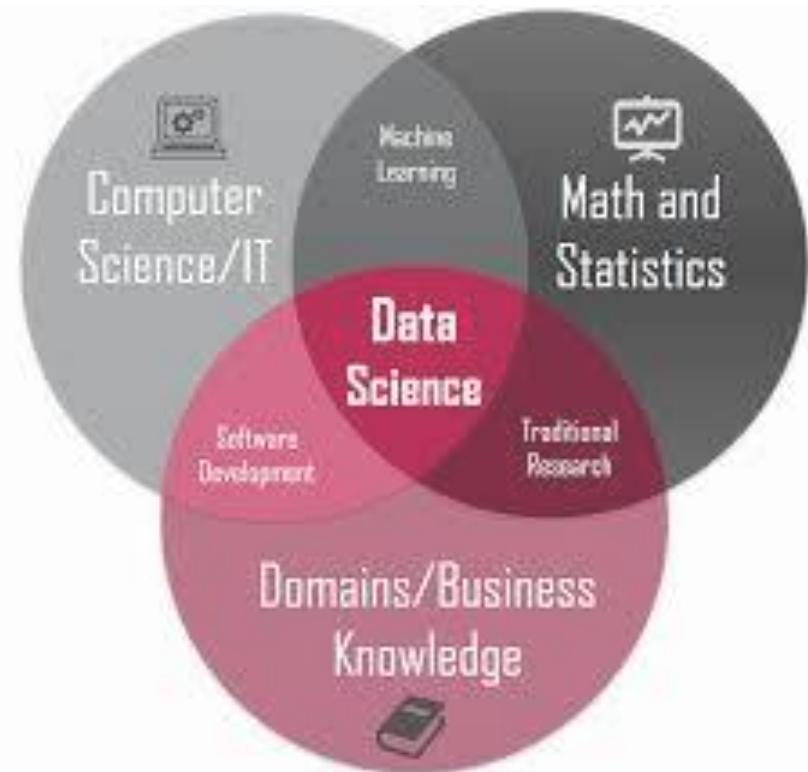


AI & DS



U12-深度學習_循環神經網路RNN

2023.11_V1.1

Data
Science

Artificial
Intelligence

Machine
Learning

Deep
Learning

Statistics

單元大綱

- 認識循環神經網路(RNN)
- 建立與訓練RNN相關語法

Part 1

認識循環神經網路 (RNN)



循環神經網路 (RNN, RECURRENT NEURAL NETWORK)

- RNN 是一種功能強大且穩健的神經網路，是唯一具有內部存儲器的算法。最初是在 1980 年代創建的，1990 年代因**長短期記憶 (LSTM)** 的發明，讓 RNN 成為主流。
- RNN主要優勢為架構內具有記憶神經元，可用於處理具有時間序列特性的數據，可以記住有關他們收到的輸入的重要訊息，這使RNN能夠非常精確地預測接下來會發生什麼。這就是為什麼RNN是時序數據、語音、文案、財務數據、聲音、影片、SEO優化、天氣等序列數據的首選算法。與其他算法相比，循環神經網路可以對序列及其上下文形成更深入的理解。RNN也成為自然語言處理(NLP)最常使用的模型。

- DNN 的缺點:

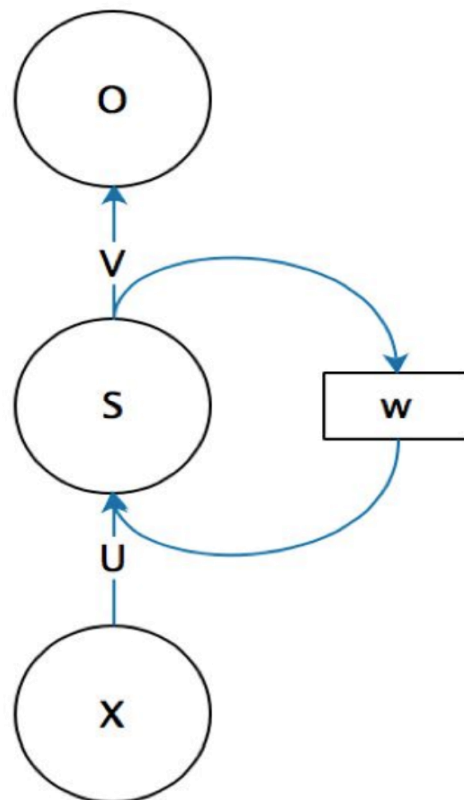
- 1.資料沒有順序性:

- 人類的語句有順序性,例如:

- 「我愛貓」與「貓愛我」的意義不同。

- 2.輸入與輸出資料為固定長度

- 由於在處理語文翻譯時,所翻譯輸入的文字長度與翻譯後輸出的文字長度不固定,因此,DNN無法處理這類問題。



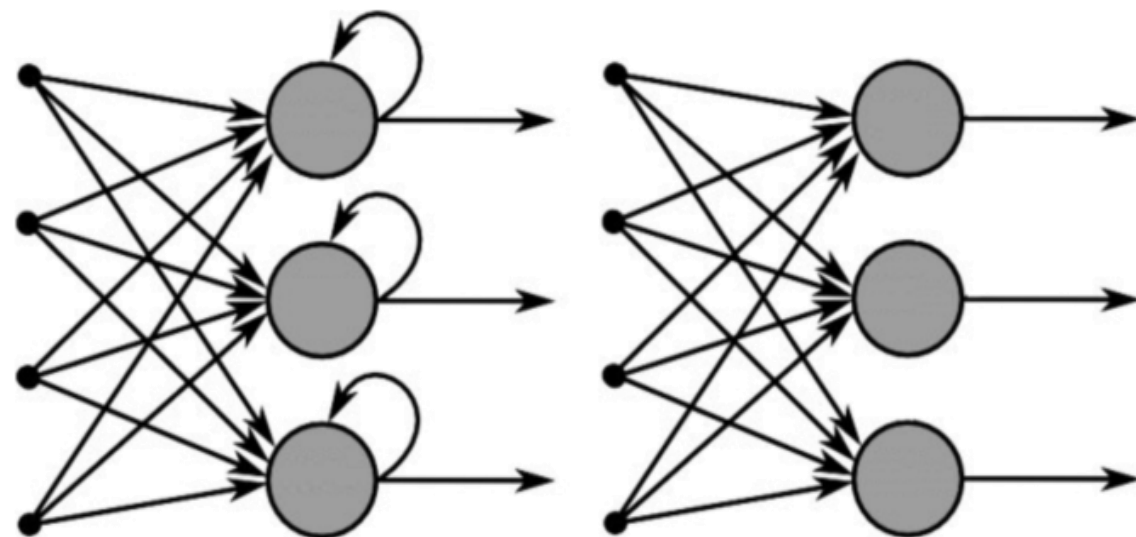
x : 輸入層
o : 輸出層
s : 隱藏層，s不僅取決於當前這次的輸入x，還取決於上一次隱藏層的值
U : 是輸入層到隱藏層的權重矩陣
V : 是隱藏層到輸出層的權重矩陣
W : 是隱藏層上一次的值作為這一次的輸入的權重矩陣(循環層)

[文章引用] <https://pse.is/4ug6j6>

[圖片引用] <https://pse.is/4ugbgx>

RNN 和前饋神經網絡的差異

- 前饋神經網絡(Feed-Forward Neural Network)中，訊息僅沿一個方向移動——從輸入層，通過隱藏層，到輸出層。訊息直接通過網絡移動，並且不會兩次觸及節點。
- 前饋神經網絡對它們接收的輸入沒有記憶，並且不善於預測接下來會發生什麼。因為前饋網絡只考慮當前輸入，它沒有時間順序的概念。除了訓練之外，它根本不記得過去發生的任何事情。
- 在 RNN 中，訊息通過一個循環。當它做出決定時，它會考慮當前的輸入以及它從之前收到的輸入中學到的東西。
- 普通的前饋神經網絡，並給“神經元”這個詞作為輸入，它會逐個字符地處理這個詞。當它到達字符“r”時，它已經忘記了“n”、“e”和“u”，這使得這種類型的神經網絡幾乎不可能預測接下來會出現哪個字符。RNN能夠記住這些字符是因為它的內部記憶。它產生輸出，複製該輸出並將其循環回網絡。

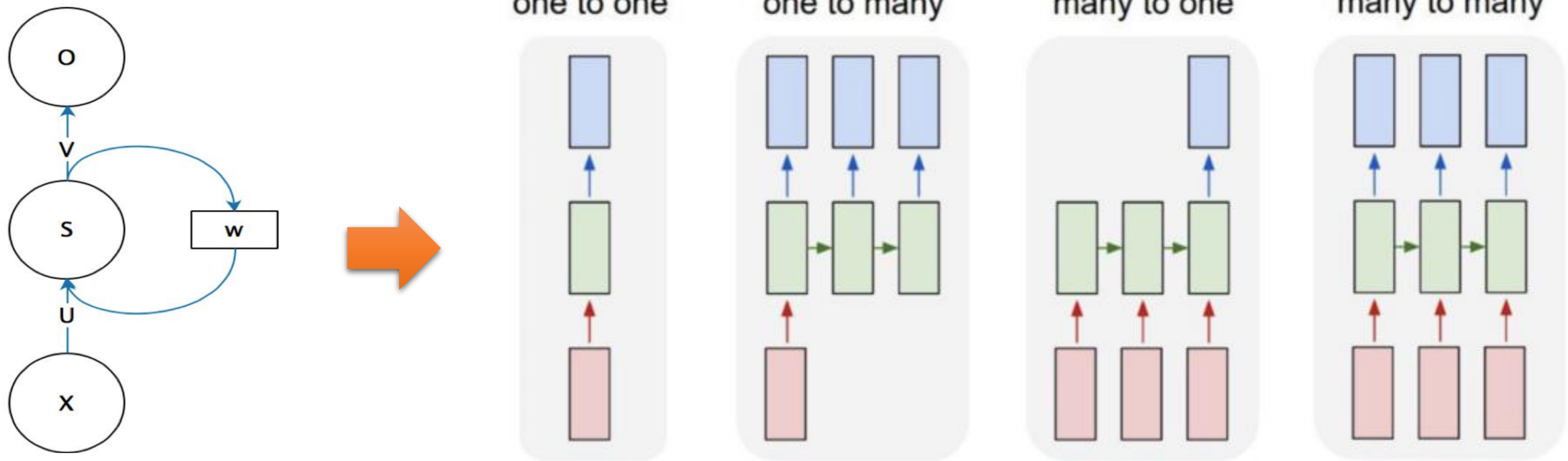


Recurrent Neural Network

Feed-Forward Neural Network

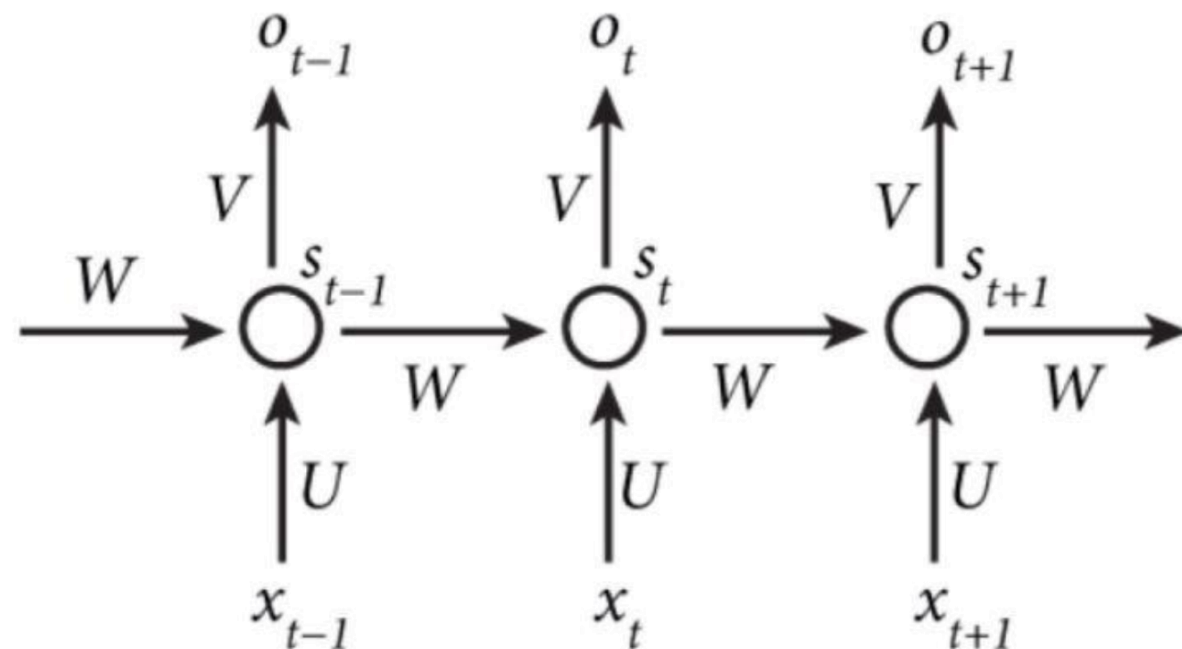
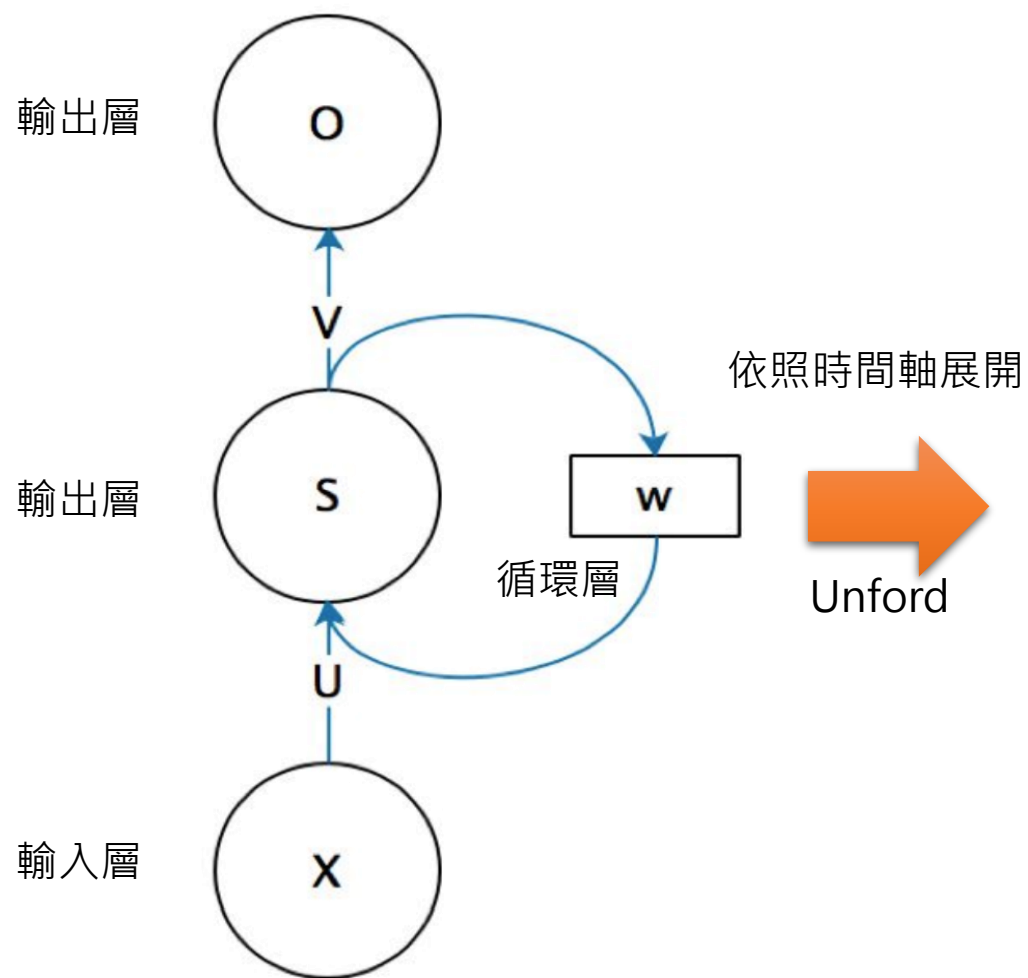
RNN將過去添加到現在 (1/2)

- RNN 有兩個輸入：**現在**和**最近的過去**。這很重要，因為數據序列包含有關接下來會發生什麼的關鍵訊息，這就是為什麼 RNN 可以做其他算法不能做的事情的原因。
- RNN 將權重應用於當前輸入以及先前的輸入。並調整梯度下降和隨時間反向傳播 (BPTT) 的權重。
- 前饋神經網絡只能將一個輸入映射到一個輸出，但 RNN 可以映射一對多、多對多（翻譯）和多對一（分類語音）。



[圖文引用] <https://pse.is/4ug6j6>

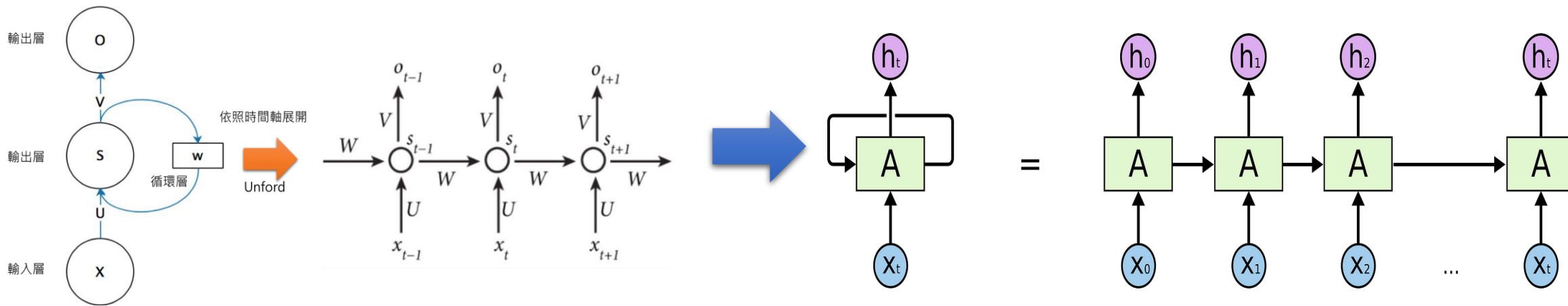
RNN將過去添加到現在 (2/2)



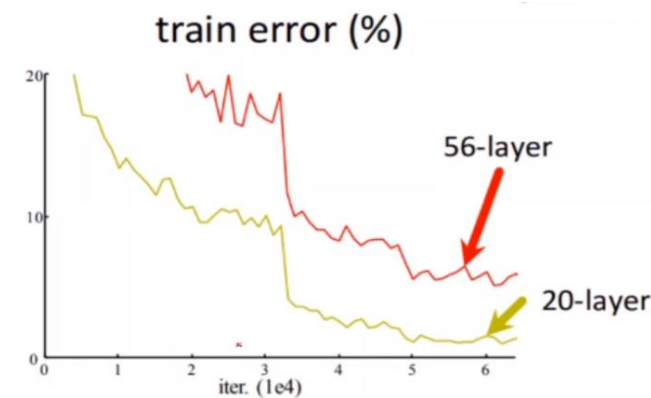
網路在 t 時刻接收到輸入 x_t 之後，隱藏層的值是 s_t ，輸出值是 o_t 。
 s_t 的值不僅僅取決於 x_t ，還取決於 s_{t-1} 。

長期記憶與過度學習對RNN造成的影響

- RNN是包含循環的網絡，在這個循環的結構中，每個神經網絡的模塊，讀取某個輸入，並輸出一個值，然後不斷循環。循環可以使得信息可以從當前步傳遞到下一步。因此RNN本質上與序列極度相關。
- **RNN長期記憶**：**RNN越前面的資訊對於後面的決策影響越小**，當所經過的時序越多，前面的資訊影響被稀釋趨近於零。

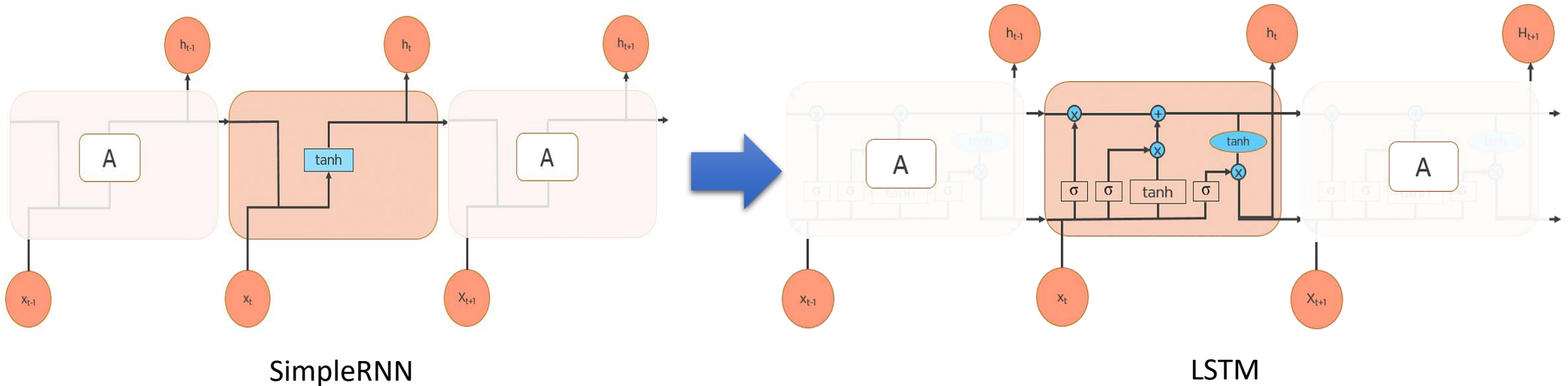
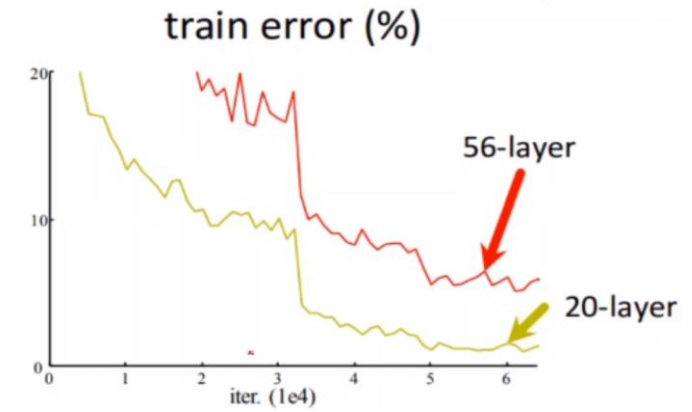


- **LSTM循環神經網路**：單純的RNN因為無法處理隨著遞歸，**權重指數級爆炸**或**梯度消失問題**，難以捕捉長期時間關聯；而結合不同的LSTM可以很好解決這個問題。長短期記憶(LSTM)網路是由Hochreiter和Schmidhuber於1997年所提出，在多個應用領域創造了精確度記錄。



長短期記憶網路 (Long short-term memory, LSTM)

- LSTM 是一種特殊的 RNN，能夠通過長時間記憶來學習長期依賴資訊。
- SimpleRNN 因為無法處理隨著遞歸，**權重指數級爆炸**或**梯度消失問題**，難以捕捉長期時間關聯。
- 長短期記憶(LSTM)網路是由Hochreiter和Schmidhuber於1997年所提出，在多個應用領域創造了精確度記錄。解決了**權重指數級爆炸**或**梯度消失問題**。LSTM 也有鏈狀結構，但重複模組的結構略有不同。它不是只有一個神經網路層，而是**四個相互作用**的層在進行訊息處理。



Part 2

建立及訓練RNN 相關語法



建立台灣股市收盤價格RNN

1.安裝 twstock 模組命令：

```
!pip install twstock
```

2. 使用 twstock 模組必須先匯入模組程式庫, 語法:

```
Import twstock
```

3.利用Stock()方法查詢個股歷史股票資料, 語法:

```
歷史股票資料變數 = twstock.Stock( '股票代號' )
```

4. 利用Stock物件屬性讀取指定歷史資料: 或 利用fetch()方法讀取指定期間歷史資料:

屬性	說明	屬性	說明	方法	傳回資料
date	日期	low	最低價	fetch(西元年, 月)	傳回參數指定月份的資料
capacity	總成交股數(單位:股)	price	收盤價	fetch_31()	傳回最近31日的資料
turnover	總成交金額(單位:元)	close	收盤價	fetch_from(西元年, 月)	傳回參數指定月份到現在資料
open	開盤價	change	漲跌價差		
high	最高價	transaction	成交筆數		

建立及訓練RNN

LSTM循環神經網路模型包含兩層: 一、輸入與LSTM層 二、輸出層

1.以`Sequential()`建立模型 :

```
model = Sequential()
```

2. 再用模型物件建立LSTM層,語法:

```
模型變數.add(LSTM(input_shape=(TIME_STEPS, INPUT_SIZE),  
units=數值, unroll=布林值))
```

- **input_shape** : 設定每一筆資料讀取次數,每次讀取多少像素,也就是每一筆資料輸入資料的維度(shape)。
- **TIME_STEPS** : 總共讀取多少個時間點的數據。例如:這裡為10個數值, 一次讀取一個數值, 需要10次。
- **INPUT_SIZE** : 每次讀取多少資料。例如:這裡一次讀取一個數值, **INPUT_SIZE** 為1。
- **units** : 設定神經元數目。
- **unroll** : **Ture**表示計算時會展開結構, 展開後可以縮短計算時間,但會占用更多記憶體。預設為: **False**。

3. LSTM層權重計算公式:

權重數 = $g \times (n \times (n+i) + n)$

- **g** : 狀態閘門數。simpleRNN 為 1, LSTM 為 4。
- **n** : 神經元數量。
- **i** : 每次讀取多少資料, 即**INPUT_SIZE**值。

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 256)	264192
dense (Dense)	(None, 1)	257

$4 \times (256 \times (256+1) + 256)$

$256 \times 1 + 1$

Total params: 264, 449
Trainable params: 264, 449
Non-trainable params: 0

卷積神經網路的互動式節點連結視覺化

- 卷積神經網路是各種電腦視覺任務的最先進方法的核心。神經網路的視覺化通常採用靜態節點連結圖的形式，它僅說明網路的結構，而不是行為。
- 在本文提出了一種新的基於手寫數字識別訓練的神經網路互動式視覺化，旨在顯示給定用戶提供的輸入的網路的實際行為。使用者可以透過繪圖板與網路交互，並即時觀察網路回應的啟動模式。
- 官網➡https://adamharley.com/nn_vis/

