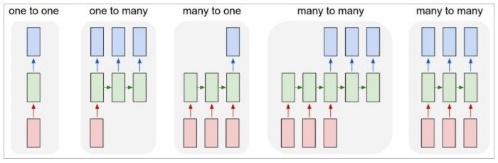
資科Final project

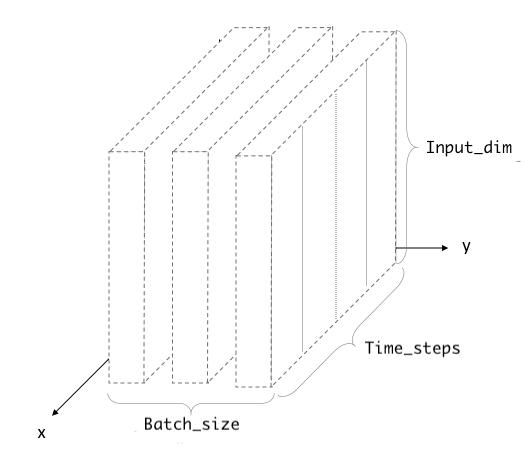
※ Why LSTM ,not RNN ?

※ 接者我們要透過keras搭建一個LSTM的神經網路，keras是一個專門用來訓練深度學習模型的API，在這邊我們要使用的是Sequential模型，從字面上翻譯為順序，給人的印象是那種簡單的線性模型；但實際上它應該理解為堆疊，通過堆疊許多層來建構深度神經網路。我們要做的事包括，建立輸入輸出以及隱藏層，並選擇優化器並選擇損失函數。以下的model.summary()函式可以將我們搭建的神經網路結構可視化出來。

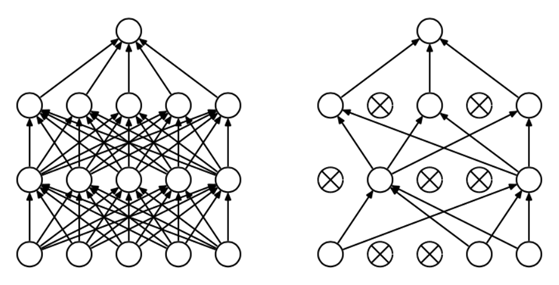
input\_shape其實是一個三維資料，代表(batch,time step,imput\_data)。

batch代表一次要輸入的資料筆數，time step 代表資料的時間維度，股票如果要藉由前30天的資料預測，則time step為30。Imput data代表單獨一個時間的資料，而units為每一個step輸出的數量，當要搭架多層的LSTM時，return sequences設置為TRUE，這樣的步驟又稱為sequence-to-sequence。

( https://ithelp.ithome.com.tw/articles/10214405)



接著，我們挑選ReLU(Rectified Linear Unit)作為激勵函數，它有兩個優點，一，可以幫助我們解決反向傳播訓練時產生的梯度消失問題，二，使部分神經元輸出為0，讓神經網路變得稀疏，我們也一邊使用dropout隨機消除神經元，緩解過度擬和的問題。

( https://medium.com/%E9%9B%9E%E9%9B%9E%E8%88%87%E5%85%94%E5%85%94%E7%9A%84%E5%B7%A5%E7%A8%8B%E4%B8%96%E7%95%8C/%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-ml-note-overfitting-%E9%81%8E%E5%BA%A6%E5%AD%B8%E7%BF%92-6196902481bb)

損失函數(loss function)可以幫助我們做現在ML模型好壞的評估

並以數值呈現。

當資料完成了一次完整的訓練，這個過程稱為一個epoch。Epoch的數量加，會使神經網路中權重的不斷更新，一般來說，當訓練次數增加，準確度會較高，以下是我們的實驗數據:(待補)