

Day 73 Gradient Descent

梯度下降 Gradient Descent 簡介



出題教練

陳宇春

知識地圖 深度學習簡介

深度學習體驗 - 啟動函數與正規化

深度神經網路 Supervised Learning Deep Neural Network (DNN)

- 簡介 Introduction
- 套件介紹 Tools: Keras
- 組成概念 Concept
- 訓練技巧 Training Skill
- 應用案例 Application

卷積神經網路 Convolutional Neural Network (CNN)

- 簡介 introduction
- 套件練習 Practice with Keras
- 訓練技巧 Training Skill
- 電腦視覺 Computer Vision

深度學習組成概念 Concept of DNN

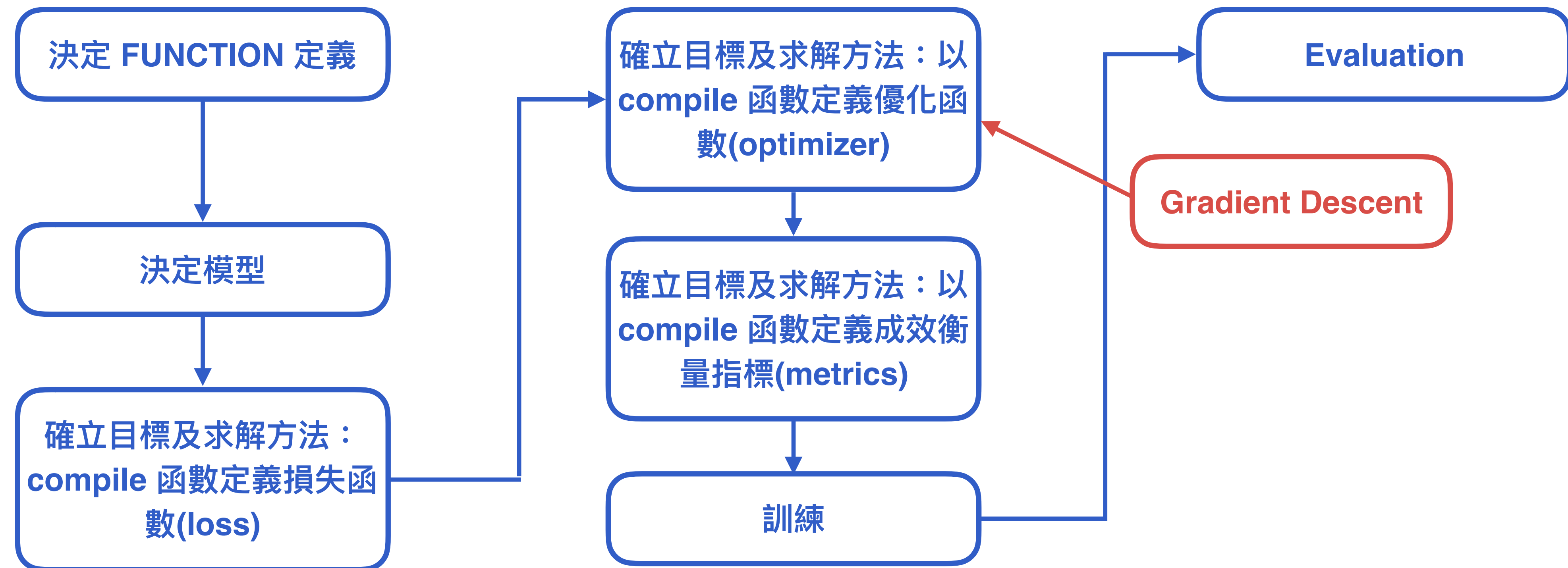
感知器概念簡介



本日知識點目標

- 了解梯度下降 (Gradient Descent) 的定義與程式樣貌
- 初步理解梯度下降 (Gradient Descent) 的概念
- 能從程式中微調 (Fine Tune) 相關的參數

梯度下降用在哪裡？



最常用的優化算法 - 梯度下降

- 機器學習算法當中，優化算法的功能，是通過改善訓練方式，來最小化(或最大化)損失函數
- 最常用的優化算法是梯度下降
 - 通過尋找最小值，控制方差，更新模型參數，最終使模型收斂
 - $w_{i+1} = w_i - d_i \cdot \eta_i$, $i=0,1,\dots$
 - 參數 η 是學習率。這個參數既可以設置為固定值，也可以用一維優化方法沿著訓練的方向逐步更新計算
 - 參數的更新分為兩步：第一步計算梯度下降的方向，第二步計算合適的學習

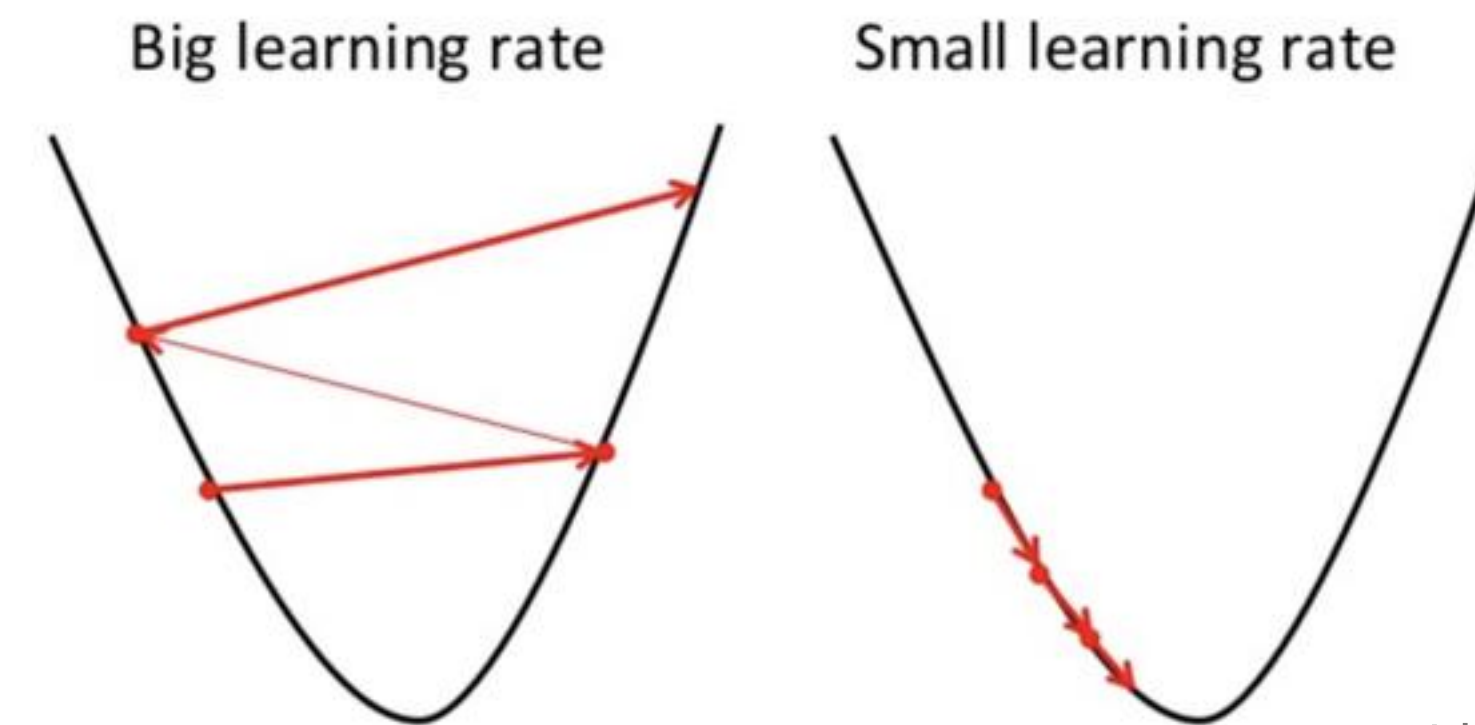
學習率對梯度下降的影響

- 學習率定義了每次疊代中應該更改的參數量。換句話說，它控制我們應該收斂到最低的速度或速度。
- 小學習率可以使迭代收斂，大學習率可能超過最小值

Compute $\partial L / \partial w$

$$w \leftarrow w - \eta \partial L / \partial w$$

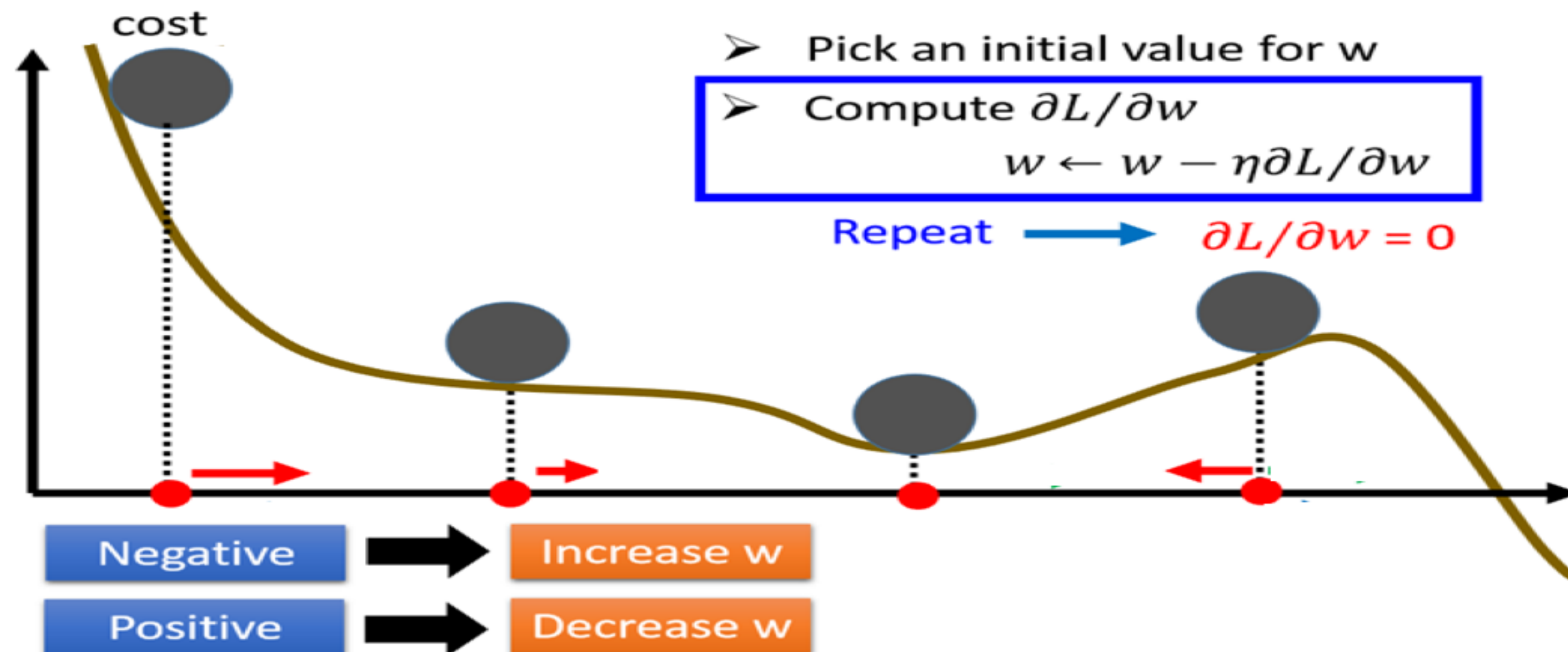
Learning rate
學習率



圖片來源：kknews.cc

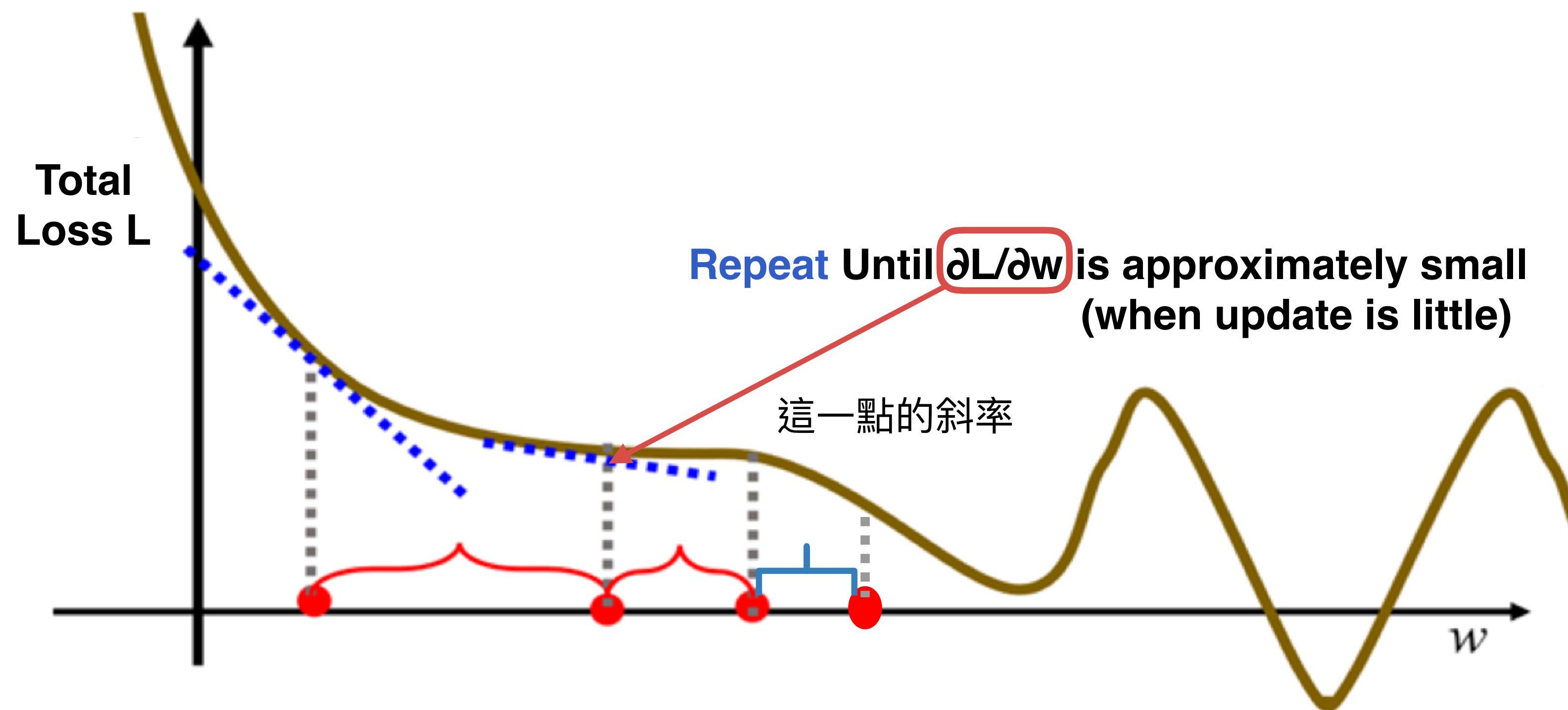
梯度下降法的過程

- 首先需要設定一個初始參數值，通常情況下將初值設為零($w=0$)，接下來需要計算成本函數 cost
- 然後計算函數的導數-某個點處的斜率值，並設定學習效率參數(lr)的值。
- 重複執行上述過程，直到參數值收斂，這樣我們就能獲得函數的最優解



怎麼確定到極值點了呢？

- η 又稱學習率，是一個挪動步長的基數， $df(x)/dx$ 是導函數，當離得遠的時候導數大，移動的就快，當接近極值時，導數非常小，移動的就非常小，防止跨過極值點

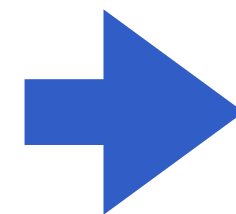


怎麼確定到極值點了呢？ (II)

- Gradient descent never guarantee global minima
- Different initial point will be caused reach different minima, so different results

But

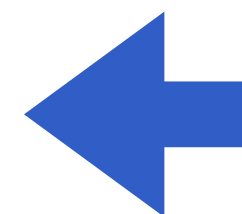
avoid local minima



Popular & Simple Idea: Reduce the learning rate by some factor every few epochs

在訓練神經網絡的時候，通常在訓練剛開始的時候使用較大的 **learning rate**，隨著訓練的進行，我們會慢慢的減小 **learning rate**

參數	意義
decayed_learning_rate	衰減後的學習率
learning_rate	初始學習率
decay_rate	衰減率
global_step	當前的 step
decay_steps	衰減週期



具體就是每次迭代的時候減少學習率的大小，更新公式：

$\text{decayed_learning_rate} = \text{learning_rate}^*$

$\text{decay_rate} ^ (\text{global_step}/\text{decay_steps})$

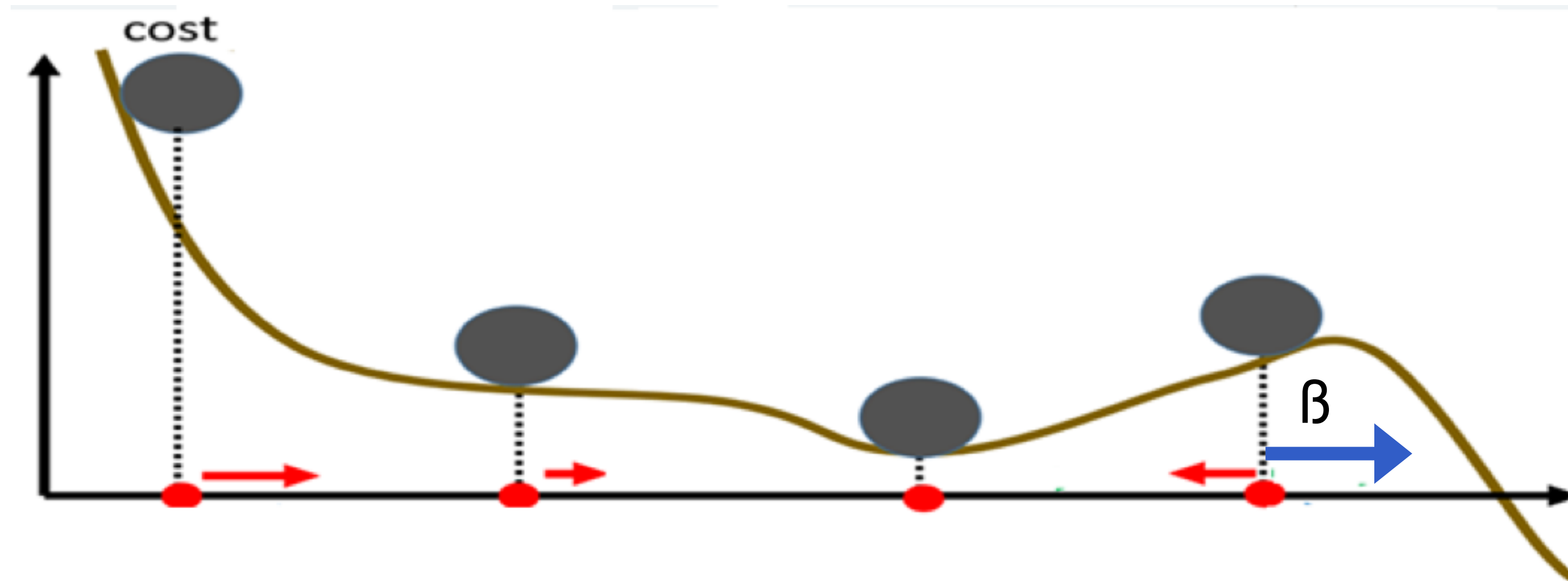
怎麼確定到極值點了呢？(III)

- 使用 momentum 是梯度下降法中一種常用的加速技術。
- Gradient Descent 的實現：SGD, 對於一般的 SGD，其表達式為

$$x \leftarrow x - \alpha * dx \text{ (x沿負梯度方向下降)}$$
- 而帶 momentum 項的 SGD 則寫生如下形式：

$$v = \beta * v - \alpha * dx$$

$$x \leftarrow x + v$$
- 其中 β 即 momentum 係數，通俗的理解上面式子就是，如果上一次的 momentum（即 β ）與這一次的負梯度方向是相同的，那這次下降的幅度就會加大，所以這樣做能夠達到加速收斂的過程



前述流程 / python程式 對照

● 前述流程

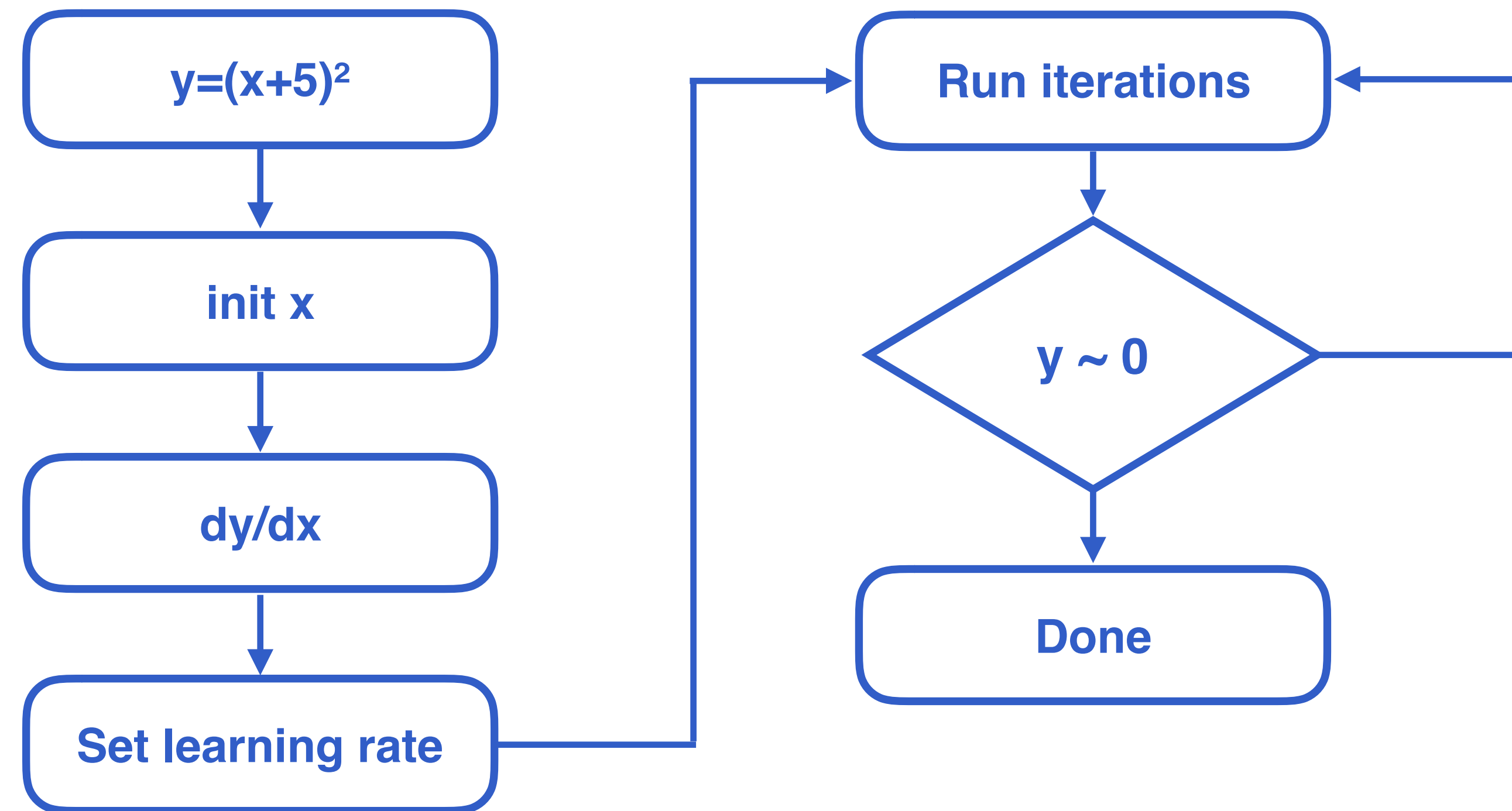
- Find the local minima of the function $y=(x+5)^2$ starting from the point $x=3$
- Step 1 : Initialize $x = 3$. Then, find the gradient of the function, $dy/dx = 2*(x+5)$.
- Step 2 : Move in the direction of the negative of the gradient, and, we use a learning rate.
Let us assume the learning rate $\rightarrow 0.01$
- Step 3 : Let's perform 2 iterations of gradient descent

Question :

We can observe that the X value is slowly decreasing and should converge to -5 (the local minima, $y=(-5+5)^2 = 0$). However, how many iterations should we perform?

前述流程 / python程式 對照

前述流程



前述流程 / python程式 對照

python 程式 (請參閱今日範例)

```
▶ In [1]: cur_x = 3 # The algorithm starts at x=3
          lr = 0.01 # Learning rate
          precision = 0.000001 #This tells us when to stop the algorithm
          previous_step_size = 1 #
          max_iters = 10000 # maximum number of iterations
          iters = 0 #iteration counter
          df = lambda x: 2*(x+5) #Gradient of our function

          iters_history = [iters]
          x_history = [cur_x]
```

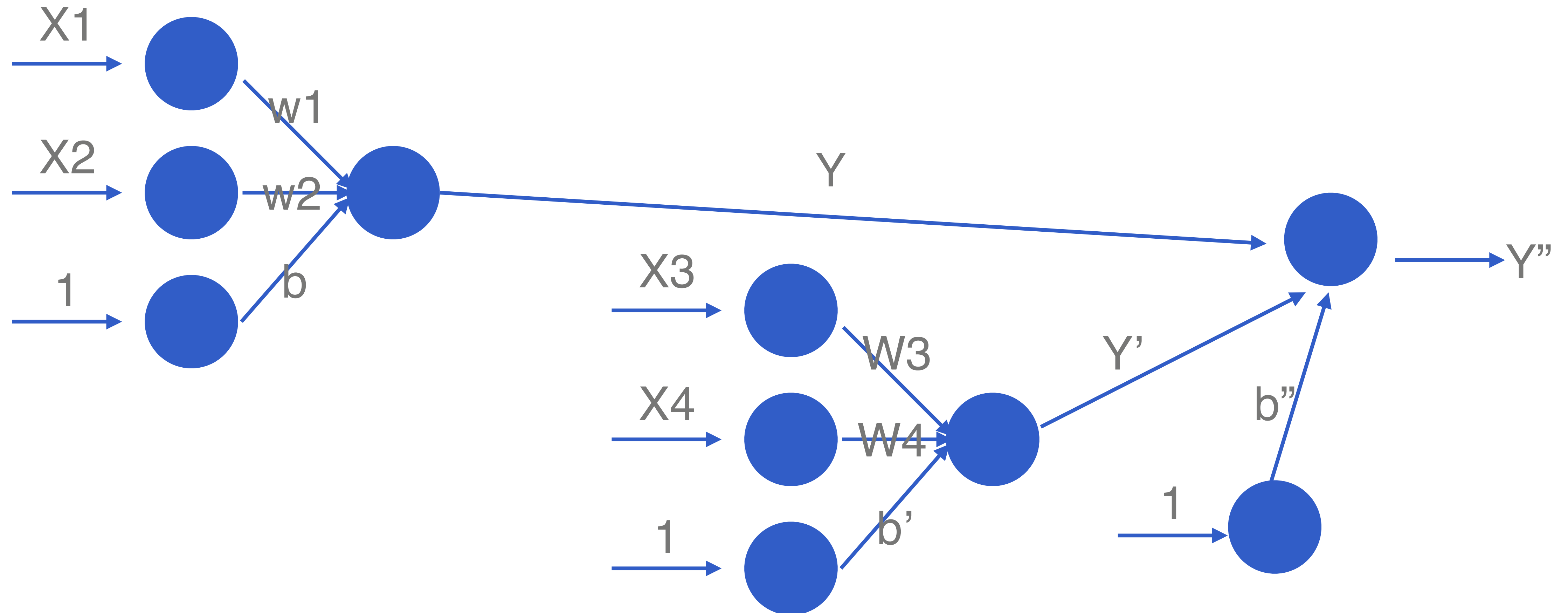
```
In [2]: while previous_step_size > precision and iters < max_iters:
        prev_x = cur_x #Store current x value in prev_x
        cur_x = cur_x - lr * df(prev_x) #Gradient descent
        previous_step_size = abs(cur_x - prev_x) # 取較大的值, Change in x
        iters = iters+1 #iteration count
        print("Iteration",iters,"\nX value is",cur_x) #Print iterations
        # Store parameters for plotting
        iters_history.append(iters)
        x_history.append(cur_x)
```

重要知識點複習：梯度下降法 (Gradient descent)

- Gradient descent 是一個一階最佳化算法，通常也稱為最速下降法。
- 要使用梯度下降法找到一個函數的局部極小值，必須向函數上當前點對應梯度（或者是近似梯度）的反方向的規定步長距離點進行疊代搜索。
- 梯度下降法的缺點包括：
 - 靠近極小值時速度減慢。
 - 直線搜索可能會產生一些問題。
 - 可能會「之字型」地下降
- avoid local minima
 - 在訓練神經網絡的時候，通常在訓練剛開始的時候使用較大的 **learning rate**，隨著訓練的進行，我們會慢慢的減小 **learning rate**

複習：網路參數(Network Parameter)

權值 (w_1, w_2, w_3, w_4) 與 Bias (b, b', b'') 都要被更新



解題時間 It's Your Turn

請跳出PDF至官網Sample Code & 作業
開始解題

