速記AI課程－深度學習入門（一）

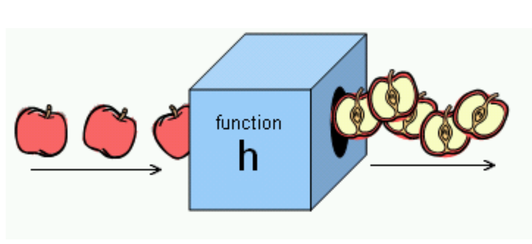
Gimi Kao Follow
Mar 10, 2018 · 10 min read

Introduction、 Practice與Tip

不知道是不是因為在校授課且年紀仍然很輕的關係，本堂課講師台大電機李宏毅教授的用詞十分鄉民（好合胃口），而且深入淺出，把複雜的概念用生活化卻不失專業的方式表達，聽得十分過癮。更難得可貴的是，對於學員各種天馬行空的問題（印象最深是老闆跟某位學員在爭辯AlphaGo到底算不算監督式學習...），也真的都知無不言言而不盡，絕對是台灣（甚至華文世界）講授深度學習（Deep Learning，以下簡稱DL）課程的最佳人選，沒有之一。

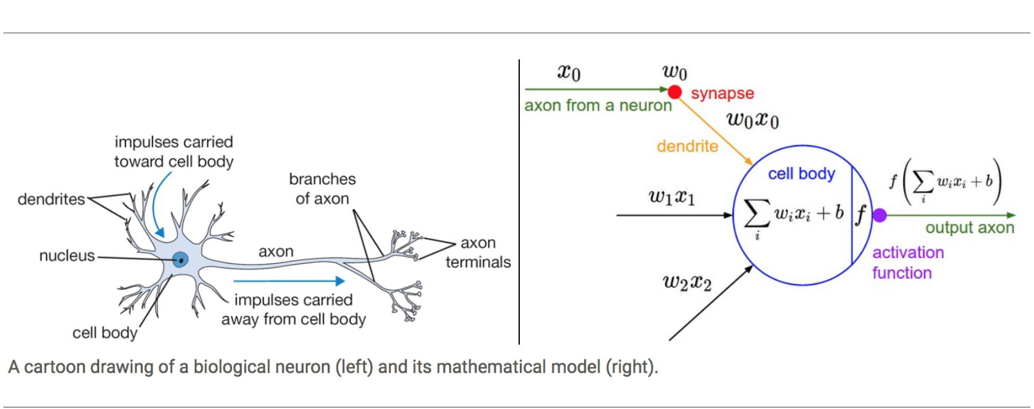
Introduction

課程一開始先提到什麼是Function，也就是給它輸入資料，就會產生結果的東西（好簡單）。比如說一個語音辨識Function，給它一段聲音，就會產生對應的文字；一個影像辨識Function，給它一張圖片，就會分辨出裡面的動物是不是貓。而多個Function集合在一起（Function set），就稱為Model。以影像辨識為例，可以辨識貓的Function，與其他可以辨識不同動物的Function集合在一起，就是所謂的Model。



Function示意圖

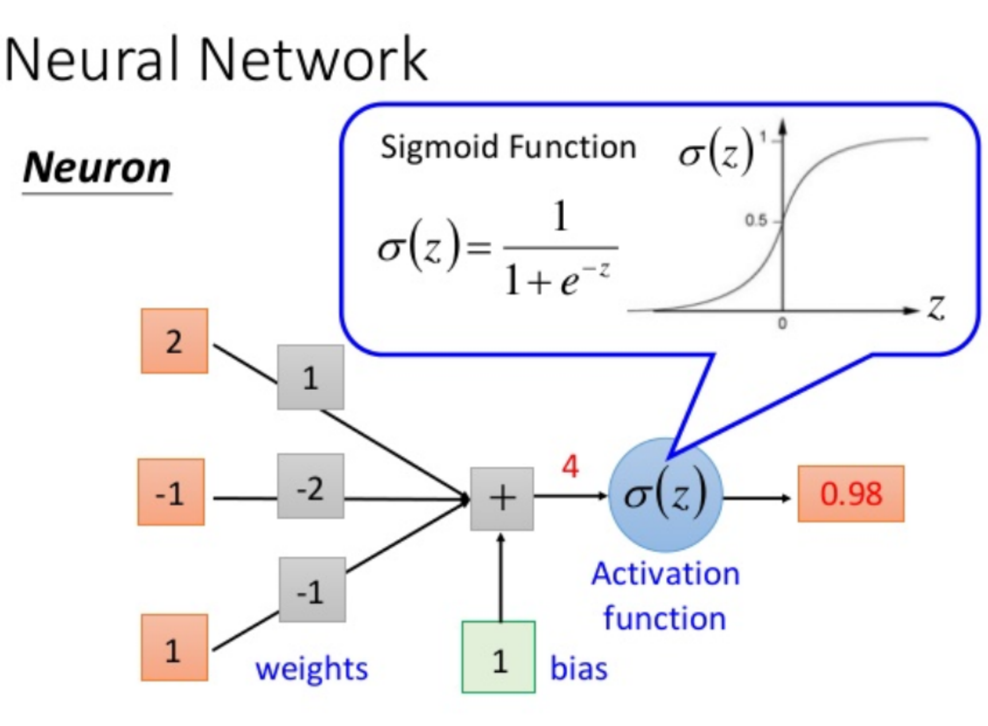
因此DL第一步，就是要找出這個Model或Function。DL（也可以說是ML）的概念就是透過大量資料，告訴Model輸入與輸出之間的關係，藉此來訓練模型。而這個模型訓練出來的結果好或不好，我們需要設計一個評斷的標準，這是第二步。最後，則是依照評斷的標準，選擇一個最好的Model。以下詳述這三步驟的細節。



A cartoon drawing of a biological neuron (left) and its mathematical model (right).

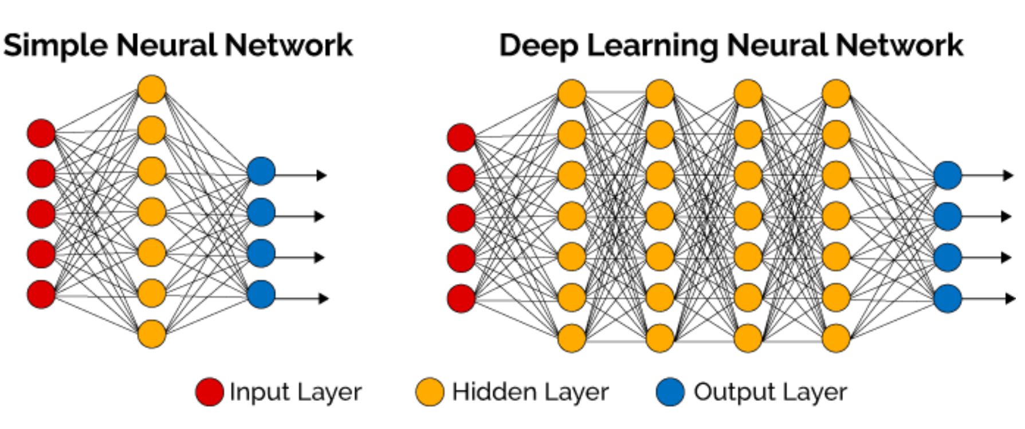
類神經網路與人類神經元示意圖

在DL領域中，這個Model是透過神經網路（Neural Network，以下簡稱NN）來訓練。NN是透過多個神經元組成，而一個神經元的基本組成是輸入的變數、權重（Weight）、誤差（Bias）以及激勵函數（Activation function），將輸入值轉換為一輸出值。除了激勵函數，基本上神經元的計算非常簡單，無非就是加減乘除，因此非常適合以GPU來進行大量簡單的運算。下圖是實際上計算的範例。



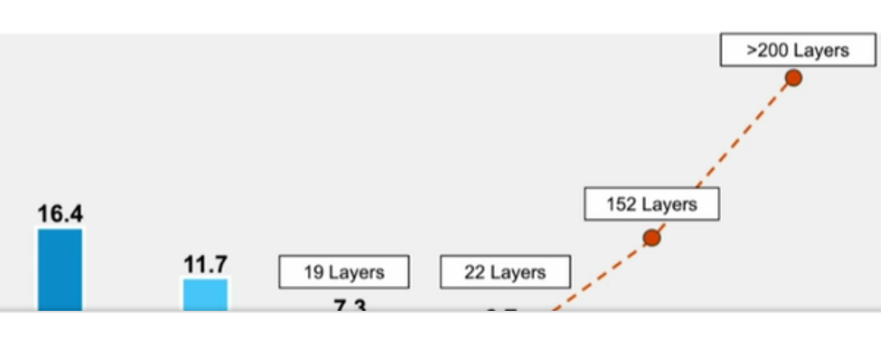
NN計算真的不複雜（From 李宏毅）

而DL簡言之，則是擁有多個且多層神經元之NN，而所謂的多層通常是指隱藏層（Hidden Layer）。DL要用幾層、每層要有幾個節點（神經元）、每個節點間要如何相連、要採用什麼激勵函數等（統稱網路架構），都是由我們決定，因此可以說是NN的天賦；而權重與誤差則是透過大量資料自動學習而得，可以視為是NN後天的努力結果。



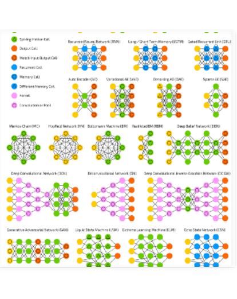
Really Deep...

那麼要多少層，才算是「深度」學習呢？以ImageNet所舉辦的大賽ILSVRC來看，歷年來冠軍所使用的層數越來越多，已超越台北101。理論上，越多層的DL，其效果應該不會比較少層之DL差。

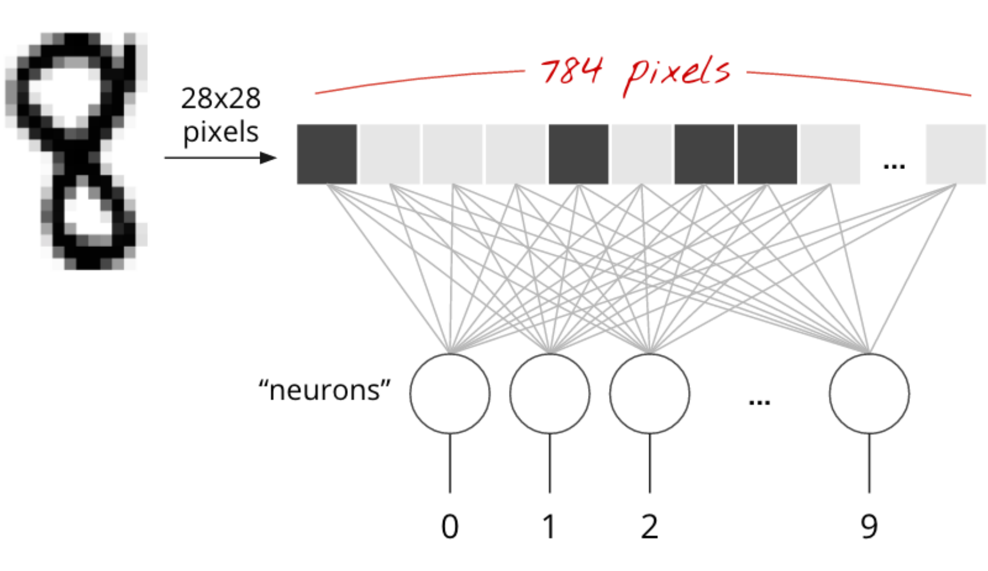


另外，除了層數以外，前面提到的網路架構也是重點之一，最常見的是所謂全連結網路（Fully Connected），即每個節點之間皆有輸入或輸出之關係。另外還有非常多種網路架構，可參考以下連結。

The mostly complete chart of Neural Networks, explained
The zoo of neural network types grows exponentially. One needs a map to navigate between many emerging architectures...
towardsdatascience.com



再以手寫數字辨識（DL界的Hello World!）為例，假設一張28x28的圖片，每一個像素都是一個輸入變數，因此有784個變數。假設某像素若有顏色則為1，沒有顏色則為0，透過大量的已知手寫數字照片來訓練NN，最後會學得到這張圖片是0的機率為多少、1的機率為多少...直到9的機率為多少，機率最高者則最有可能是該數字。



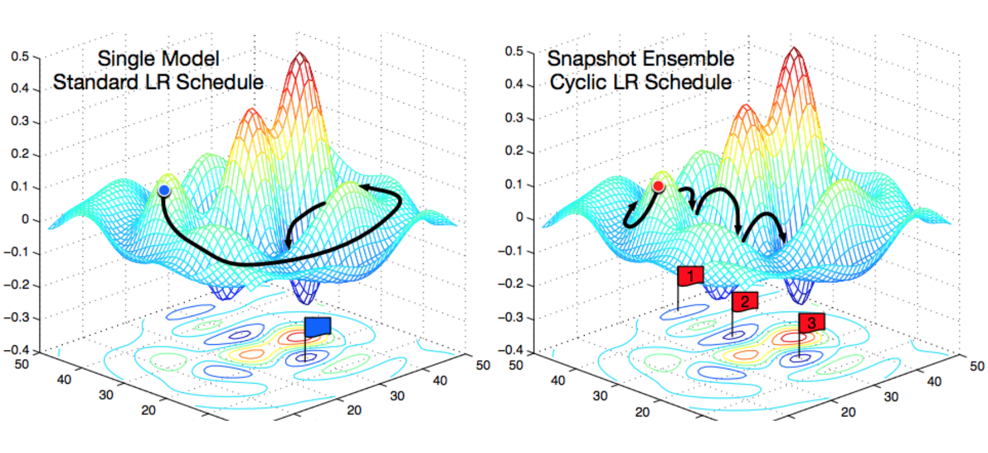
Hello Number!

目前對於DL需要幾層、架構如何設計才是最佳，仍沒有定論，還是需要AI訓練大師們的試誤與直覺來協助，雖然已有AutoML可以自己學出最好的網路構，不過目前多數應用都不太適用。若我們想要自己設計網路架構，則是所謂的卷積神經網路（Convolutional Neural Networks，以下簡稱CNN），針對某些應用來說特別有效。

模型訓練出來後，第二步驟則要設計評估模型好壞的標準。通常模型越好，其輸出值與我們的期望應該會越接近（如手寫數字為8，模型辨識出來也是8），換句話說，跟期望的差距（Loss）越小越好，所以總差異（Total Loss）最小的模型，應該就是最好的模型。

而最後一步則是找到最好的模型。更專業一點的說法，為了讓總差異最小，我們需要找到一組權重的組合，這個組夢幻數字又稱為θ，念做theta。其夢幻之處不僅是總差異可以最小，還有非常難找。通常DL的節點數都不會太少，而且權重的數字變化又可以非常多種，這樣的排列組合以現今計算能力要窮舉幾乎不可能，因此就需要利用梯度下降演算法（Gradient Descent，以下簡稱GD）的協助。

Top highlight



不是Big Bang的GD

因為無法窮舉，所以GD的概念是先隨機找一組參數，然後計算周圍是否有更好（更低）的參數組合，若有就移動過去，然後再計算還有沒有更好的，如此周而復始直到沒有更好的組合。此種方法的缺點是，有可能找到的點並非真正的最好的點，而是所謂區域最小值（Local Minima）。解決的方法是多由不同的起始點多做幾次，但這種方式還是看人品，只能多扶老太太過馬路了。

另一個說明GD的案例是電玩世紀帝國。正如同我們不知道哪裡有最低點，世紀帝國玩家一開始也僅可以看見周圍的環境，周圍都被霧所籠罩。我們只能派出斥侯一步步探索周圍，直到找到最低點。

熟悉的畫面...

GD的應用十分廣泛，許多深度學習的框架（如Tensorflow、Theano、Caffe）其實都是GD的不同實作而已。而GD應用在深度學習，又有個專有名詞，稱為反傳遞演算法（Backpropagation）。

Practice

這部分我個人覺得非常受用，李似乎是想打破DL的神話，提到了許多黑暗面，算是近期對於DL大量吹捧報導中的一股清流。

與ML的完整理論與解釋性相比，DL可以說是完敗。DL通常都是先試試看，有時候得出來的好結果是反直覺的，需要再回過頭解釋為什麼，因此實務上是遠遠超過理論的。（新領域很合理）

李當場訓練了手寫數字辨識的模型，一開始效果並不好，但把每個像素除以255以後（每個像素有256個色階），效果奇佳。這個「不知道為什麼就成功了」是在執行DL專案時常常遇到的狀況。

李做了一個很生動的比喻，深度學習就像手拉坏，看起來簡單，實際上不容易。拉坏機（各種DL框架）跟電源（GPU）差異並不多，拉坏師傅（人）才是關鍵，而且DL需要多方嘗試，老闆們與其一直催促什麼時候有結果，不如花多點錢買好的設備吧！

Tip

當模型效果不好時，不要先認為是Overfitting的錯。應該先檢查這個模型在訓練資料上的表現，如果訓練資料正確率不好，那代表一開始模型就沒有學好，要重新學習。若訓練資訊正確率高，但測試資料仍不好，才有可能是Overfitting。

如果是訓練資料表現不好，有以下幾個方法：

1. 加大網路節點數與層數，畢竟越大不會越差。李以知名的FizzBuss考題故事為例，只要把多加幾層，結果就會完全不一樣。

Use PyTorch to solve FizzBuzz
FizzBuzz 是一個常見的程式考題，題目很簡單，就是給一個整數，如果可以被 15 整除就回傳 FizzBuzz；可以被 3 整除就回傳 Fizz；被 5 整除就回傳 Buzz；都不能整除就回傳原本的數字。 用 Python...
ssarcandy.tw

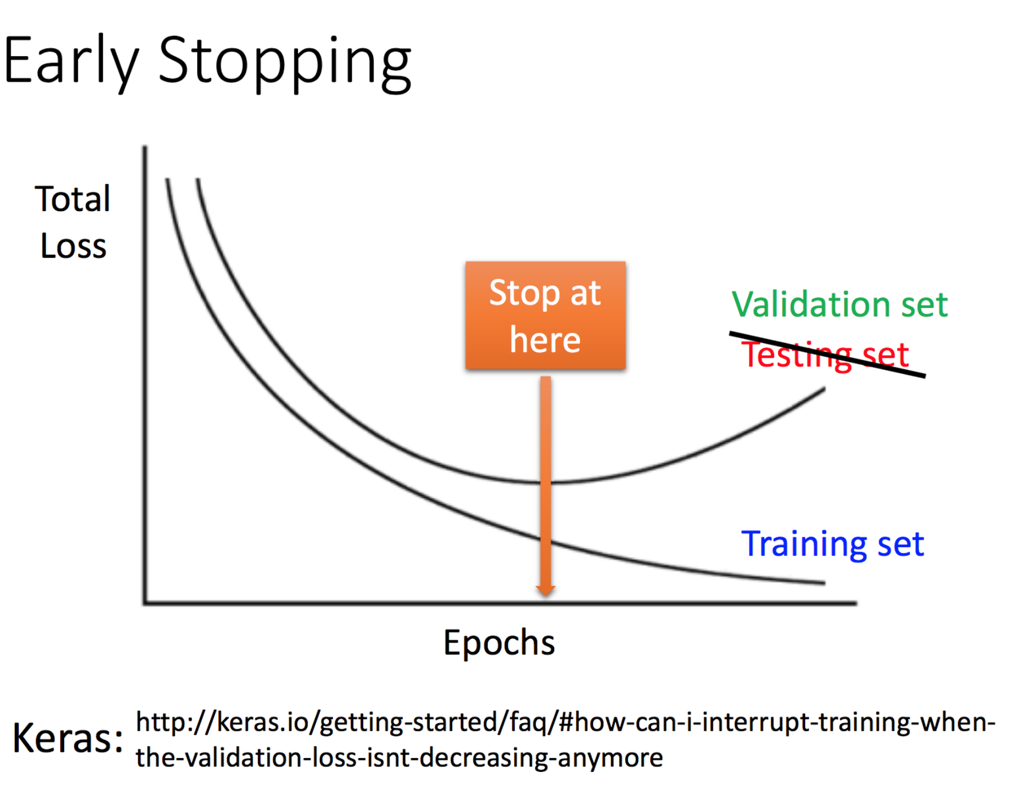
2. 調整Learning Rate。Learning rate決定了多快要更新DL內的參數，可以想成是多久學會新的東西。（有了新人就忘了舊人）

3. 資料標準化，如溫度與身高之Scale不同，需要經過轉換，讓各特徵值更接近。

4. 更換激勵函數。目前最常用的為ReLU（線性整流函數，Rectified Linear Unit），以及其他各種變形。李現場Demo一個SELU的新方法，結果較舊版本大幅進化。

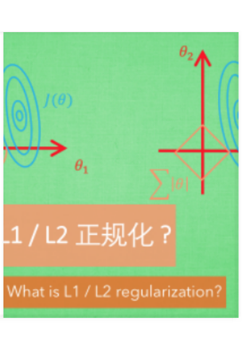
那如果是測試資料表現不好怎麼辦？有以下幾個方式：

1. 更多的資料：不管是取得更多資料，或是自行創造更多資料都可以，但資料的創況需要取決於對問題的理解。如Google Home的遠場語音辨識技術，就是利用虛擬房間、虛擬語者、虛擬噪音源之方式創造各種資料，來訓練其產品。另外，透過類似領域的資料來訓練（又稱Transfer learning），如先用中文語音較少，先透過英文語音訓練後，再來訓練中文，效果也不差。
2. 調整網路架構：把全連結的網路縮減為CNN，排除訓練資料表現好，但測試資料不好的function（模擬考強者），縮小網路架構。
3. Early Stopping：每次訓練時，透過驗證資料確認總差異沒有在變小後，隨即停止。

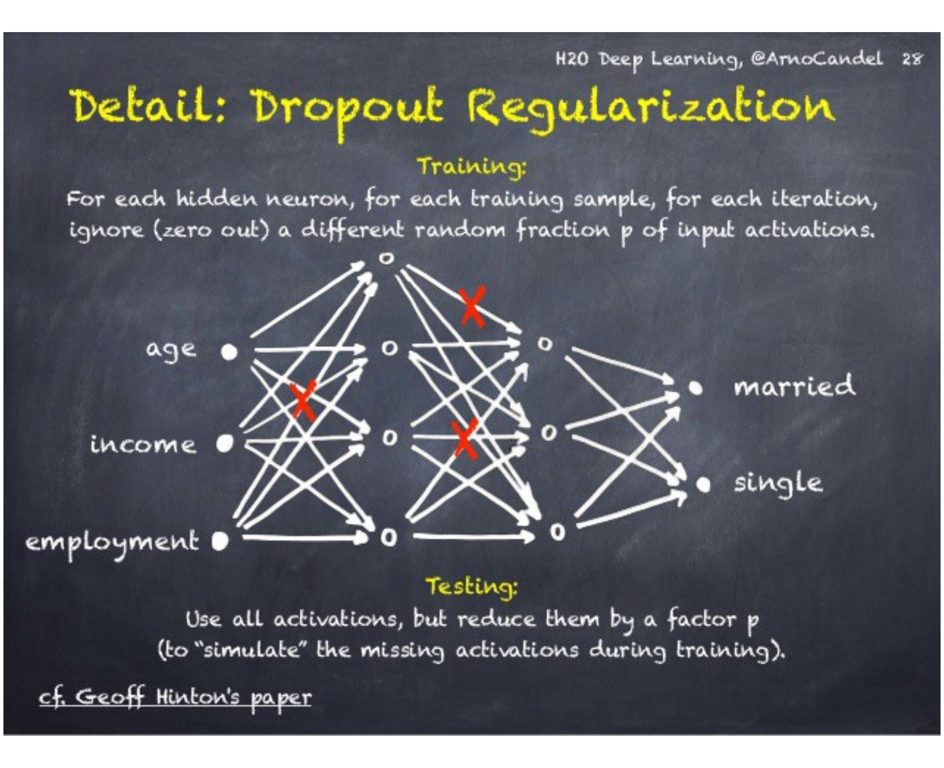


4. Regularization：為了避免Overfitting，我們必須不能僅透過總差異最小來判定模型的好壞，還要考量權重的大小。權重越大的變數，越有可能造成Overfitting，因此Regularization就是在懲罰這些權重大的變數，可以想成NBA的球隊只靠單一球星，是很難贏過團隊默契佳的球隊的。以下是更深入易懂的文章。

L1 / L2 正规化 (Regularization) - 有趣的机器学习 | 莫烦Python
我们知道, 过拟合就是所谓的模型对可见的数据过度自信, 非常完美的拟合上了这些数据, 如果具备过拟合的能力, 那么这个方程就可能是一个比较复杂的非线性方程 , 正是因为这里的 x^3 和 x^2 使得这条虚线能够被弯来弯去...
morvanzhou.github.io



5. Dropout：訓練時隨機丟棄某些神經元，雖然會導入訓練結果變差，但反而會提升測試的效果。



速記AI課程－深度學習入門（二）
Recurrent Neural Network、Application、Generative Adversarial Network
medium.com