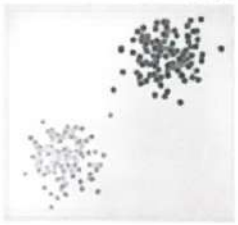
Homework 4: Explore TensorFlow Playground

(C) Gaussian data set: Can you obtain test loss of less than 0.1 using only the raw inputs? i.e., using only X_1 and X_2? Try to use as few neurons as possible. Report the number of neurons used in your model and justify your answer.

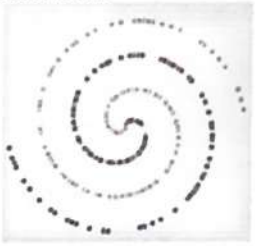


Now suppose you use features X_1^2 and X_2^2 instead of X_1 and X_2. Can you still obtain a test loss of less than 0.1? Justify your answer.

Homework 4: Explore TensorFlow Playground

(D) Spiral data set: Can you obtain test loss of less than 0.1 using only the raw inputs? i.e., using only X_1 and X_2? Please share your screenshot of the network as described.

Now you are allowed to use any/all of the 7 different input features (X_1, X_2, X_1^2, X_2^2, X_1X_2, $\sin(X_1)$, and $\sin(X_2)$). Can you obtain test loss of less than 0.1 for the Spiral data set with fewer neurons? Please share your screenshot of the network as described.

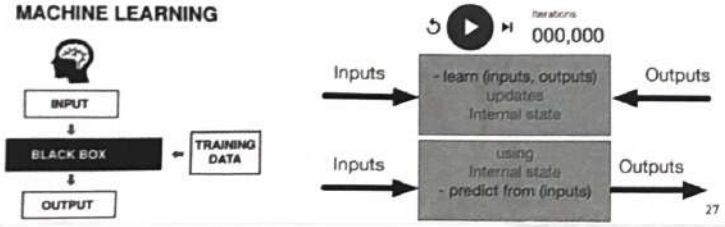


(E) What is L1 regulations? Explain detailly.
(F) What is L2 regulations? Explain detailly.
(G) What is regulation rate? Explain detailly.
(H) What is Noise? Explain detailly.
(I) What is Batch size? Explain detailly.

TensorFlow Playground

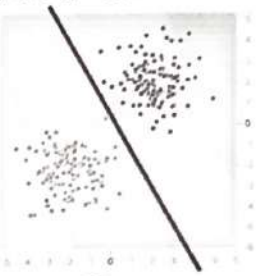
http://playground.tensorflow.org/

- An interactive visualization of neural networks, written in typescript using d3.js.由Daniel Smilkov、Shan Carter設計。開源碼：https://github.com/tensorflow/playground
- 這是一個用 JavaScript 寫的網頁應用，能讓你在瀏覽器中測試、訓練及執行類神經網路，並且可以讓你點擊按鈕和調整參數，從而了解它是怎麼工作的。

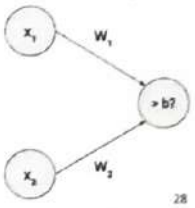


一個簡單的分類問題：傳統解題做法

- 假設有一個資料集合：每個資料有兩個座標值：x1（橫坐標）和x2（縱坐標）。分類為橙色組和藍色組。
- 要區分一個資料是橙色的還是藍色的，你該如何編寫程式？
- 也許你可畫一條對角線來分隔兩群資料，定義一些參數值以確定每個資料屬於哪一群。程式應該看起來是這樣：



if (x1 *w1 + x2 *w2 > b) then "屬於藍色組"
else "屬於橙色組".

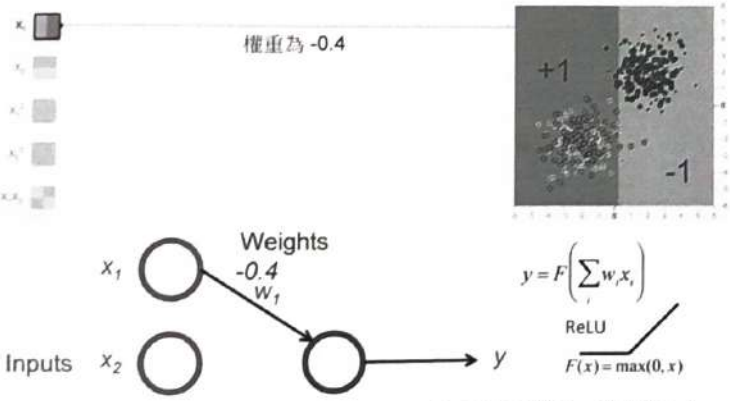


✓ 但問題的關鍵是程式設計師必須為 w1、w2 和 b 找到合適的值——即所謂的參數值。

原文網址：https://read01.com/yMg576.html

Playground 的機器學習如何解決這問題？

✓ 先遞減為0 HIDDEN LAYERS，X_1一個特徵。Activation設定為ReLU。

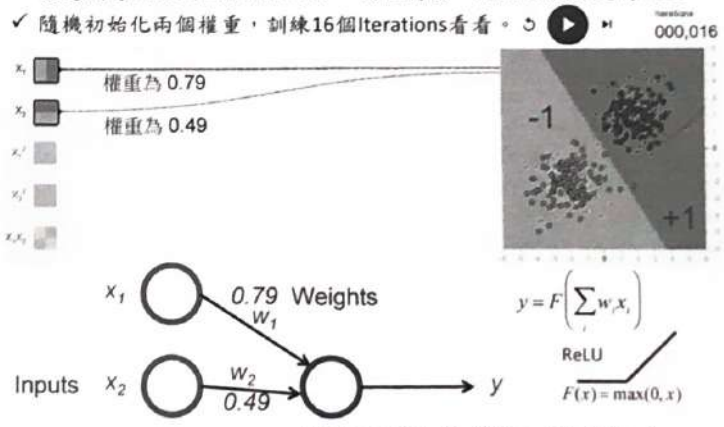


$y = max(0, -0.4 * x_1)$

Playground 的機器學習如何解決這問題？

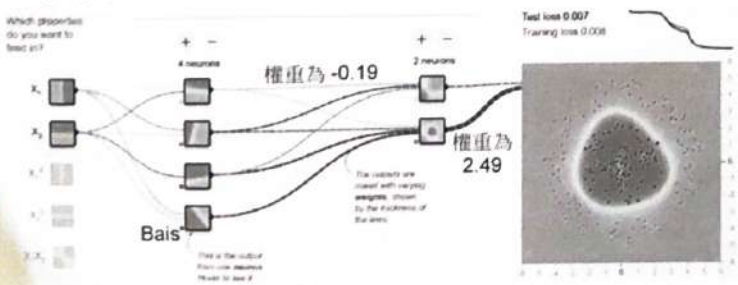
✓ 先遞減為0 HIDDEN LAYERS，兩個特徵。Activation設定為ReLU。

✓ 隨機初始化兩個權重，訓練16個Iterations看看。



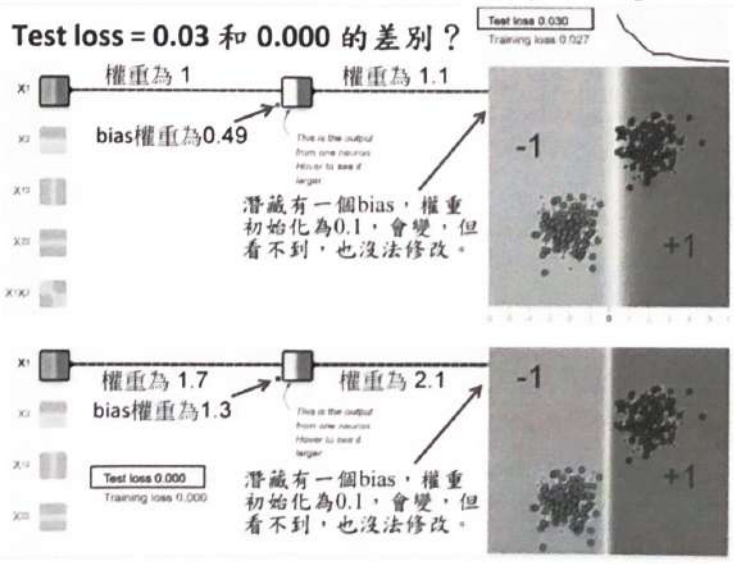
$y = max(0, 0.79*x_1 + 0.49*x_2)$

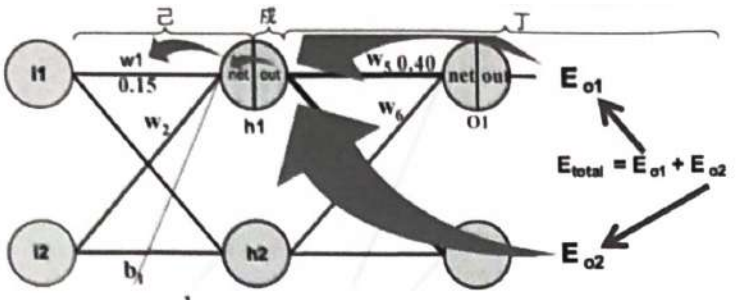
- 可以選擇這個神經網路中自己想要輸入的特徵(feature)和層數，以及每層有幾個神經元。
- 層與層之間的線條顏色由連接神經元之間的權重決定。藍色代表權重爲正，橙色代表權重爲負。線條的粗細代表權重大小。按兩下線條可人工設定其權重。



- 每個神經元左下角有一個小點Bais，可以觀察並按兩下調整。
- 把游標停留在神經元上，可以在右側觀察它的輸出。

Test loss = 0.03 和 0.000 的差別？





$$out_{h1} = \frac{1}{1+e^{net_{h1}}}$$

求 $\frac{\partial out_{h1}}{\partial net_{h1}} = out_{h1}(1-out_{h1}) = 0.59326999(1-0.59326999) = 0.241300709$

$net_{h1} = w_1 * i_1 + w_2 * i_2 + b_1 * 1$ 故 $\frac{\partial net_{h1}}{\partial w_1} = i_1 = 0.05$

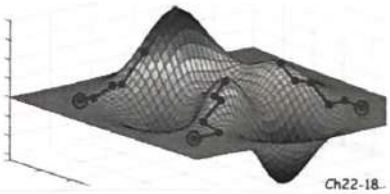
$\frac{\partial E_{total}}{\partial w_1} = \frac{丁\partial E_{total}}{\partial out_{h_1}} * \frac{戊\partial out_{h_1}}{\partial net_{h1}} * \frac{己\partial net_{h1}}{\partial w_1} = 0.036350306 * 0.241300709 * 0.05 = 0.000438568$

更新w_1： $w_1^+ = w_1 - \eta * \frac{\partial E_{total}}{\partial w_1} = 0.15 - 0.5 * 0.000438568 = 0.149780716$

同理更新： w_2=0.2 w_3=0.25 w_4=0.3

$w_2^+ = 0.19956143$ $w_3^+ = 0.24975114$ $w_4^+ = 0.29950229$

- 現在已經更新所有的w權重。(我們沒有更動b_1及b_2) 練習
- 在最初，輸入為0.05和0.1的時候，網路的誤差為0.298371109。
- 經過第一次Backpropagation後，誤差降低到了0.291027924，看來降得不多。
- 在重複這個過程10000次以後，網路的誤差就降到了0.0000351085。 開源碼：https://github.com/mattm/simple-neural-network
- 這個時候，當我們把0.05和0.1再輸入進去，兩個神經元的輸出為0.015912196(理想值0.01)和0.984065734(理想值 0.99)。

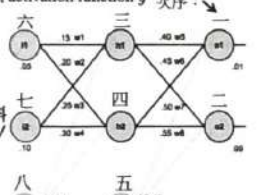


Ch22-18

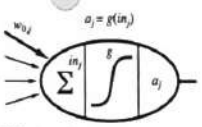
function BACK-PROP-LEARNING(*examples*, *network*) **returns** a neural network //反向傳播學習法

inputs: *examples*, a set of examples, each with input vector **x** and output vector **y**
network, a multilayer network with L layers, weights $w_{i,j}$, activation function g △[j]的計算次序：↘

local variables: Δ, a vector of errors, indexed by network node

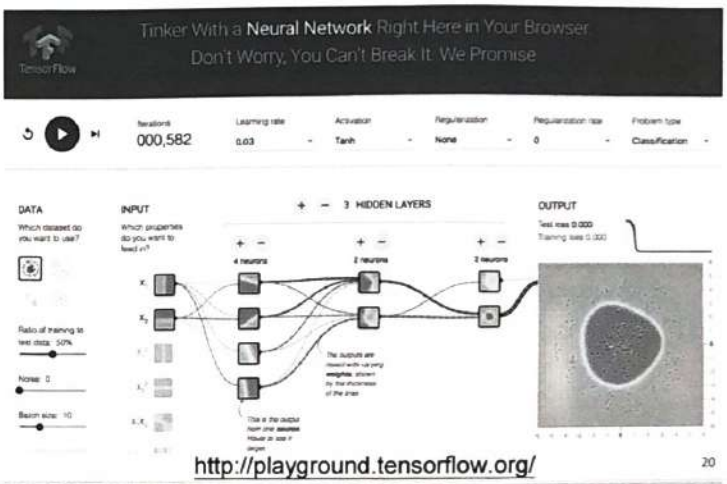


repeat
  for each weight $w_{i,j}$ in *network* **do**
    $w_{i,j} \leftarrow$ a small random number // 隨機設定初始權重
  for each example (**x**, **y**) in *examples* **do** //輸入某個訓練資料
    /* *Propagate the inputs forward to compute the outputs* */
    for each node i in the input layer **do**
      $a_i \leftarrow x_i$ //讀取輸入層資料
    for $\ell = 2$ **to** L **do**
      for each node j in layer ℓ **do** //計算中間層資料，
        $in_j \leftarrow \sum_i w_{i,j}\, a_i$ // 正向傳遞預測的值
        $a_j \leftarrow g(in_j)$
    /* *Propagate deltas backward from output layer to input layer* */
    for each node j in the output layer **do** //計算輸出層誤差
      $\Delta[j] \leftarrow g'(in_j) \times (y_j - a_j)$
    for $\ell = L-1$ **to** 1 **do**
      for each node i in layer ℓ **do** //由後往前計算中間層誤差，**backpropate errors**
        $\Delta[i] \leftarrow g'(in_i) \sum_j w_{i,j}\, \Delta[j]$ //反向傳遞誤差的值
    /* *Update every weight in network using deltas* */ // $a_j = g(\sum_{i=0}^{n} w_{i,j} a_i)$
    for each weight $w_{i,j}$ in *network* **do** //更新網路中的每一個權重，α是**learning rate**
      $w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta[j]$
  until some stopping criterion is satisfied //結束條件：總誤差夠小、權值變化很小、執行逾時。
  return *network*



Ch22-19

TensorFlow Playground體驗操練



http://playground.tensorflow.org/

20

Homework 4: Explore TensorFlow Playground

due：2023-06-25, 2 points penalty for each late day (within 3 days)

In this homework, you will experiment playing the Tensorflow Playground (http://playground.tensorflow.org). Please submit your homework report files in PDF format by logging onto your Moodle account.

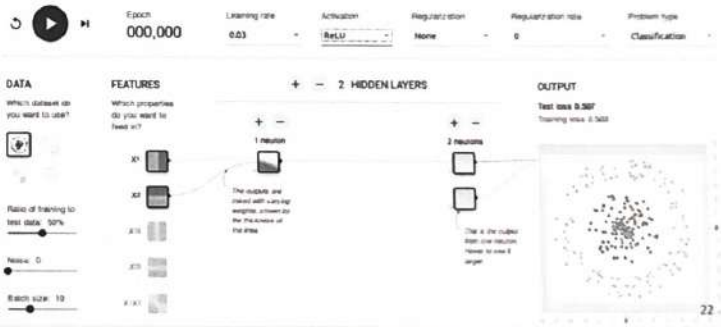
For each of the following questions, you will need to submit a full screenshot including the network configuration, test loss, and your parameter settings (including the DATA column on the left). Please also give the run time for getting the results. For each part of this question, you will be asked to classify a different dataset (available in the DATA column).

The following figure is a sample screenshot similar to what you are going to submit in your writeup. After you configure your network and parameter settings, you will use the "play" button to train the neural network and check whether the test loss can drop below the 0.1 threshold.

21

Homework 4: Explore TensorFlow Playground

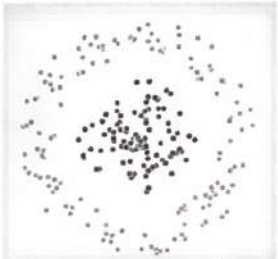
You can change the network structure (number of layers, and number of neurons in each layer), and adjust the following hyperparameters: learning rate, activation, regularization type and regularization rate. Please leave other parameters fixed: leave ratio of training to test data to 50%, leave Noise at 0, and Batch size at 10.



22

Homework 4: Explore TensorFlow Playground

(A) Circle data set: Can you obtain a test loss of less than 0.1 using only the raw input features? i.e., using only X_1 and X_2 as features? Please share a screenshot of your network as described.



Now suppose you use features X_1^2 and X_2^2, instead of raw features X_1 and X_2. Your objective now is to obtain a test loss of less than 0.1 in as few neurons as possible. How many neurons would you need to classify the dataset? Explain and justify your answer.

23

Homework 4: Explore TensorFlow Playground

(B) Exclusive-Or data set: Consider the following method to train your neural network model with raw features X_1, X_2. Set the activation function to be tanh, use a learning rate of 0.03 and don't use any regularization. Use a neural network with two hidden layers with the first hidden layer having 3 neurons and the second layer having 2 neurons. Do you obtain a test loss of less than 0.1?

Now change the learning rate to 0:3 and report the final test loss. Repeat the same for learning rate = 3 now. How does the test loss change as you increase the learning rate? Briefly explain why.

Find another network architecture that also gives you a test loss of less than 0.1 on this dataset. Please share a screenshot of this network as described earlier.

24