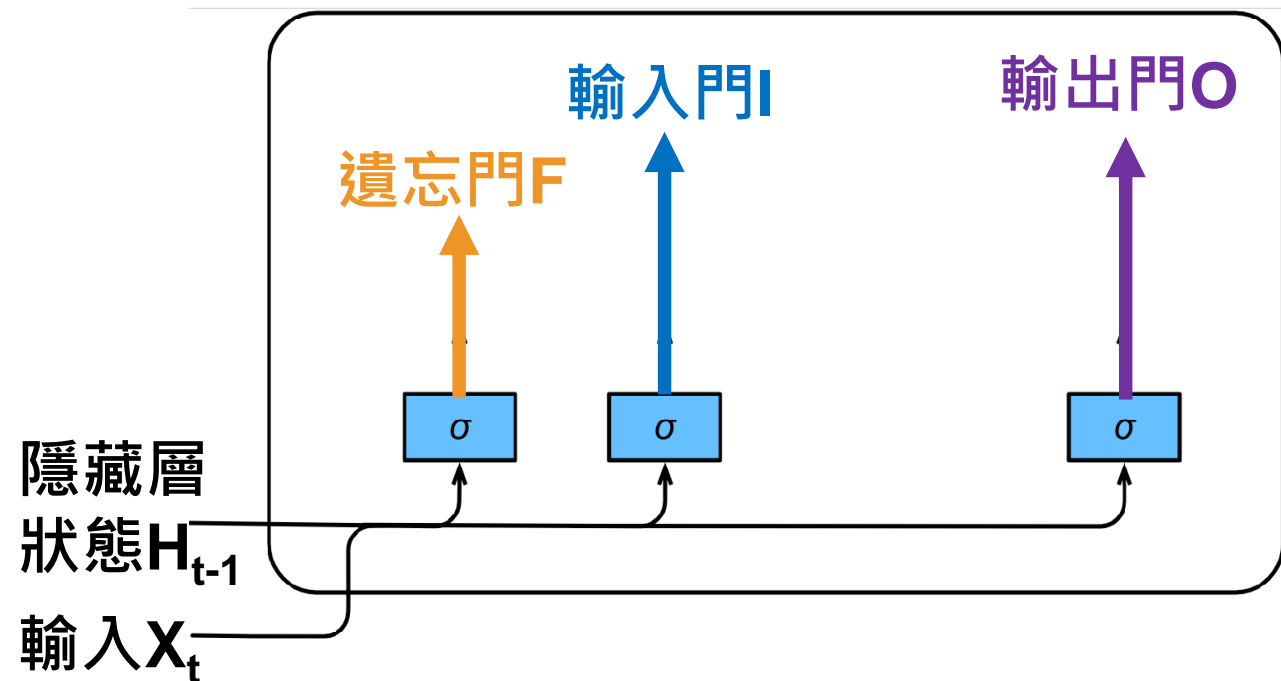
長短期記憶網路 (LSTM)

單元3



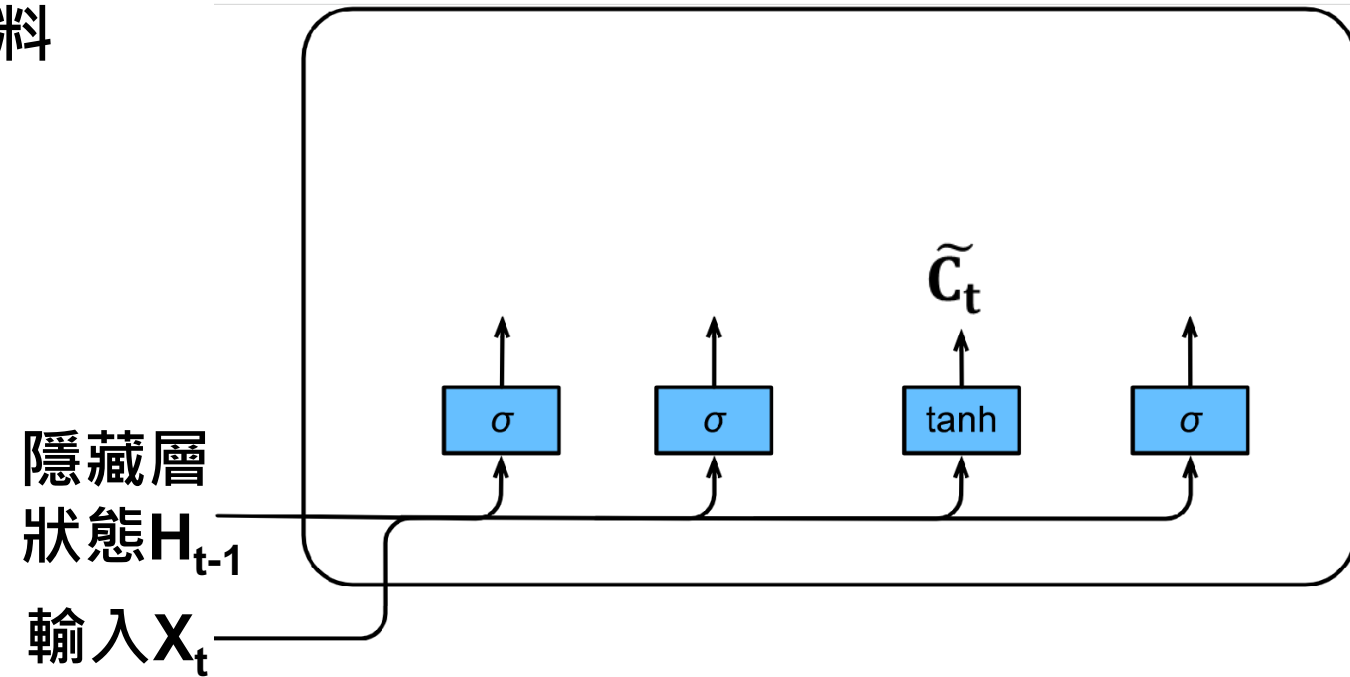
GRU	LSTM
<p data-bbox="122 278 675 392">輸出層 $\hat{y}_1 \hat{y}_2 \dots$</p> <p data-bbox="122 521 1090 635">隱藏層 $h_1^{(1)} h_2^{(1)} \dots$ 短期+長期記憶</p> <p data-bbox="122 763 675 878">輸入層 $x_1^{(0)} x_2^{(0)} \dots$</p> <p data-bbox="484 906 675 963">$\uparrow X_t$</p>  <p>The diagram illustrates the GRU (Gated Recurrent Unit) architecture. It consists of three main layers: an input layer at the bottom, a hidden layer in the middle, and an output layer at the top. The input layer receives a sequence of inputs $x_1^{(0)}, x_2^{(0)}, \dots$ from X_t. These inputs are processed by the hidden layer, which contains hidden states $h_1^{(1)}, h_2^{(1)}, \dots$. The hidden layer is represented by a single green box with a brain icon, labeled '短期+長期記憶' (Short-term + Long-term memory). The hidden layer is connected to the output layer, which produces a sequence of outputs $\hat{y}_1, \hat{y}_2, \dots$. Weights $W_{ij}^{(1)}$ and $W_{ij}^{(2)}$ are shown connecting the input and hidden layers, and the hidden and output layers, respectively. Arrows indicate the flow of information between these components.</p>	<p data-bbox="1251 278 1803 392">輸出層 $\hat{y}_1 \hat{y}_2 \dots$</p> <p data-bbox="1251 521 2356 635">隱藏層 $h_1^{(1)} h_2^{(1)} \dots$ 短期記憶 長期記憶</p> <p data-bbox="1251 763 1803 878">輸入層 $x_1^{(0)} x_2^{(0)} \dots$</p> <p data-bbox="1617 906 1803 963">$\uparrow X_t$</p>  <p>The diagram illustrates the LSTM (Long Short-Term Memory) architecture. It consists of three main layers: an input layer at the bottom, a hidden layer in the middle, and an output layer at the top. The input layer receives a sequence of inputs $x_1^{(0)}, x_2^{(0)}, \dots$ from X_t. These inputs are processed by the hidden layer, which contains hidden states $h_1^{(1)}, h_2^{(1)}, \dots$. The hidden layer is represented by two separate components: a green box with a brain icon labeled '短期記憶' (Short-term memory) and a green box with a chalkboard icon labeled '長期記憶' (Long-term memory). The hidden layer is connected to the output layer, which produces a sequence of outputs $\hat{y}_1, \hat{y}_2, \dots$. Weights $W_{ij}^{(1)}$ and $W_{ij}^{(2)}$ are shown connecting the input and hidden layers, and the hidden and output layers, respectively. Arrows indicate the flow of information between these components.</p>
<p data-bbox="84 999 1205 1213">隱藏層狀態，同時存放短期和長期記憶。利用2個門控機制來確定何時更新記憶，以及時重置記憶狀態。</p>	<p data-bbox="1245 999 2448 1285">隱藏層狀態只存放短期記憶。另有一個記憶元Cell負責存放長期記憶。利用3個門控機制來確定何時更新記憶，以及時重置記憶狀態。</p>

- 遺忘門(Forget Gate)：用來重置記憶單元的內容，控制記憶單元 C_{t-1} 要保留多少比重。
$$F = \text{Sigmoid}(H_{t-1}W_{HF} + X_tW_{XF} + B_F)$$
- 輸入門(Input Gate)：用來控制記憶單元暫態 \tilde{C}_t 要考慮多少比重用來計算新記憶單元 C_t ，即要加入多少新記憶。
$$I = \text{Sigmoid}(H_{t-1}W_{HI} + X_tW_{XI} + B_I)$$
- 輸出門(Output Gate)：用來決定將資料輸出到隱藏層狀態 H_t 的比重。
$$O = \text{Sigmoid}(H_{t-1}W_{HO} + X_tW_{XO} + B_O)$$



- 記憶單元暫態 \tilde{C}_t ，同時考慮當下事件資料 X_t 及過去隱藏層狀態 H_{t-1} 。

$$\tilde{C}_t = \tanh(X_t W_{XC} + H_{t-1} W_{HC} + B_C)$$



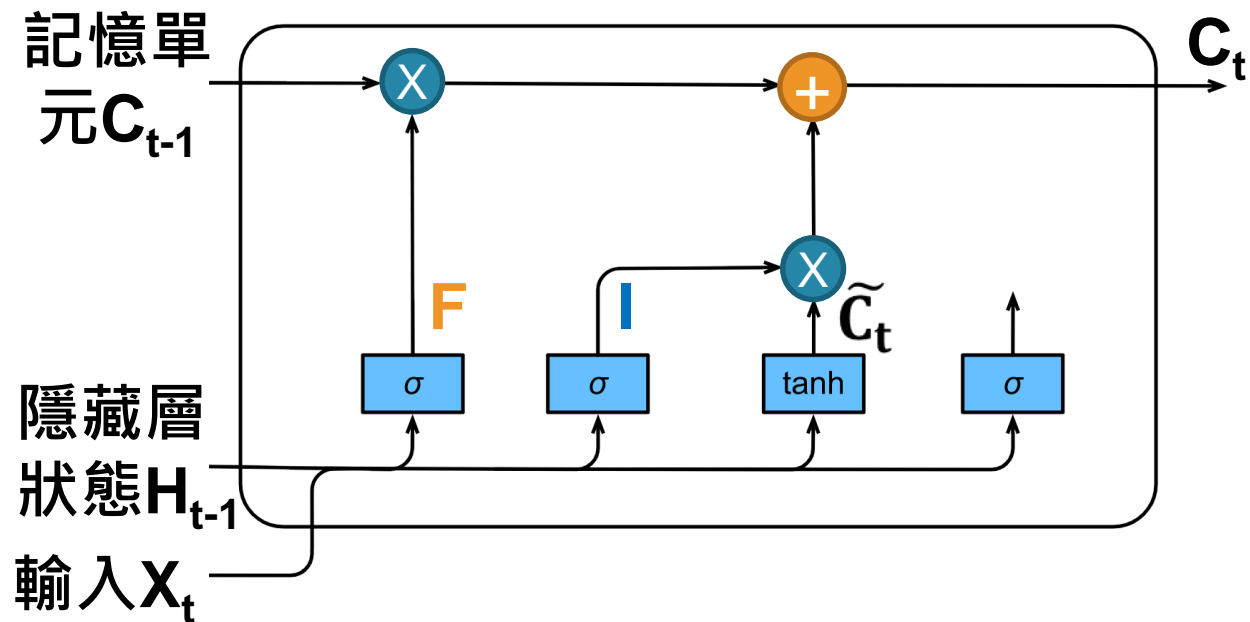
- 記憶單元的輸出狀態 C_t ，同時考慮記憶單元暫態 \tilde{C}_t 及過去過去記憶單元 C_{t-1} 的保留狀態。

$$C_t = F \otimes C_{t-1} + I \otimes \tilde{C}_t$$

- 過去記憶單元 C_{t-1} 經過遺忘門 F 控制多少比重參與；記憶單元暫態 \tilde{C}_t 經過輸入門控制多少比重參與。

□若 $F=1$ 、 $I=0$ ：只保留記憶狀態，忽略當下的事件訊息 X_t ，不會加入記憶中。

□若 $F=0$ 、 $I=1$ ：將記憶狀態完全清除，用目前的記憶單元暫態取代。



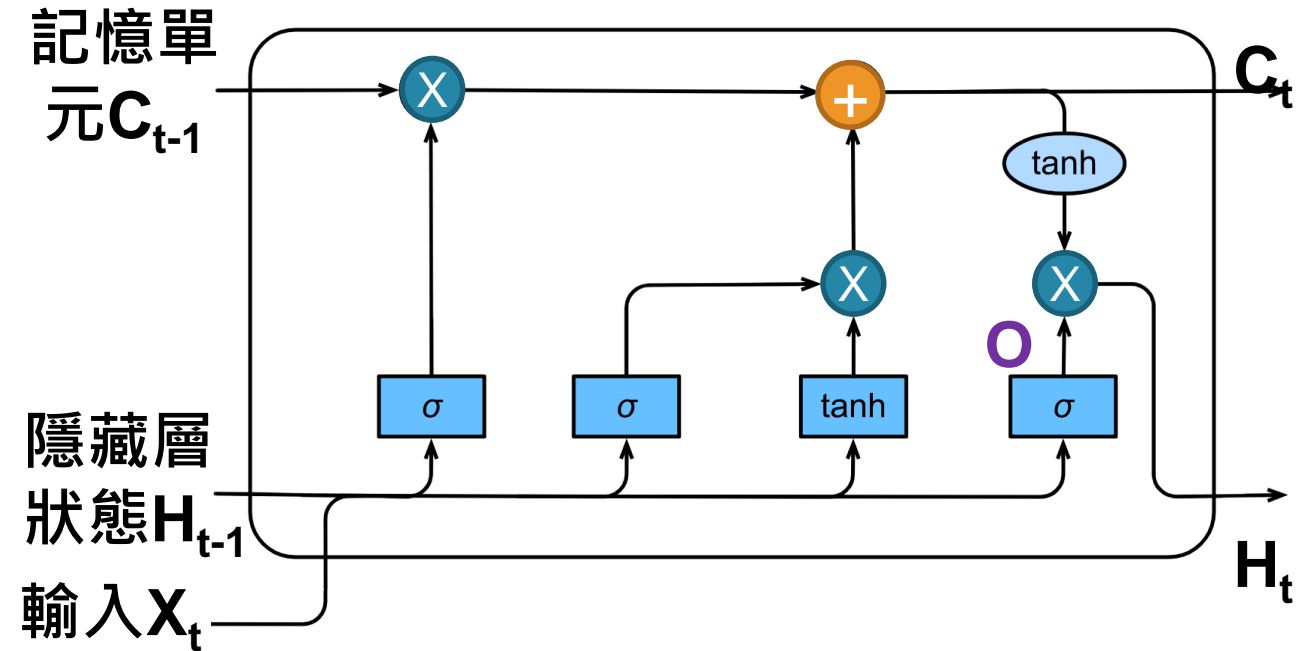
- 記憶單元的輸出狀態 C_t ，同時考慮記憶單元暫態 \tilde{C}_t 及過去過去記憶單元 C_{t-1} 的保留狀態。

$$C_t = F \otimes C_{t-1} + I \otimes \tilde{C}_t$$

- 過去記憶單元 C_{t-1} 經過遺忘門 F 控制多少比重參與；記憶單元暫態 \tilde{C}_t 經過輸入門控制多少比重參與。

□若 $F=1$ 、 $I=0$ ：只保留記憶狀態，忽略當下的事件訊息 X_t ，不會加入記憶中。

□若 $F=0$ 、 $I=1$ ：將記憶狀態完全清除，用目前的記憶單元暫態取代。



- 在資料集\股價資料目錄內，有各公司自2009年起10年內的股價資料：

欄位1：交易日期

欄位2：交易量

欄位3：開盤價

欄位4：收盤價

欄位5：盤中最高價

欄位6：盤中最低價

欄位7：還原權值股價

	A	B	C	D	E	F	G
1	date	volume	open	close	high	low	adjclose
2	2009/1/2	20238000	16.01	16.99	17.04	15.75	14.50893
3	2009/1/5	25421400	16.8	17.62	18.36	16.78	15.04693
4	2009/1/6	38858500	18.07	19.58	19.93	18	16.72071
5	2009/1/7	30848100	19.07	18.1	19.38	17.76	15.45684
6	2009/1/8	21116300	17.9	18.82	18.98	17.65	16.0717
7	2009/1/9	35802100	18.97	19.06	19.98	18.69	16.27665
8	2009/1/12	49187500	20.08	18.79	20.95	18.07	16.04608
9	2009/1/13	31384000	18.27	18.86	19.58	17.86	16.10585
10	2009/1/14	39491900	18.25	17.19	18.46	16.96	14.67973

- 請取前面2000筆交易資料作為訓練資料，其後做為測試資料。

- 搭建一個LSTM神經網路：

時序為20天

輸入層的節點為6個，即欄位2~7為特徵資料

隱藏層有512個節點

輸出層有1個節點，即預測第21的受盤價

- 以測試資料集測試訓練完成的模型，其準確度。

自然語言處理 (NLP)

單元4



- 何謂自然語言：自然演化而成的語言，基本上人類所使用的語言都算是自然語言(中文，英文，韓文，法文，手語等)。
- 自然語言的特性：
 - 人類的語言具高度曖昧性，一句話可能有多重的意思或隱喻。
 - 一段文字中不同的停頓點，停頓點不同，對文字的理解也不同。
 - 人們不斷創造新詞，語言學家越來越難建立規則(文法及語意學)來規範語文。

努力才能成功 → 努力 才 能 成功

他的領導才能很突出 → 他 的 領 導 才 能 很 突 出

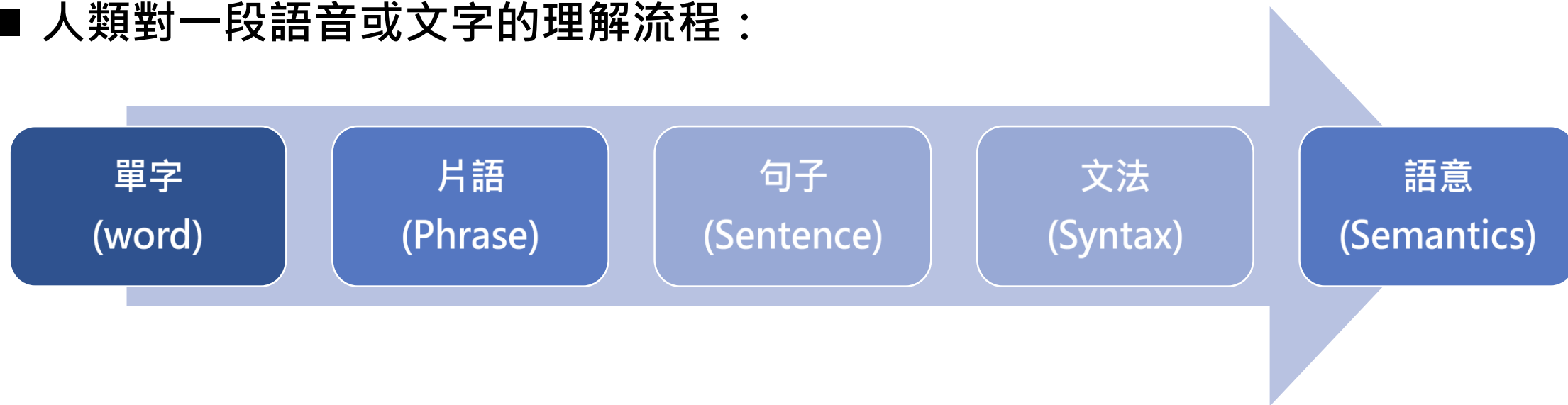
- 自然語言的處理 (NLP Natural Language Processing)：專注在如何讓計算機處理並分析大量（人類的）自然語言數據。

- NLP 常見的研究主題：

- ☐ 語音辨識
- ☐ 自然語言理解
- ☐ 機器翻譯
- ☐ 自然語言的生成

發展自然語言處理的兩個方向

■ 人類對一段語音或文字的理解流程：



規則基準 (Rule-based Method)

遵循上面的流程來了解文字或語音的意義，除了必須為每個階段建立龐大的資料庫，各階段間的交互規則更是複雜，可能是天文數字。

機器學習 (Machine Learning)

設計神經網路，改為提供大量字句當作輸入，讓機器自我學習，產生適當的回應。



語言模型的分類：

- 統計語言模型：統計的方法去解決語言模型的問題（條件概率）。
- 神經網路語言模型：用神經網路的方法去解決語言模型的問題。

- 訓練過程：從大量的文本資料中去統計每一個單詞出現下個單詞的機率。
例如： $P(\text{天氣}|\text{今天})$ 、 $P(\text{氣溫}|\text{今天})$ 、 $P(\text{新聞}|\text{今天})$ 、 \dots
 $P(\text{很好}|\text{天氣})$ 、 $P(\text{晴朗}|\text{天氣})$ 、 $P(\text{涼爽}|\text{天氣})$ 、 \dots
- 推測過程：依據句子出現單詞的順序，利用條件機率的鍊式法則，求出這個句子的機率。
 $P(\text{今天天氣晴朗，想去野餐})$
 $= P(\text{天氣}|\text{今天}) \times P(\text{晴朗}|\text{天氣}) \times P(\text{想}|\text{晴朗}) \times P(\text{去野餐}|\text{想})$
- 如果句子很長，前面的條件機率很小，我們可以指定只計算前N 個單詞的條件機率，稱為N元語言模型(N-gram)。

語言表示的種類

詞袋模型 BOW (Bag of Word)

將句子分詞後，裝進一個袋子裡，並給唯一的識別編碼。

不考慮其詞法與語序，即每個詞語都是獨立的。

不須經過訓練

常見的有one-hot、TF-IDF、Huffman編碼。



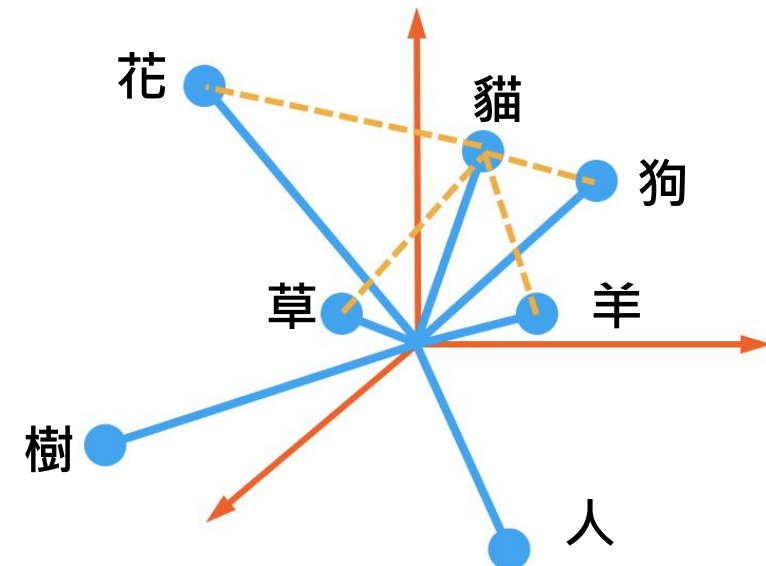
詞向量模型(Word Embedding)

將每一個編碼後的詞語映射到高維度的向量當中。

考慮詞語位置關係，通過求餘弦的方式，可以判斷兩個詞語之間的關係。

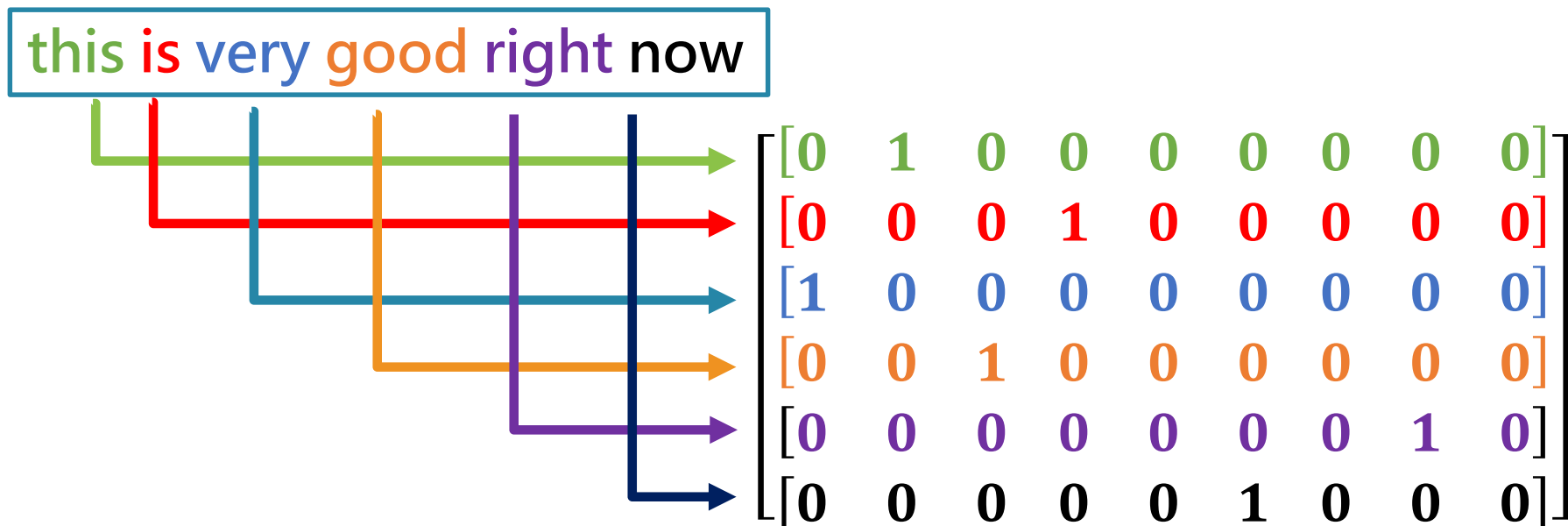
需經過神經網路模型的訓練

基於CBOW和Skip-Gram演算法的神經網路模型。



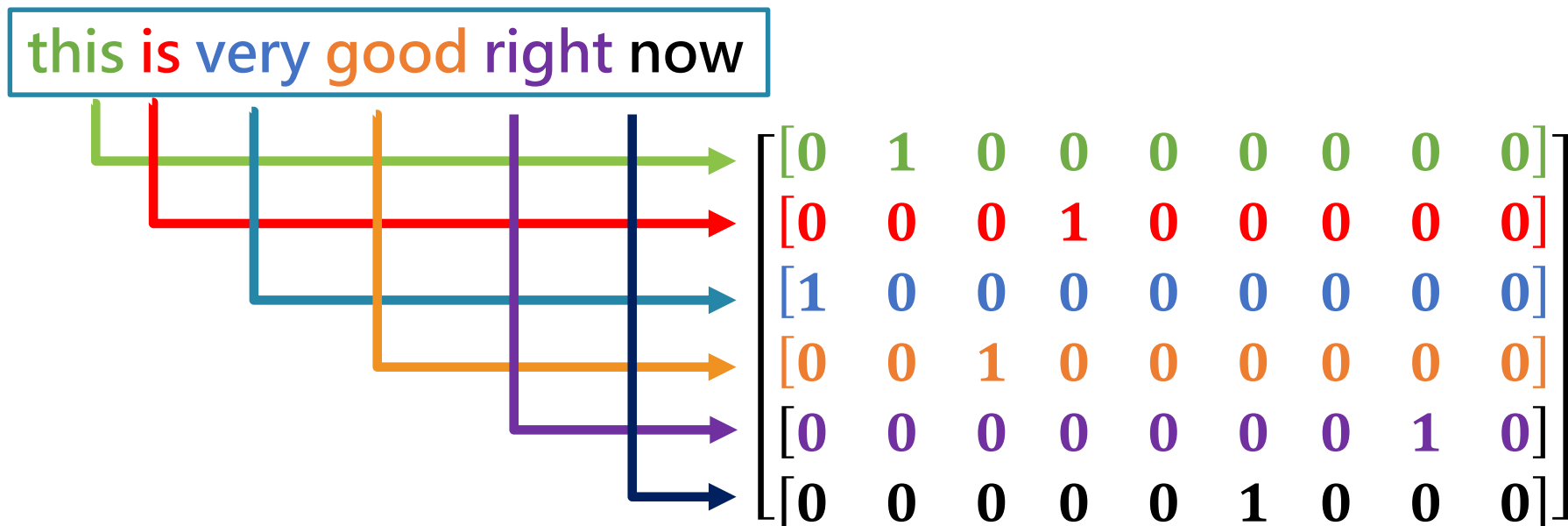
One-Hot 表示字、詞的步驟

- 將文本中的文字敘述(句子)，取出所有的單字(單詞)，建立一個沒有重複單字(單詞)的詞庫。
- 資料庫中的單字可以用one hot encode來表示，即每個單字都是多維向量中的獨立向量。
- 依序將句子中的單字化為one hot encode的向量，組成一個有序不定長度的序列，稱為句向量。



One Hot 表示字、詞的問題

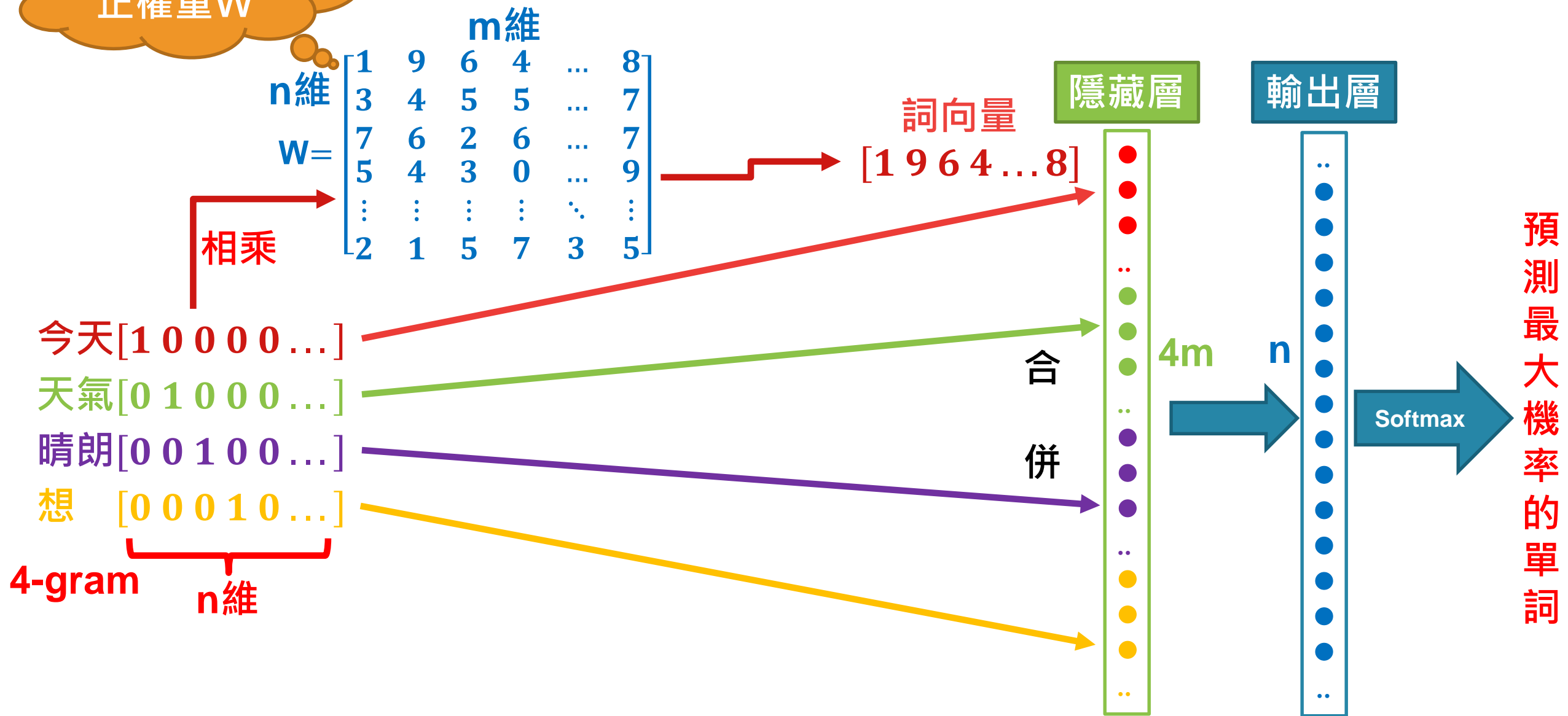
- 維度爆炸問題：若詞庫內的單詞數量有10000個，One-Hot編碼的維度將達10000個維度。
- 稀疏矩陣：One-Hot編碼中，大量維度都是0、所含有效訊息很少，會很難訓練。
- 向量垂直：每一個單詞都只佔有一個維度，彼此間毫無相干，相互投影(餘弦相關度)都是零。



- Feeling.csv資料檔含有78筆描述心情的英文句子，並標示0(悲觀)、1(樂觀)二元分類。
- 請用one hot encode 將詞庫內的單字(詞)做編碼，組成句向量，取最長句子長度為時序，不足的以0補足長度。
- 搭建一個RNN神經網路：
 - 時序為最長句子的長度
 - 輸入層的節點由單字one hot encode的維度來決定
 - 隱藏層有64個節點
 - 輸出層有1個節點。
- 以測試資料集測試訓練完成的模型，其準確度。
- 自己輸入表達情緒樂觀和悲觀的語句，讓訓練好的模型做判斷，看看是否判斷正確。

反向傳播修正權重W

神經網路語言模型簡介



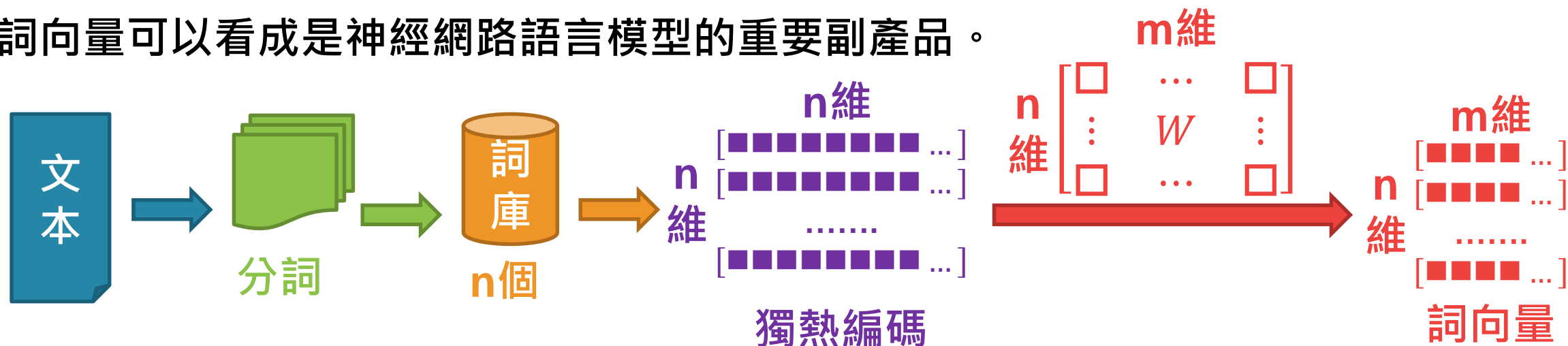
單元5

詞向量 Word Embedding



詞向量(Word Embedding)

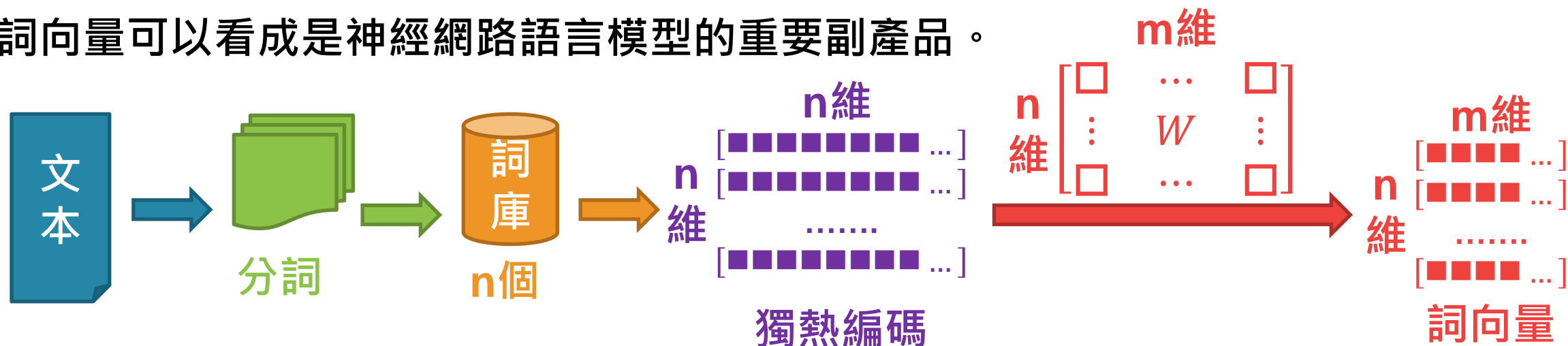
- 詞向量可以看成是神經網路語言模型的重要副產品。



- 可以彈性調整神經網路語言模型 W 權重的行數(m 維)，來控制詞向量的維度。
- 詞向量的維度越高，其所能提供的資訊也就越多，計算結果的可靠性就更值得信賴。
- 詞性和語意相近的單詞，經過訓練後的詞向量相似度會比較高。
- Word2vector**模型是由Google在2013年提出的，把字詞換成向量表達的一個預訓練模型。

詞向量(Word Embedding)

- 詞向量可以看成是神經網路語言模型的重要副產品。

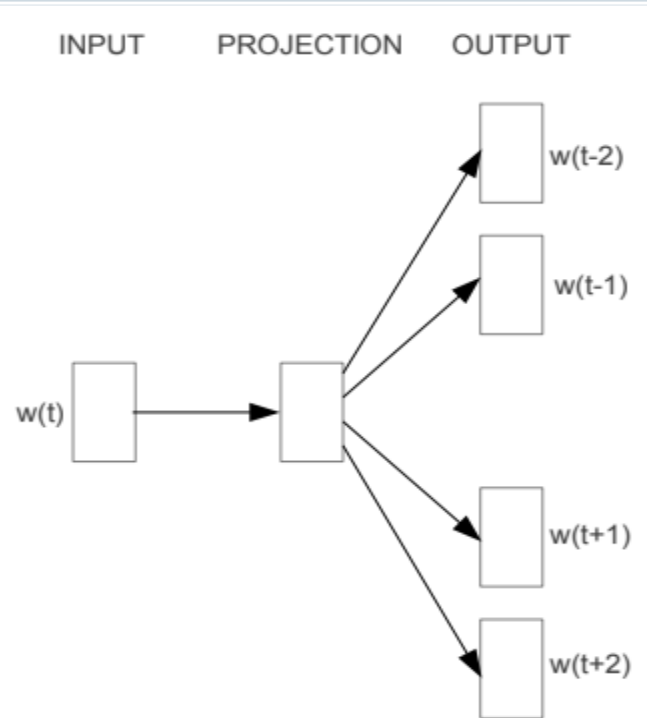


- 可以彈性調整神經網路語言模型 W 權重的行數(m 維)，來控制詞向量的維度。
- 詞向量的維度越高，其所能提供的資訊也就越多，計算結果的可靠性就更值得信賴。
- 詞性和語意相近的單詞，經過訓練後的詞向量相似度會比較高。
- Word2vector**模型是由Google在2013年提出的，把字詞換成向量表達的一個預訓練模型。

Word2vector兩種訓練模型

Skip-gram

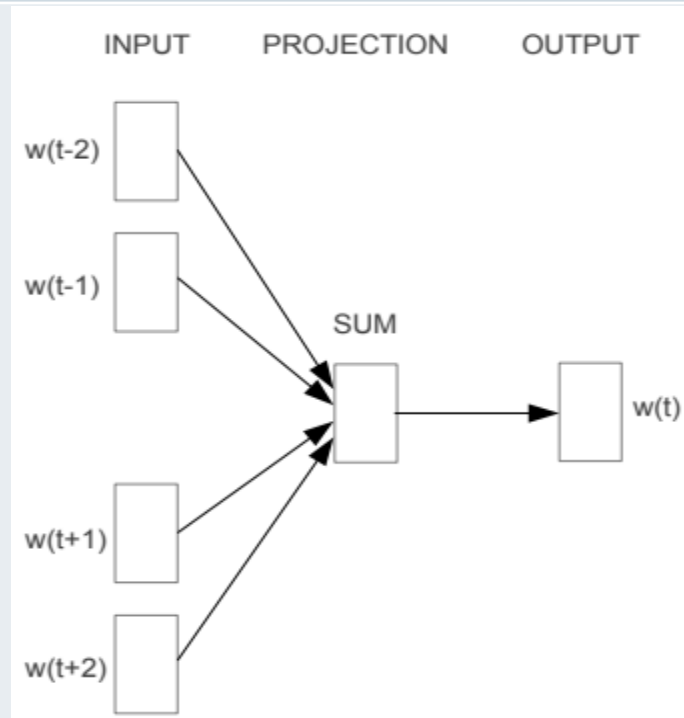
用中間詞作為輸入，來預測上下文



預測的詞較少，花費的時間較短

CBOW

用上下文作為輸入，來預測中間詞



預測的詞較多，但花費的時間長

- [gensim](#) 是使用 google 釋出的 word2vec 模型的套件，可找到字的向量、相似字，計算向量之間的相似度。
- 語法：`gensim.models.word2vec.Word2Vec(參數)`
- 參數說明：
 - `sentences`：這是要訓練的句子集合
 - `size`：這是訓練出的詞向量會有幾維
 - `alpha`：機器學習中的學習率，這東西會逐漸收斂到 `min_alpha`
 - `sg`：`sg=1`表示採用skip-gram，`sg=0` 表示採用cbow
 - `window`：能往左往右看幾個字的意思
 - `workers`：執行緒數目
 - `min_count`：若這個詞出現的次數小於`min_count`，那他就不會被視為訓練對象

- 如何把單字或詞轉換為變數，輸入神經網路。
- 字句有長有短，即input 變數數目不等，回應也是一樣的狀況，output 變數數目也不等。
- 語文會有上下文的關係，斷章取義常會扭曲全文代表的意義。
- 文字前後間有很強的連結關係，故神經網路必須有記憶力機制，能參酌之前的判斷，預測準確率就會高很多。
- 『遞迴神經網路』演算法，它是『自然語言處理』領域最常使用的神經網路模型，能額外考慮上下文的關係，準確率就會顯著提高。
例：如果餐館的服務生要從顧客的聊天中，聽到她喜歡吃辣，就可以預測顧客晚餐會點麻婆豆腐的機率比較高。