巨匠線上真人

Python 資料科學應用開發

第十一堂：感知器與適應線性神經元（Perceptron and Adaline ）

同學，歡迎你參加本課程

- 請關閉你的FB 、Line等溝通工具，以免影響你上課。
- 考量頻寬、雜音，請預設關閉攝影機、麥克風，若有需要再打開。
- 隨時準備好，老師會呼叫你的名字進行互動，鼓勵用麥克風提問。
- 如果有緊急事情，你必需離開線上教室，請用聊天室私訊給老師，以免老師癡癡呼喚你的名字。
- 軟體安裝請在上課前安裝完成，未完成的同學，請盡快進行安裝。

課程檔案下載


巨匠電腦線上真人
開課查詢
免費體驗專區
課程總覽
專業師資
學員專區
講師專區
最新消息
您好!
登出
點數卡產品兌換
APCS檢測專區
公告專區
我的課表
IT真人課程劃位
電腦分校課程劃位
外語真人課程劃
美語分校課程劃
取消劃位
課程檔案下載
上課權益查詢
教學平台測試
學習諮詢
常見問題
個資維護
忘記密碼
登出
課程檔案下載
程式語言好難學?
那是因為
你還沒學過Python!
線上老師LIVE直播教學．搶先看
巨匠電腦真人課程

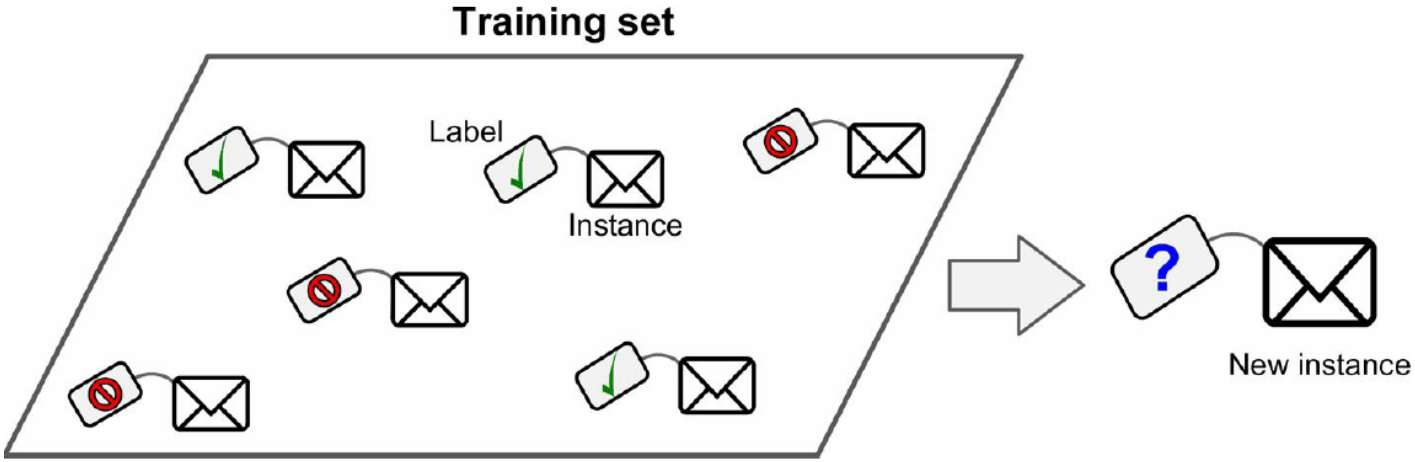
ZOOM 學員操作說明


www.pcschool.com.tw
您正在觀看蘇世宗的螢幕
查看選項
原始大小
請求遠端控制
共同註記
退出全螢幕
5 查看選項/共同註記/筆（連連看）
滑鼠
文字
筆
箭頭
擦除
格式
撤銷
重做
清除
與會者 (15)
查找參會者
張齡月 (我)
婷婷
解除靜音
舉手
2 共享螢幕（指導演練；點評作品）
老師須先停止共享螢幕
才能請學生共享螢幕
1 聊天
解除靜音
啟動視訊
邀請
與會者
共享螢幕
聊天
錄影
4 解除靜音
3 與會者/舉手

監督式學習

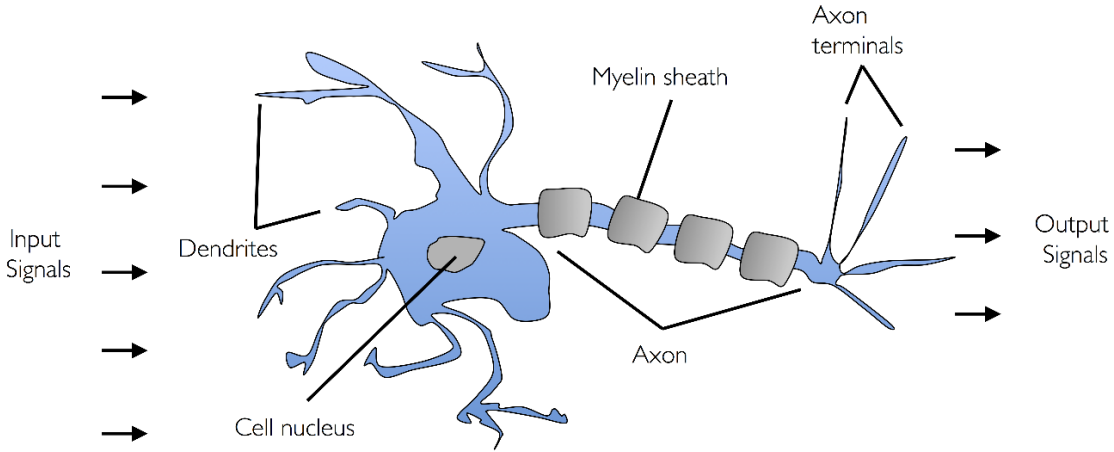
- 監督式學習（Supervised learning）分為兩類
  - regression：predict results within a continuous output, meaning that we are trying to map input variables to some continuous function.
  - classification：predict results in a discrete output. In other words, we are trying to map input variables into discrete categories.

分類（Classification）



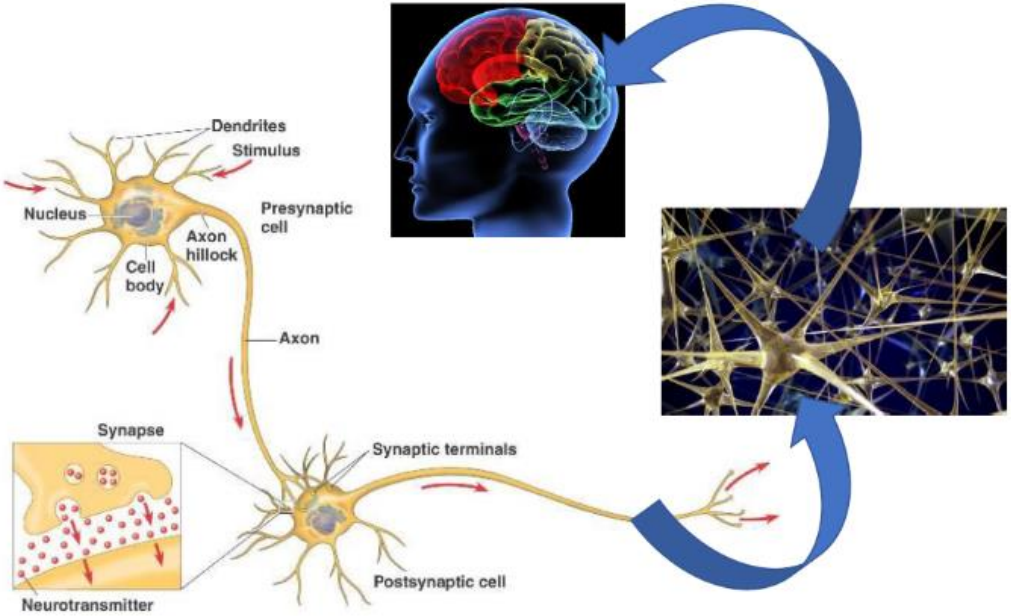
神經元（Neuron）

- Simple logic gate with binary outputs
- Signals arrive at dendrites（樹突）
- Integrated into cell body
- If signal exceeds threshold, generate output, and pass to axon（軸突）

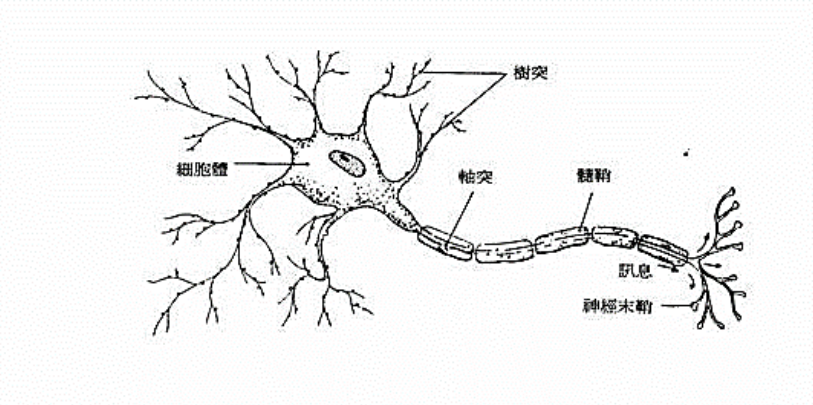


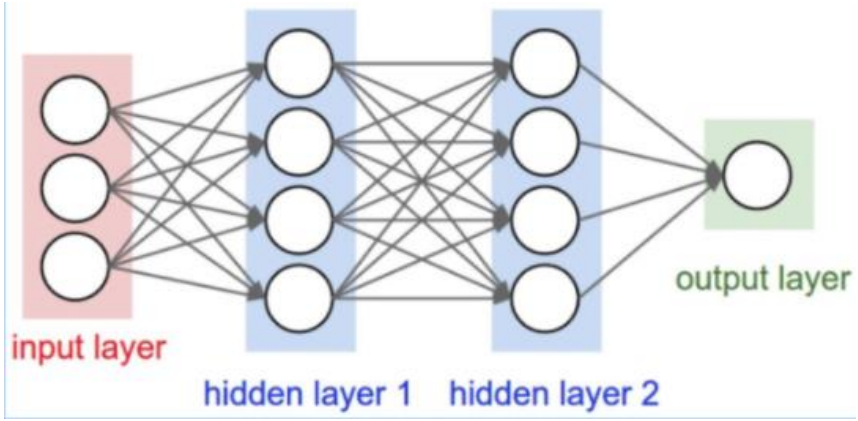
生物神經系統

- 經過層層的傳導，最後傳至大腦，作出反應



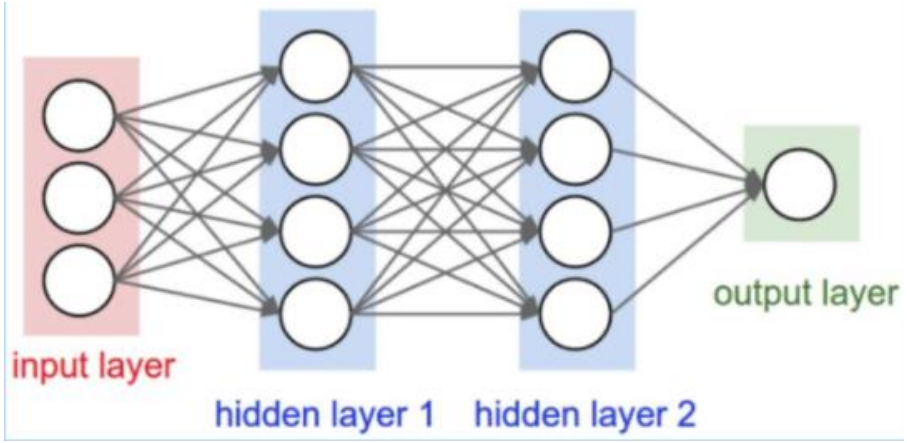
模擬神經系統


樹突
細胞體
軸突
髓鞘
訊息
神經末鞘


input layer
hidden layer 1
hidden layer 2
output layer

Neural Network vs. Deep learning

- Neural Network：模擬生物神經系統運作原理的演算法
- Deep learning：hidden layers 超過2層的 Neural Network



Rosenblatt 感知器（Perceptron）

- Binary classification task
  - Positive class (1) vs. negative class (-1)
- Define activation function Φ(z)
  - Takes as input a dot product of input and weights
  - Net input: $z = w_1x_1 + \ldots + w_mx_m$

$$\mathbf{w} = \begin{bmatrix} w^{(1)} \\ w^{(2)} \\ \vdots \\ w^{(m)} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \end{bmatrix}$$

單位階梯函數（Unit Step Function）

- Φ(z)：activation function，超過門檻值θ（Threshold）就輸出1，否則就輸出-1

$$\phi(z) = \begin{cases} 1 & \text{if } z \geq \theta \\ -1 & \text{otherwise} \end{cases}$$

- 令 $w_0=\theta$，$x_0=1$

$$z = w_0x_0 + w_1x_1 + \cdots + w_mx_m = \mathbf{w}^T\mathbf{x}$$

$$\phi(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

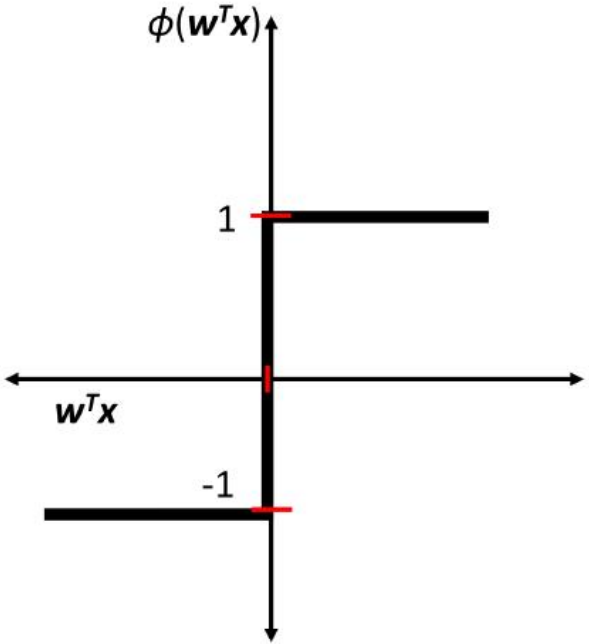
基本線性代數

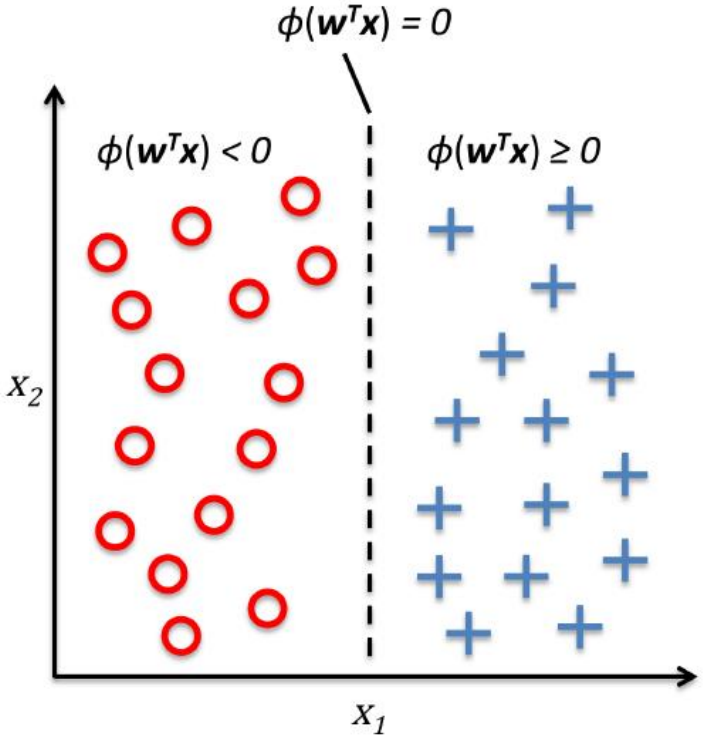
Vector dot product

$$z = \mathbf{w}^T\mathbf{x} = \sum_{j=0}^{m} \mathbf{w}_j\mathbf{x}_j$$

$$\begin{bmatrix}1 & 2 & 3\end{bmatrix} \times \begin{bmatrix}4\\5\\6\end{bmatrix} = 1 \times 4 + 2 \times 5 + 3 \times 6 = 32.$$

二分類輸出（binary output）





感知器（Perceptron）演算法

1. 設定權重（Weights）初始值為0，或是很小的隨機亂數
2. 針對訓練樣本X，反覆訓練：
   1) 計算預估值$\hat{y}$
   2) 更新權重（Weights）

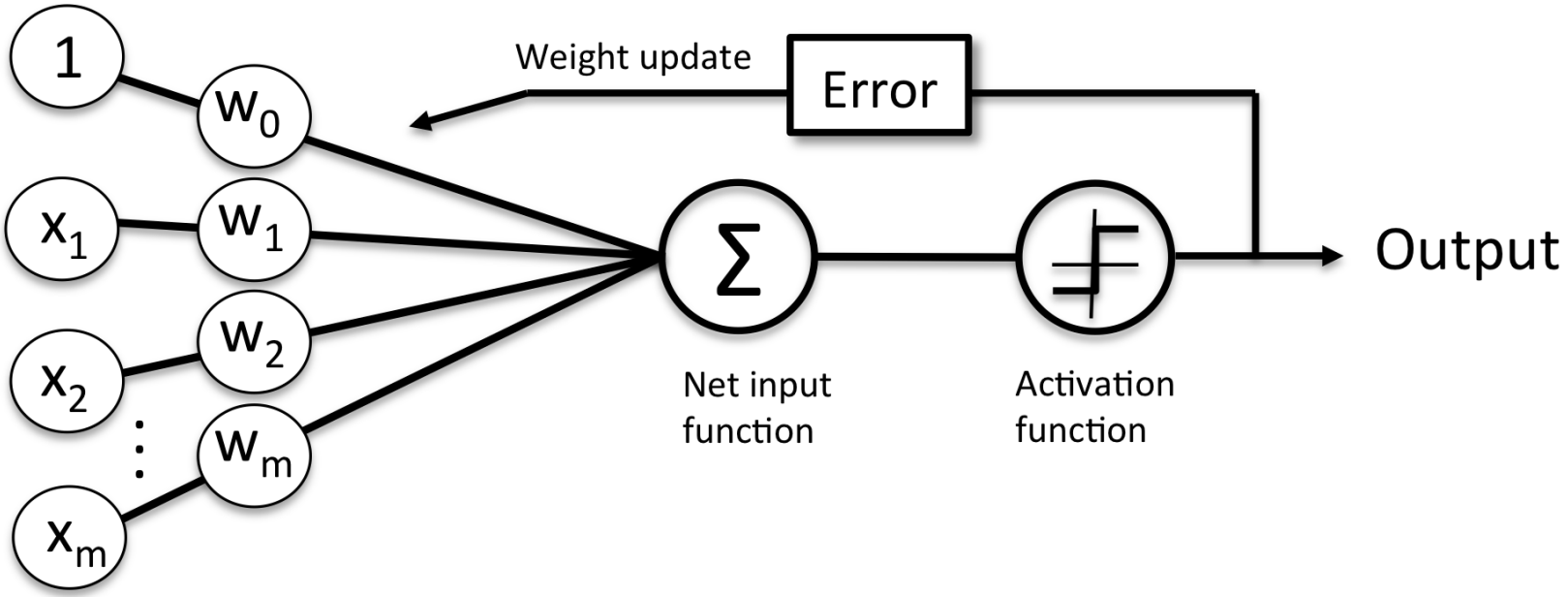
$$w_j := w_j + \Delta w_j$$

$$\Delta w_j = \eta\left(y^{(i)} - \hat{y}^{(i)}\right)x_j^{(i)}$$

η 是學習率（learning rate），通常是常數介於 (0, 1) 之間

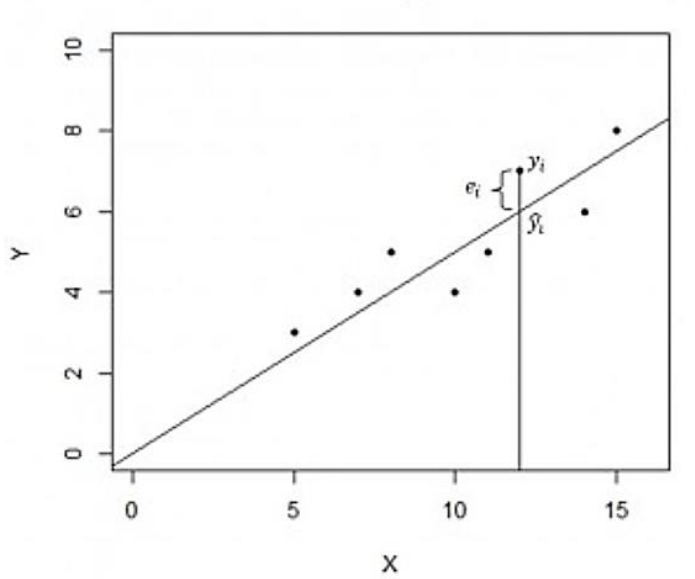
   3) 直到收斂（Convergence）為止，或達到事先設定的最大訓練週期為止

圖解

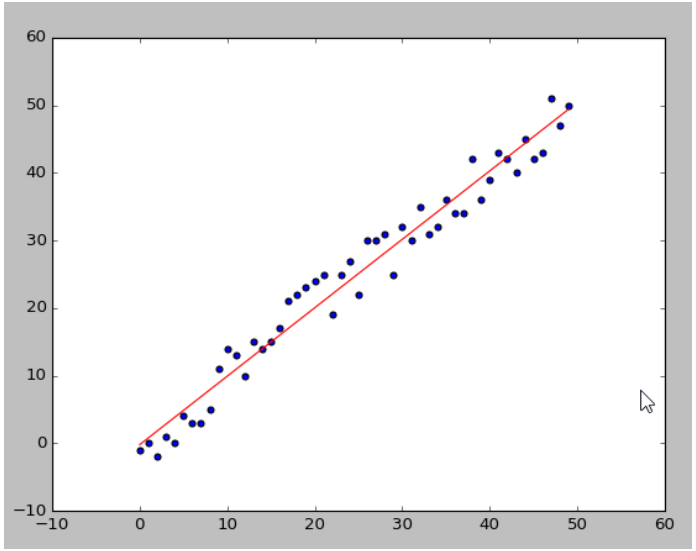

1
w_0
x_1
w_1
x_2
w_2
w_m
x_m
Weight update
Error
Σ
Output
Net input function
Activation function

線性迴歸（Linear Regression）

y = x * w + b + ε



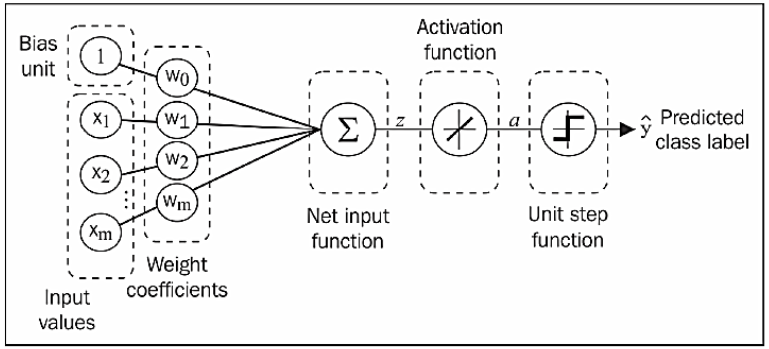
linearRegress.py

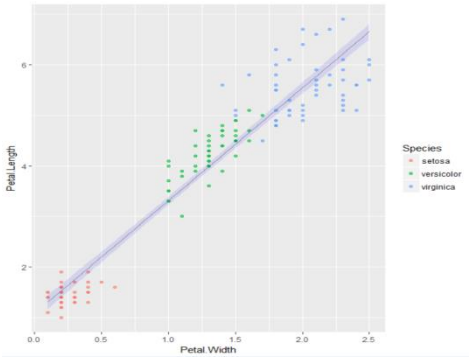


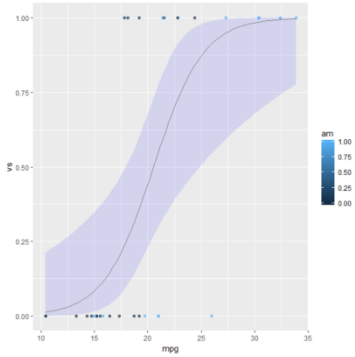
簡單感知器（Perceptron）模型

- y = x * w + b
- 根據output決定是否往下一層傳送，設一門檻（threshold)，以一個Activation function來處理，例如：g = Sigmoid

➔ $y = g(x * w + b) = \frac{1}{1+e^{-(x*w+b)}}$

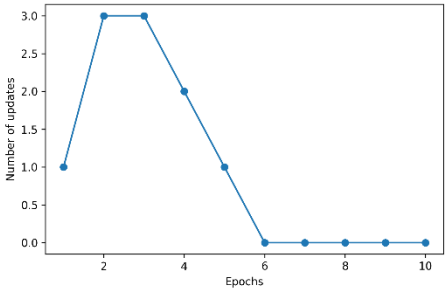




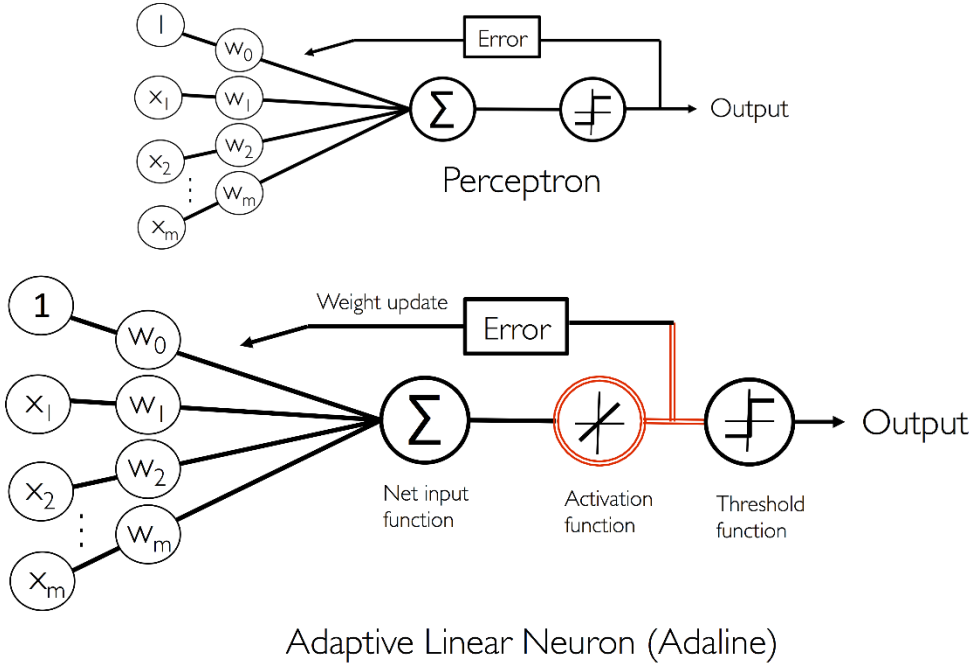


感知器（Perceptron）實作

- 程式碼：ch02.ipynb
  - Perceptron class
  - 呼叫並作圖
    - ppn = Perceptron(eta=0.1, n_iter=10)
    - ppn.fit(X, y)
    - plt.plot(range(1, len(ppn.errors_) + 1), ppn.errors_, marker='o')
    - plt.xlabel('Epochs')
    - plt.ylabel('Number of updates')
    - # plt.savefig('images/02_07.png', dpi=300)
    - plt.show()



適應線性神經元（adaptive linear neuron）求解



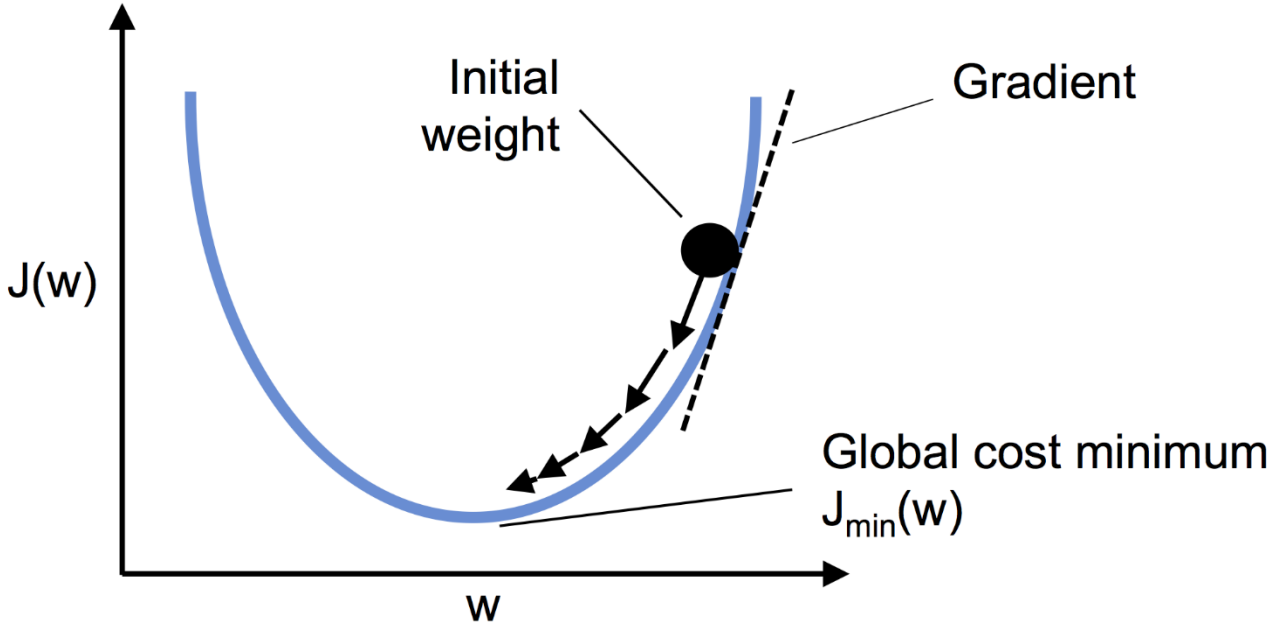
Adaline cost function

- 更新權重（Weights）改為

$$J(\mathbf{w}) = \frac{1}{2}\sum_{i}\left(y^{(i)} - \phi\left(z^{(i)}\right)\right)^2$$

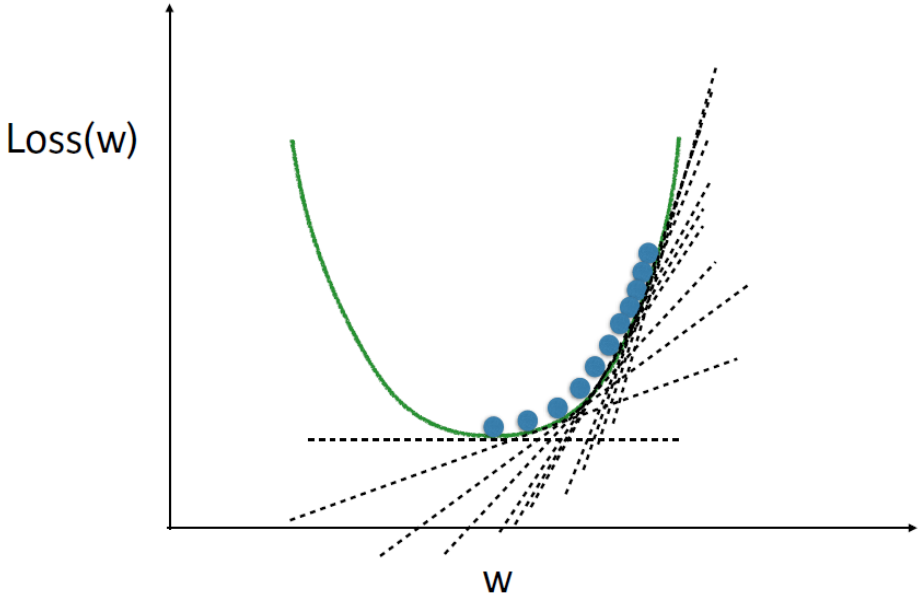
- The linear activation function is differentiable（可微分）
- Cost function is convex（凸函數）
- Can use gradient descent（梯度下降）to learn the weights

梯度下降（Gradient Descent）



逐步逼近最佳解

Gradient descent



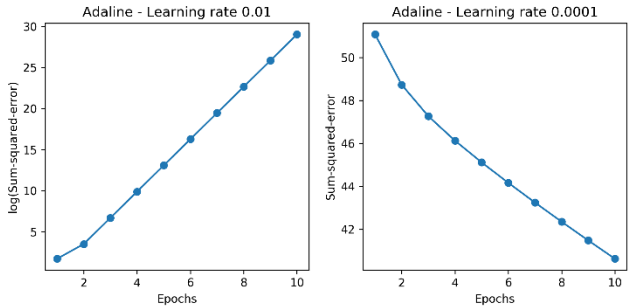
適應線性神經元（Adaline）實作

- 程式碼：ch02.ipynb
  - AdalineGD class
  - 呼叫並作圖
    - fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(10, 4))

    - ada1 = AdalineGD(n_iter=10, eta=0.01).fit(X, y)
    - ax[0].plot(range(1, len(ada1.cost_) + 1), np.log10(ada1.cost_), marker='o')
    - ax[0].set_xlabel('Epochs')
    - ax[0].set_ylabel('log(Sum-squared-error)')
    - ax[0].set_title('Adaline - Learning rate 0.01')

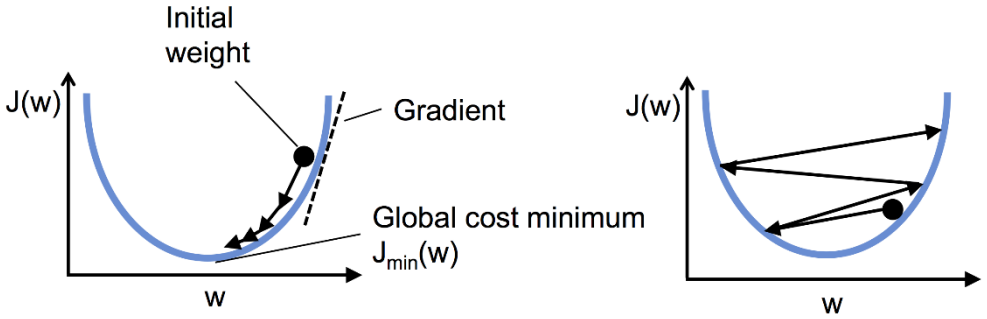
    - ada2 = AdalineGD(n_iter=10, eta=0.0001).fit(X, y)
    - ax[1].plot(range(1, len(ada2.cost_) + 1), ada2.cost_, marker='o')
    - ax[1].set_xlabel('Epochs')
    - ax[1].set_ylabel('Sum-squared-error')
    - ax[1].set_title('Adaline - Learning rate 0.0001')

    - # plt.savefig('images/02_11.png', dpi=300)
    - plt.show()



範例練習

- 程式碼：gd1.py
  - 最小化成本函數（Cost Function），或稱目標函數（Target Function）或損失函數（Loss Function），此例成本函數為 x^2（第6行）。
  - 一階偏微分為 2x（第9行）。
- 測試─調整下列參數：
  - 起始值（第32行）
  - 優化週期數（第34行）
  - 學習率（第36行）



Quiz (1)

- 起始值（第32行）改為 -10
  1) 最佳解仍趨近於 0
  2) 最佳解不趨近於 0

Quiz (2)

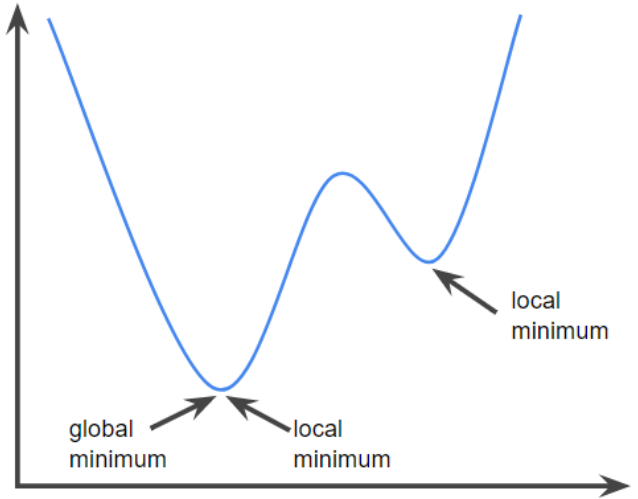
- 學習率（第36行）改為 0.03
  1) 最佳解仍趨近於 0
  2) 最佳解若不趨近於 0，其值為何？

Quiz (3)

- 學習率（第36行）改為 0.9
  1) 最佳解仍趨近於 0
  2) 最佳解若不趨近於 0
- 求解過程會呈現何種狀況？

全局最佳解 vs. 區域最佳解

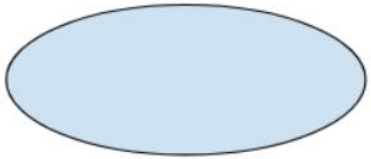
- 目標函數必須是凸函數（Convex Function），才能保證可求得全局最佳解（Global Cost Minimum）



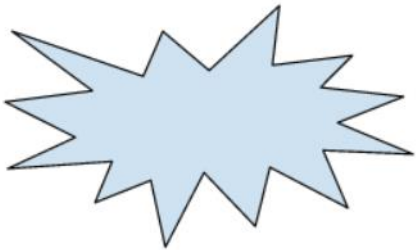
凸函數（Convex Function）

convex set

A subset of Euclidean space such that a line drawn between any two points in the subset remains con
subset. For instance, the following two shapes are convex sets:



By contrast, the following two shapes are not convex sets:



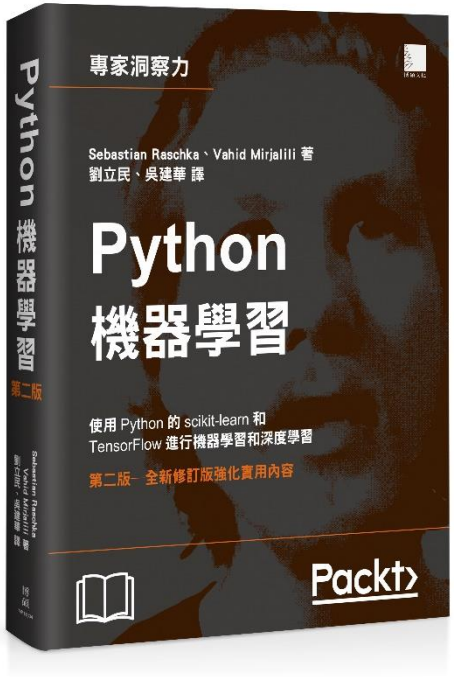
作品：以梯度下降法訓練模型

- 資料集：鳶尾花（Iris）
- data = datasets.load_iris()
- 函數：AdalineSGD
  - ch02.ipynb 第 23 格
- 使用
  - ch02.ipynb 第 24 格

展示

Iris Versicolor　Iris Setosa　Iris Virginica

參考用書



- 書名：Python機器學習（第二版）

  http://www.drmaster.com.tw/bookinfo.asp?BookID=MP11804
- 作者：Sebastian Raschka, Vahid Mirjalili ISBN
- 譯者：劉立民、吳建華
- 出版社：博碩

問卷

http://www.pcschoolonline.com.tw

www.pcschoolonline.com.tw