

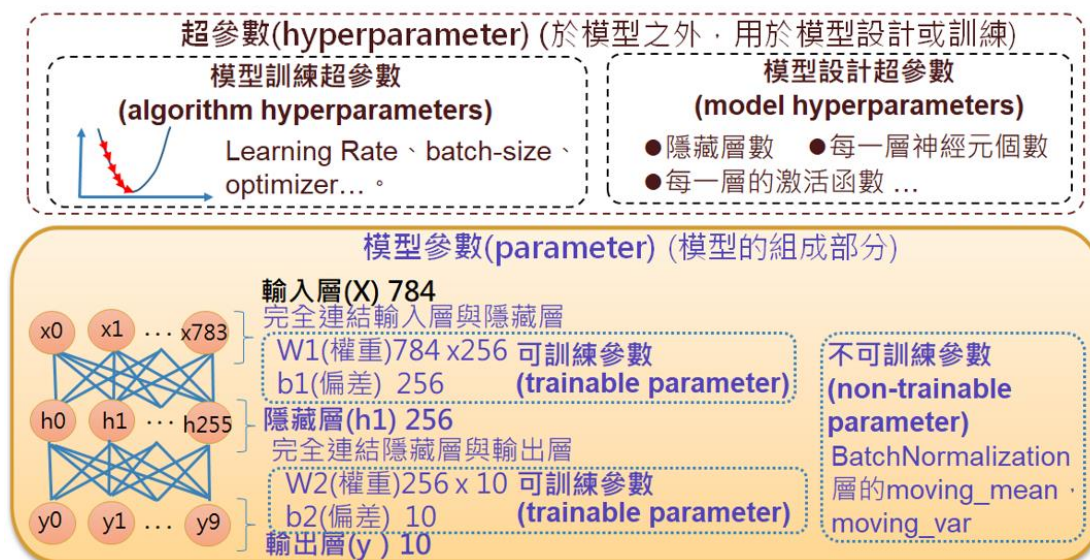
玩 TensorFlow Playground 理解深度學習模型(超參數與模型參數)

https://tensorflowkeras.blogspot.com/2022/07/tensorflow-playground_15.html?m=1

「深度學習」的初學者，接觸很多書或課程，看了很多數學公式與理論，可是對於初學者而言，可能覺得很抽象，也是一大進入障礙。本章教您如何透過玩 TensorFlow Playground，理解深度學習的概念。透過實際示範，介紹機器學習或深度學習的初學者，常混淆的 2 個名詞：什麼是超參數(hyperparameter)? 什麼是模型參數(parameter)? 為何越多層隱藏層功能越強? 透過實際操作瞭解如何設定超參數: Learning rate、Activation...等?

超參數與模型參數

機器學習或深度學習的初學者，常混淆的 2 個名詞：什麼是超參數(hyperparameter)與模型參數(parameter)?



以上詳細說明整理如下表格：

	超參數(hyperparameter)		模型參數(parameter)	
	於模型之外，用於模型設計或訓練		模型的組成部分	
分類	模型訓練超參數 (algorithm hyperparameters)	模型設計超參數 (model hyperparameters)	可訓練參數 (trainable parameter)	不可訓練參數 (non-trainable parameter)
說明	用於控制訓練過程：會影響訓練的速度、準確率...等。	用於模型設計：會影響模型的表現、準確率、underfitting、overfitting..等。	模型訓練過程中，反向傳播會更新「可訓練參數」，學習資料的規律。	模型訓練過程中，反向傳播不會更新「不可訓練參數」，通常是常數或統計數字。
例如	Learning Rate、batch-size、optimizer..等。	隱藏層數量、每一層神經元個數、Activation..等。	Weight(權重)、Bias(偏差)	BatchNormalization層的moving_mean、moving_var

以下會介紹模型訓練超參數、模型設計超參數、可訓練參數。本書第 25 章會詳細介紹 BatchNormalization 層的不可訓練參數：moving_mean、moving_var。

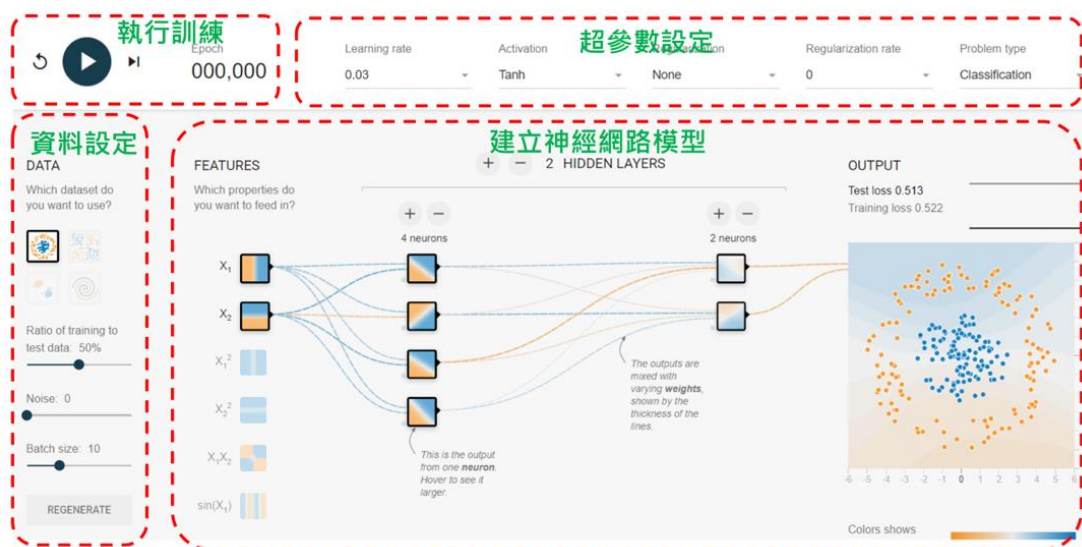
開始使用 TensorFlow Playground

TensorFlow Playground 是一個網頁應用程式以 d3.js 開發，您只需要有瀏覽器就可以執行，不需要數學與程式就可以瞭解神經網路，網址: <http://playground.tensorflow.org/>

The screenshot shows the TensorFlow Playground interface. A red dashed box highlights the URL in the browser's address bar, with a callout bubble stating: "1. 輸入網址：http://playground.tensorflow.org/". Another red box highlights the introductory text: "Tinker With a Neural Network Right Here in Your Browser. Don't Worry, You Can't Break It. We Promise.", with a callout bubble stating: "2 文字說明：因為TensorFlow Playground在瀏覽器中執行，你可以盡量使用，不會壞掉也不會影響你的電腦". The interface itself shows a neural network diagram with 4 input neurons, 2 hidden layers, and 2 output neurons. The training progress is shown as Epoch 000,025. The output section displays Test loss 0.328 and Training loss 0.

功能區塊介紹

TensorFlow Playground 功能區塊介紹如下：

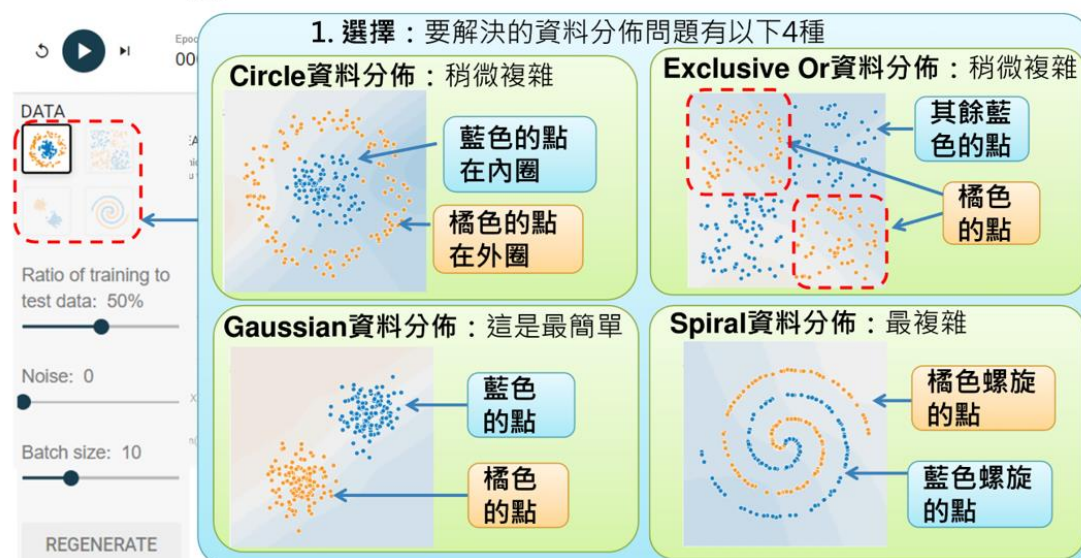


TensorFlow Playground 資料設定

建立模型前，必須進行資料設定。

Step1.選擇要處理的資料分類問題

TensorFlow Playground 提供 4 種資料分類問題，讓你進行實驗。



後續我們將建立神經網路模型，經過訓練後，將藍色的點與橘色的點進行分類。

Step2.產生資料設定

請依照下列步驟，產生訓練資料與測試資料：

The screenshot shows the TensorFlow Playground interface with the following annotations:

- 1. 點選：Circle資料分佈** (Click: Circle data distribution) - Points to the 'Circle' dataset icon in the 'DATA' section.
- 2. 訓練與測試資料比例：預設是50%** (Training and test data ratio: default is 50%) - Points to the 'Ratio of training to test data: 50%' slider.
- 3. 產生雜訊比例：預設是0** (Generate noise ratio: default is 0) - Points to the 'Noise: 0' slider.
- 4. 訓練時的Batch Size：預設是10** (Batch size during training: default is 10) - Points to the 'Batch size: 10' slider.
- 5. 按下此圖示：開始產生訓練與測試資料** (Press this icon: start generating training and test data) - Points to the 'REGENERATE' button.
- 6. 顯示資料分佈：預設只顯示訓練資料，藍色的點在內圈，橘色的點在外圈** (Display data distribution: default only shows training data, blue dots in the inner circle, orange dots in the outer circle) - Points to the 'OUTPUT' plot showing two concentric circles of points.

At the top, the settings are: Epoch: 000,000, Learning rate: 0.03, Activation: Tanh, Regularization: None, Regularization rate: 0, Problem type: Classification.

TensorFlow Playground 建立模型

選擇的資料類型問題後，我們就可以建立神經網路模型解決問題。進入 TensorFlow Playground 後，預設的神經網路，說明如下：

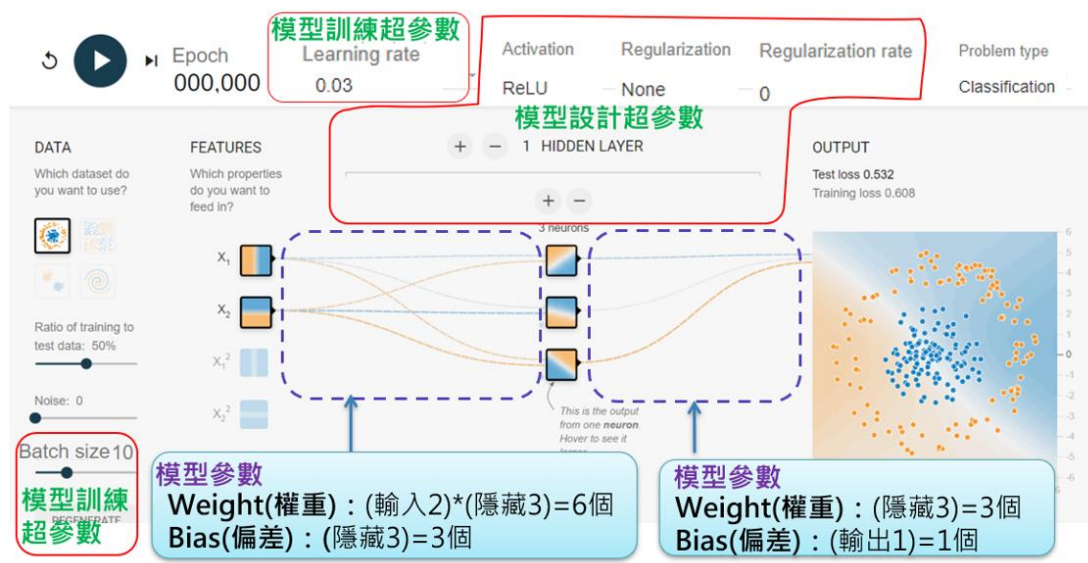
The screenshot shows the TensorFlow Playground interface with the following annotations for the default neural network structure:

- 輸入層：預設2個神經元** (Input layer: default 2 neurons) - Points to the input layer with 2 neurons.
- 隱藏層1：預設4個神經元** (Hidden layer 1: default 4 neurons) - Points to the first hidden layer with 4 neurons.
- 隱藏層2：預設2個神經元** (Hidden layer 2: default 2 neurons) - Points to the second hidden layer with 2 neurons.
- 輸出層：1個神經元。訓練後，能夠分類黃色與藍色的點** (Output layer: 1 neuron. After training, it can classify yellow and blue points) - Points to the output layer with 1 neuron.
- Weight與Bias：讓輸入層與隱藏層1，完全相連結** (Weight and Bias: fully connect input layer and hidden layer 1) - Points to the connections between the input and first hidden layers.
- Weight與Bias：讓隱藏層1與隱藏層2，完全相連結** (Weight and Bias: fully connect hidden layer 1 and hidden layer 2) - Points to the connections between the first and second hidden layers.
- Weight與Bias：讓隱藏層2與輸出層，完全相連結** (Weight and Bias: fully connect hidden layer 2 and output layer) - Points to the connection between the second hidden layer and the output layer.

At the top, the settings are: Epoch: 0.03, Learning rate: 0.03, Activation: Tanh, Regularization: None, Regularization rate: 0, Problem type: Classification. The 'OUTPUT' plot shows the training loss (0.522) and test loss (0.526).

TensorFlow Playground 超參數與模型參數

TensorFlow Playground 的「模型訓練超參數」、「模型設計超參數」與「模型參數」說明如下：

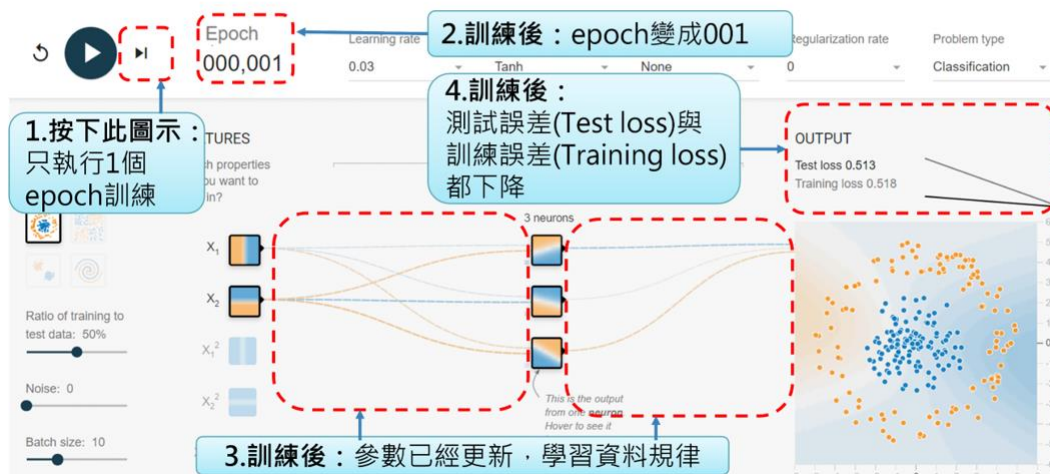


模型參數(訓練時學習資料的規律)

初學神經網路模型，我們常常會有疑問，究竟模型如何學習資料的規律?其實就是模型訓練過程中，反向傳播會會更新模型「可訓練參數」(Weight 與 Bias 的數值)，來學習資料的規律，使測試誤差(Test loss)與訓練誤差(Training loss)都下降。

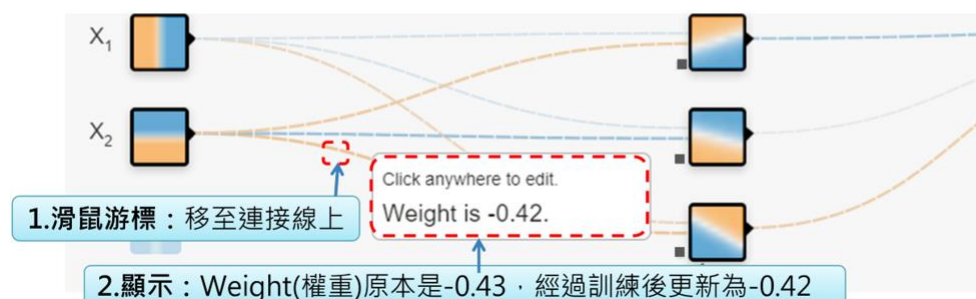
Step1. 執行一個 epoch 的訓練

例如以下執行一個 epoch 的訓練(當全部資料訓練一次，稱為一個 epoch 訓練)。



Step2.查看訓練後的模型參數 Weight(權重)

訓練模型後，會透過更新 Weight(權重)的數值，也就是改變訊號傳遞至接收神經元容易的程度，來學習資料的規律。



Step2. 查看訓練後的模型參數 Bias(偏差)

訓練模型後，還會透過更新 Bias(偏差)的數值，也就是改變接收神經元越容易接收訊息的程度，來學習資料的規律。



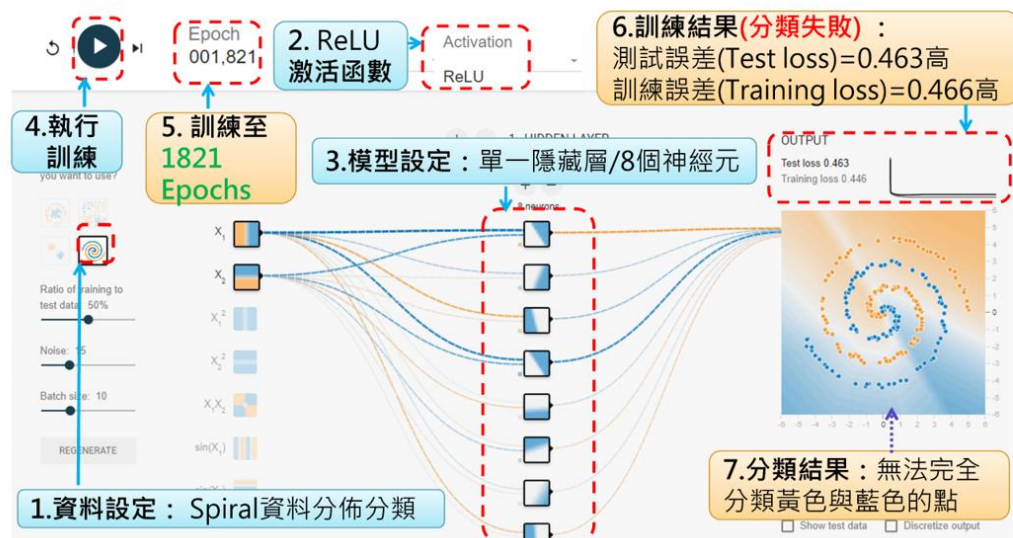
如上圖，執行一個 epoch 的訓練後，將滑鼠游標移至連接線上，你可以看到 Bias(偏差)已經改變。

模型設計超參數：隱藏層數

以下分別以 1 個隱藏層、2 個隱藏層，試圖分類 Spiral 資料分佈的問題，並且觀察結果有何不同？

Step1.一個隱藏層 8 個神經元(分類失敗)

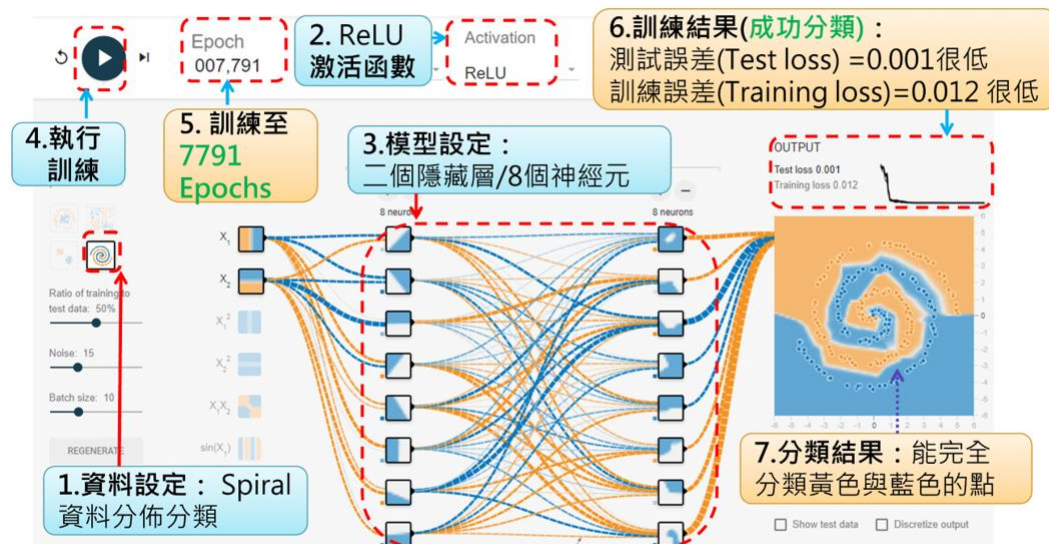
先設定一個隱藏層 8 個神經元，請依照以下說明設定超參數，設定後執行訓練。



如上圖，由於一個隱藏層 8 個神經元模型太簡單，分類 Spiral 資料分佈是比較複雜的問題，所以執行訓練後：Loss=0.463 很高，分類失敗(Underfitting 擬合不足)。

Step2.二個隱藏層/8 個神經元(分類成功)

增加為二個隱藏層，其餘超參數不變，設定後執行訓練。



如上圖，模型複雜度，與要分類的資料分佈複雜度相當，執行訓練後：Loss 只有 0.01，分類成功(Appropriate fitting 適當擬合)。

Step3.結論：隱藏層越多，能分類越複雜的資料分佈

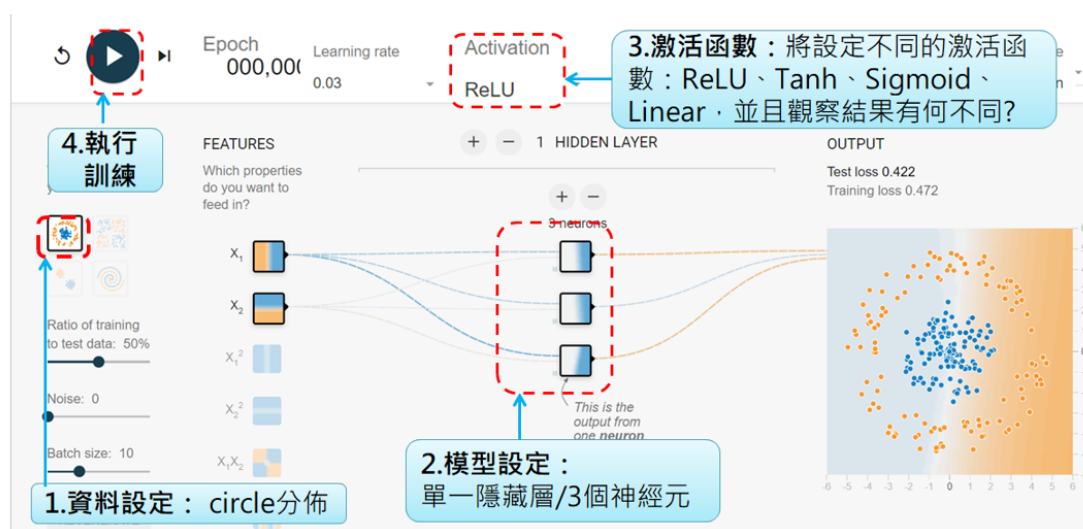
以下我們整理當隱藏層分別是 1、2 時，隨著隱藏層的增加，Loss 隨之降低。代表隱藏層越多，能分類越複雜的資料分佈。



模型設計超參數： Activation 激活函數

以下設定不同的激活函數：ReLU、Tanh、Sigmoid、Linear，並且觀察結果有何不同？

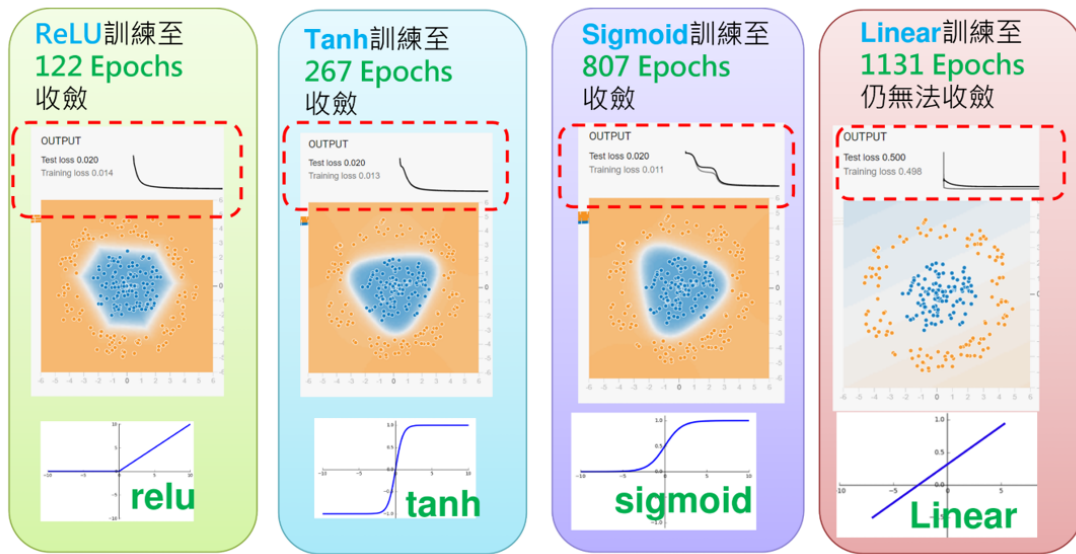
Step1. 設定不同的 Activation 超參數



關於詳細的激活函數設定與訓練過程，詳細請參考本書

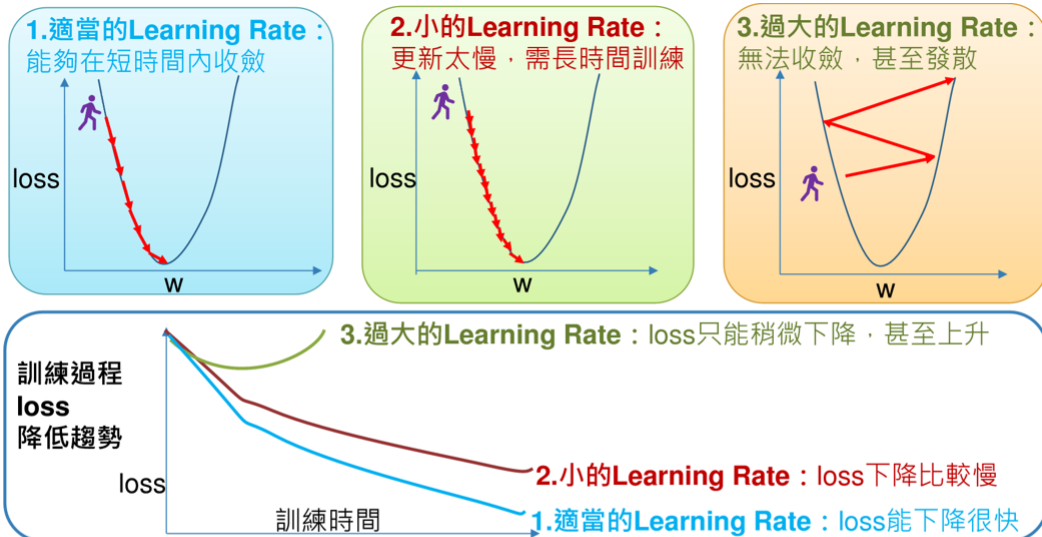
Step2.比較不同的激活函數

以下的比較不同的激活函數分別是 ReLU、Tanh、Sigmoid、Linear：結論是 ReLU 最快收斂。



模型訓練超參數： Learning Rate

Learning Rate 是模型訓練超參數，用於控制訓練過程。當我們使用梯度下降法：進行訓練使 loss 降低。必須選擇適當的 Learning Rate。如下圖，不同的 Learning Rate，會有以下不同的結果。



以老師教學為例說明如下：Learning Rate 相當於老師的教學進度，老師教的慢學生當然學的慢。但是如果老師教得太快，反而學生學不到知識，也就是所謂欲速則不達。只有老師適當的教學速度，學生才能學得又快又好。

Step1. 設定不同的 Learning Rate

4. 執行訓練

Epoch: 000,000

Learning rate: 3

3. 設定：不同的Learning Rate：3、0.3、0.03並且觀察結果有何不同？

1. 資料設定： Circle資料分佈分類

2. 模型設定： 單一隱藏層/3個神經元

FEATURES: Which properties do you want to feed in? X_1 , X_2 , X_1^2 , X_2^2 , X_1X_2

HIDDEN LAYER: 3 neurons

OUTPUT: Test loss 0.509, Training loss 0.525

Ratio of training to test data: 50%

Noise: 0

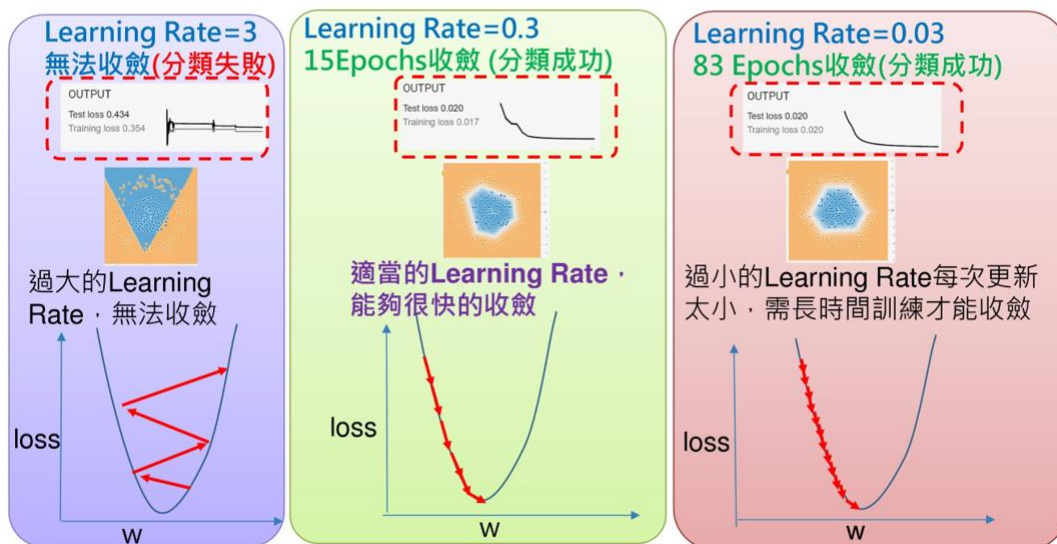
Batch size: 10

This is the output from one neuron. Hover to see it.

關於設定不同的 Learning Rate 與訓練的詳細過程，請參考本書。

Step2.比較不同的 Learning Rate

以下的比較不同的 Learning Rate 分別是 3、0.3、0.03：結論是以本範例而言，適當的 Learning Rate=0.3，能夠很快的收斂。



以上內容節錄自這本書，很適合 Deep Learning 深度學習入門初學者:

[圖解 TensorFlow 2 初學篇：實作 tf.keras + Colab 雲端、深度學習、人工智慧、影像辨識](#)

[購買本書 限時特價專區》](#)

[天瓏網路書店](#)

[博客來網路書店](#)