

深度學習

Lab Assingment 1

1. 目的

- 撰寫感知器學習演算法(Perceptron Learning Algorithm; PLA)
- 熟悉 Python 語法及相關函式庫(如：numpy，matplotlib 等)

2. 課程章節與實驗環境

2.1. 課程章節

Week 2: Neural Networks Basics 中 Perceptron Learning Algorithm

2.2. 實驗環境

限定使用 Python 程式語言

3. 實驗進行步驟及結果

3.1. 自行撰寫 PLA 程式，不可使用套裝軟體實現程式

使用 Python 相關函式庫畫圖(如：matplotlib，此部分最多加 20 分)

3.2. 實驗：

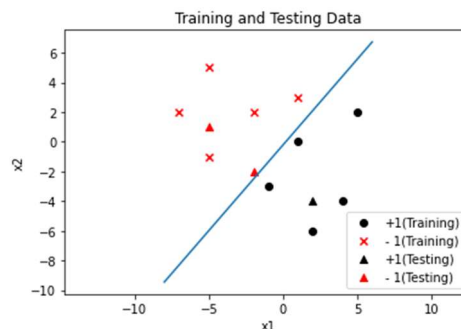
(a) 實驗一

- 給定以下訓練／測試資料：

Example	x_1	x_2	Class
1	1	0	1
2	1	3	-1
3	2	-6	1
4	-1	-3	1
5	-5	5	-1
6	5	2	1
7	-2	2	-1
8	-7	2	-1
9	4	-4	1
10	-5	-1	-1

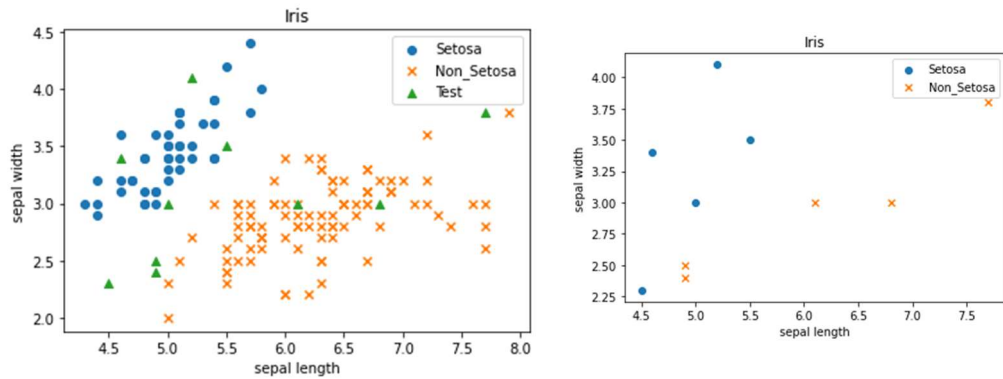
Example	x_1	x_2
1	2	-4
2	-5	1
3	-2	-2

- 輸出分類線性方程式 $w_1x_1 + w_2x_2 + b = 0$ 之 w_1, w_2, b 值
- 預測 3 筆測試資料之類別
- 畫圖(加分)



(b) 實驗二

- 使用讀檔方式讀取訓練資料(Iris_training.txt)，測試資料(Iris_test.txt)
- Iris_training.txt 含 45 筆 Setosa 蘭花 及 95 筆 Non_Setosa 蘭花
- Iris_test.txt 各含 5 筆 Setosa 蘭花 及 Non_Setosa 蘭花
- 下圖顯示訓練／測試資料形分佈情形：



- 輸出分類線性方程式 $w_1x_1 + w_2x_2 + b = 0$ 之 w_1, w_2, b 值
- 計算測試準確率(預測需以分類線性方程式為基準，本次作業準確率高低不影響作業分數)
- 如實驗一畫圖(加分)

4. 說明

4.1. PLA 演算法

假設每筆訓練資料為 (\mathbf{x}, y) ，則 $\mathbf{x} = (x_1, x_2)$ ，而 $y = \pm 1$ 。

令 \hat{y} 為訓練資料估算值， $\mathbf{w} = (w_1, w_2)$ ，

則 $\hat{y} = \text{sign}(w_1x_1 + w_2x_2 + b) = \text{sign}(\mathbf{w} \bullet \mathbf{x} + b) = \pm 1$ 。

下表為 PLA 演算法：

Given a training data set:

$D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ // N training examples

Initialize the weight vector $\mathbf{w} = (w_1, w_2)$, and the bias b to random values

UNTIL one of the termination conditions is met, DO

FOR each training example $\mathbf{x} \in D$

If $\text{sign}(\mathbf{w} \bullet \mathbf{x} + b) \neq y$ then

$\mathbf{w} \leftarrow \mathbf{w} + y \mathbf{x}$ // \mathbf{w} and \mathbf{x} are vectors, and y is a scalar

$b \leftarrow b + y$ // b and y are scalars

4.2. $w_1x_1 + w_2x_2$ 之計算

可用內積(Inner Product)或一般矩陣乘法計算：

- 內積 $w_1x_1 + w_2x_2 = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{w} \bullet \mathbf{x}$

- 一般矩陣乘法 $w_1x_1 + w_2x_2 = \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{w}^T \mathbf{x}$

4.3. PLA 停止條件

- 若是線性可分離(Linearly Separable)訓練資料，PLA 停止條件為所有訓練資料在某一世代(Epoch)均分類正確。
- 演算法最好還是給定最大世代(Epoch)數強迫停止，以免訓練資料為非線性可分離(Linearly Inseparable)時，致使 PLA 造成無窮迴圈，無法停止。

4.4. PLA 演算法變型，可具有學習率(Learning Rate)：

$$\mathbf{w} \leftarrow \mathbf{w} + \eta y \mathbf{x} \quad // \mathbf{w} \text{ and } \mathbf{x} \text{ are vectors ; } y, \text{ and } \eta \text{ are scalars}$$

$$b \leftarrow b + \eta y \quad // b, y, \text{ and } \eta \text{ are scalars}$$

學習率 η 之目的，可控制 \mathbf{w} 及 b 調整更新幅度大小，其值一般為：

$$0 < \eta < 1。$$