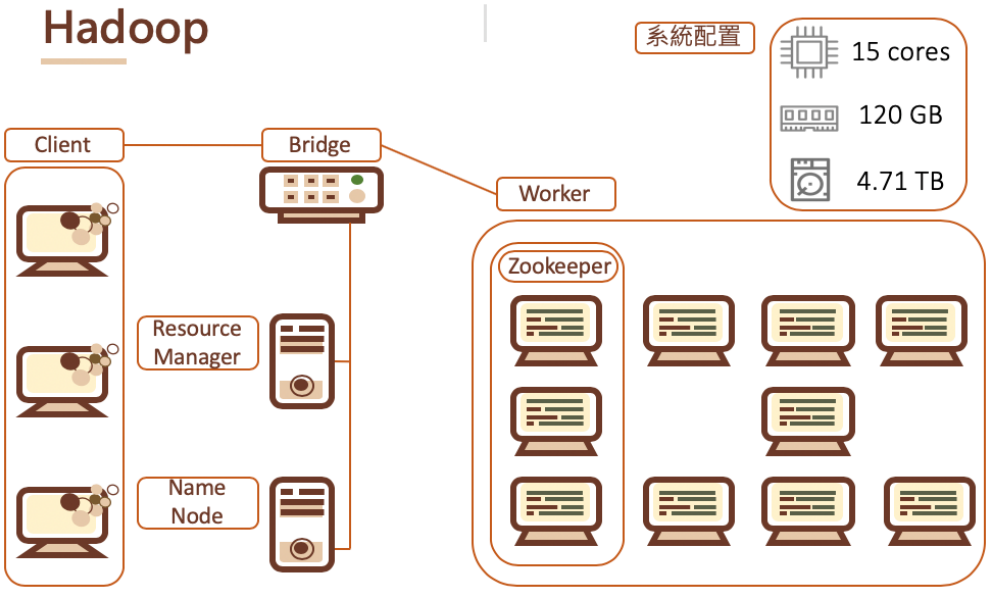
專題 Deep Learning 模型建置與結果說明

此次專案目標為**預測零售通路未來 16 天商品的銷量**，我主要負責 Hadoop 的建置管理以及深度學習的模型建置，我是使用 Keras，以 TensorFlow 作為後端，搭建三種深度學習的模型，分別為 Long Short-Term Memory、Deep Learning Neural Network 與 Convolutional Neural Network，並對每日做逐天預測。

環境架構部分我們使用 Hadoop 系統，先將 5 台實體主機切分為 15 台虛擬主機。故叢集運算資源有 15 顆 CPU 及 120GB 的記憶體，儲存空間為 4.71T [詳見圖一]。

我們在邊緣節點 Client 端配置了三台主機供我們的組員使用，其中 Name Node 與 Resource Manager 負責管理和分配叢集的資源，它們也互為彼此的備援主機，避免當機而導致叢集的故障與資料流失。Worker 主機一共 10 台，每台主機扮演一到兩種身分，其中最重要的就是 zookeeper 叢集，它的作用是實現 Name Node 和 Resource Manager 的 automatic failover(自動故障轉移)，當故障發生時備援主機會自動啟用，從而實現叢集的 HA(高可用性)。



圖一. Hadoop 系統架構

三種模型建置過程說明與結果分別如下：

1. LSTM (Long Short-Term Memory)

由於專題目標有時間序列的問題，所以 RNN(遞迴神經網路)是我首先考慮的模型，RNN 可不斷的更新自己的記憶，當前輸出與先前輸出都有關聯，故此架構擅長處理有序列的數據，但也由於 RNN 遞迴的特性，而造成它容易有梯度爆炸與梯度消失的問題。

RNN 的變體 LSTM(長短期記憶)可以解決這個問題，LSTM 可以通過 input gate, forget gate, output gate 三個控制閥來決定記憶的儲存和使用，將不重要的長期記憶遺忘，將重要的短期記憶加入長期記憶，這個架構比較適合我們的題目。

下圖右方為模型的架構，先使用一層 LSTM，後則用全連接的 Dense 層連接；最後，這個模型取得不錯的表現。不過，在組員討論後發現由於我們主要是使用移動平均法建立特徵欄位，原本的時序問題可能因此而降低，所以接下來我們使用深度學習神經網路的架構來建置模型

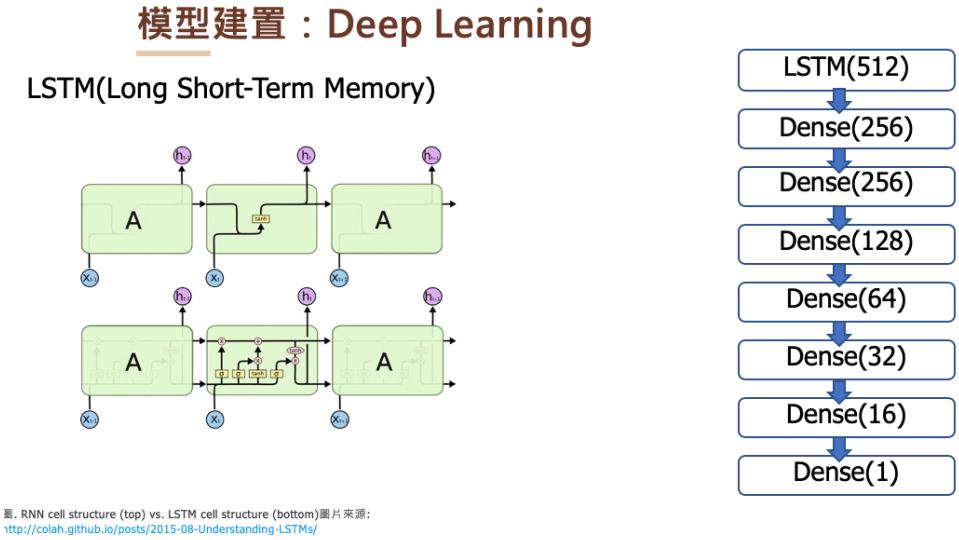


圖. RNN cell structure (top) vs. LSTM cell structure (bottom)圖片來源:
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

2. Deep Learning Neural Network

理論上網路越深效果可能會越好，而此模型共疊加 10 個 Layer，且全連接層的結構較為簡單。故運算的速度大幅度提升，每個 epoch 所需時間大約 3-4 秒，預測準確率也略優於前者 LSTM 的表現。

由於它的處理速度提升許多，故我使用此模型作為基準，最後，在將特徵欄位擴充到 339 個後，誤差值降到了 0.510，在 Kaggle 的排名也進入前 100 名。

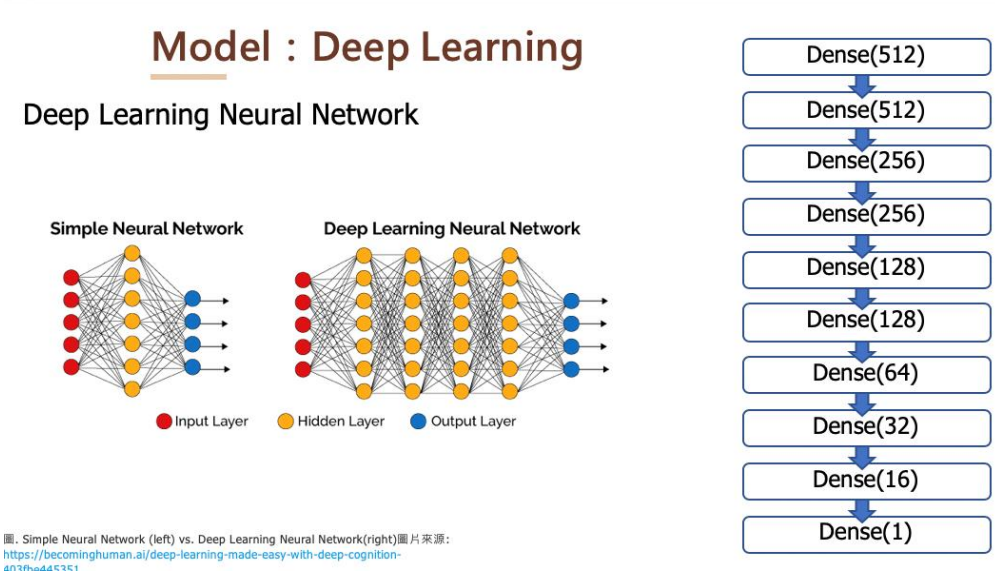


圖. Simple Neural Network (left) vs. Deep Learning Neural Network(right)圖片來源：
https://becominghuman.ai/deep-learning-made-easy-with-deep-cognition-

3. CNN (Convolutional Neural Network)

最後，我嘗試建立一個 CNN 的模型，此模型主要用於影像辨識，透過數層的卷積層和特徵擷取器來取得圖片的特徵進一步來辨識圖片。

我將特徵值處理成一串數列，讓它與圖片的感覺相仿，接著模仿 VGG16 的架構建立這個模型，再將原本 2D 結構改成 1D，讓電腦在一串數字中自行擷取重要的特徵，並進一步的對未來銷量做預測，結果雖然沒有比前一個模型好，但是相差的幅度也很小。我相信如果再進一步的加深網路，甚至疊加至 VGG16 一樣的 16 層，或 ResNet 的 50 層，或許會有更亮眼的表現。

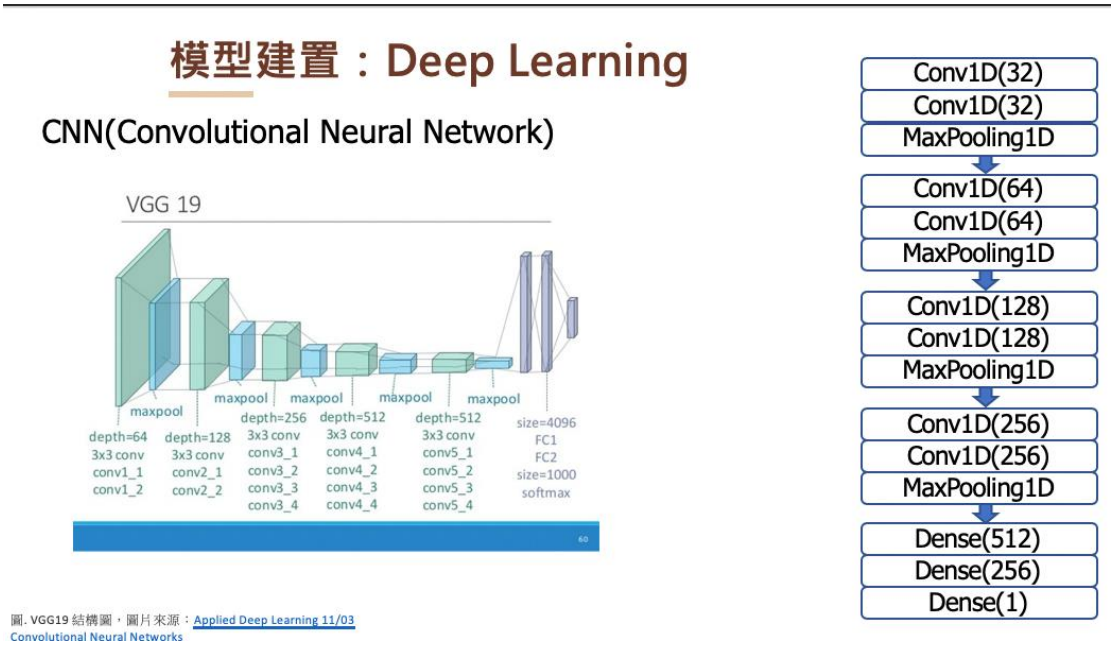


圖. VGG19 結構圖，圖片來源：Applied Deep Learning 11/03
Convolutional Neural Networks

4. 結論

下圖是在採用相同欄位數和資料筆數所繪製的誤差折線圖，X 軸是新增欄位的數量，因為使用的資料是以 7 天為一個週期建立，故 68 欄(3 天)則是增加資料的筆數，Y 軸則是誤差值。從這張圖我們可以看到隨著欄位數的增加，模型的誤差都會降低，但在增加資料筆數後，Catboost 與 LGBM 兩者表現反而變差，而 NN 是進步的。

從結果來說 Catboost 的表現是最差的。但由於 Catboost 的優勢是在於處理類別型欄位，由於我們的資料集是以數值行為主，故此次我們並沒有發揮它的潛力。在這次專題裡 LGBM 無論在處理速度或是預測誤差值的表現都是最好的，它也是現在 Kaggle 上最熱門的算法之一。而 NN 的表現略差於 LGBM，由於深度學習模型的上限一定程度上是取決於設備，所以相較於其他模型它仍有向上發展的彈性。