

通訊所 碩一 107064522 李曼鈴

Computer Assignment 4-2

- 語言格式: Matlab (R2018a)
- 程式流程
 1. 讀入 dataset
 2. 將每筆 examples 分類 label
 3. 實作 perceptron learning
- 程式說明與截圖輸出
 1. 首先將指定的 dataset 讀入，存到 20*6 的矩陣 **data** 之中，其中 20 表示 20 個 examples，6 表示 1 個 w_0 weight(預設為 1)，以及 5 個 attributes 的 weights。

```

9      %%%% 4.2 %%%%
10 -   fprintf('Computer Assignment 4.2\n\n')
11      %Reading the dataset from hw3_dataset.txt
12 -   [ex, val] = textread('hw3_dataset.txt', '%s %s');
13 -   data = ones(20, 6);
14 -   for i=1:20
15 -       for j=1:5
16 -           data(i, j+1) = str2double(val{i}{j});
17 -       end
18 -   end

```

2. 依照題目要求將每筆 example 做分類，並存於 **c(x)** 中

```

20      %Using c for storing the labels
21 -   c = zeros(20, 1);
22 -   data_sum = sum(data, 2); %sum each row
23 -   for i = 1:20
24 -       if data_sum(i, 1) >= 4
25 -           c(i, 1) = 1;
26 -       else
27 -           c(i, 1) = 0;
28 -       end
29 -   end

```

3. 設定 learning rate 的值(**eta**)，以及等等要儲存 epoch 次數的 **epoch**。

```

31      %Setting the learning rate
32 -   eta = [0.2 0.4 0.6 0.8];
33 -   epoch = zeros(4, 1);

```

4. 接下來在 **for** 迴圈中實現 perceptron learning:

- (1) **For n = 1:4**，用來跑過四組 learning rate，迴圈中首先設定一些等等會用到的參數，例如 **classifier** 用來儲存一個 epoch 中 update 的 weights，會在記錄下一個 epoch 的 weights 時被洗掉。

```

36 -   for n = 1:4
37       %Setting the initial weights
38 -       w = [0.2 0.2 0.2 0.2 0.2 0.2];
39       %h(x) denotes the class returned by the classifier
40 -       h = zeros(20, 1);
41 -       err = zeros(20, 1); %Record c(x)-h(x)
42 -       classifier = zeros(20, 6);
43
44 -       epochs = 0; %number of example-presentations
45 -       hold = 1;

```

- (2) 執行 **while** 迴圈直到一個完整的 epoch 中，每筆 example 都正確分類。其中 line 49~53 在利用當前 classifier 做分類，line 54~59 檢查將分類錯誤的 classifier 進行修改(修改準則依上課所教的 perceptron learning)，並儲存於 **classifier** 矩陣之中。最後 line 62~66 檢查這個 epoch 中是否每筆資料都正確分類，如果沒有的話就將目前記錄 **epochs** 數加 1 繼續做 learning，反之則將 hold 設為 0 並且跳出迴圈。

要注意 line 49 classifier 做分類時式檢查是否 $> 1e-10$ 而不是 > 0 ，這是由於 matlab 在儲存數值並作計算時並不與我[平常計算相同，有可能輸出是一個很小的值(例如 $5e-17$)，但在我們手算時是 0，因此我設定一個稍微比 0 大的值，來避免上述的情形發生。

```

47 - while hold
48 -     for i = 1:20
49 -         if sum(data(i, :).*w, 2) > 1e-10
50 -             h(i, 1) = 1;
51 -         else
52 -             h(i, 1) = 0;
53 -         end
54 -         err(i, 1) = c(i, 1)-h(i, 1);
55 -         w = w+eta(n)*err(i, 1)*data(i,:);
56 -         %Recording new classifier
57 -         for j = 1:6
58 -             classifier(i, j) = w(j);
59 -         end
60 -     end
61 -     %Check if all of the data are correctly classified
62 -     if sum(err ~= zeros(20, 1)) ~= 0
63 -         epochs = epochs+1;
64 -     else
65 -         hold = 0;
66 -     end
67 - end

```

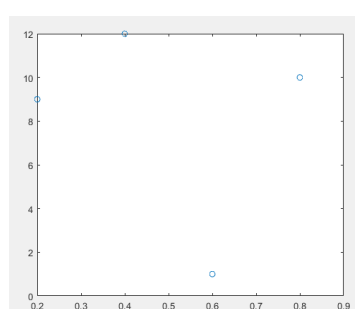
- (3) 成功找出 final classifier 之後跳出 **while** 迴圈並且記錄 **epochs** 值於 **epoch** 中，最後再 print 出結果，4 組 learning rate 都試完之後即跳出 **for** 迴圈。

```

68 - epoch(n, 1) = epochs;
69 - fprintf('For learning rate = %.1f : \n', eta(n))
70 - fprintf('\t%d epochs Done! \n', epochs)
71 - fprintf('\tFinal classifier : [w0, w1, w2, w3, w4, w5] = [%.1f, %.1f, %.1f, %.1f, %.1f, %.1f]\n\n', classifier(n, :))
72 - end

```

5. 最後以 learning rate 為 x 軸、epoch 為 y 軸，將 4 組結果標於圖上：



其中 y 軸為 epoch 數，若乘上 20 則表示 the number of example-presentations (the number of training epochs to achieve zero error rate times the number of examples in the training set)

6. Output 結果如下:

```

Computer Assignment 4.2

For learning rate = 0.2 :
  9 epochs Done!
  Final classifier : [w0, w1, w2, w3, w4, w5] = [-1.2, 0.6, 0.4, 0.6, 0.4, 0.6]

For learning rate = 0.4 :
  12 epochs Done!
  Final classifier : [w0, w1, w2, w3, w4, w5] = [-3.0, 1.4, 1.0, 1.4, 1.0, 1.4]

For learning rate = 0.6 :
  1 epochs Done!
  Final classifier : [w0, w1, w2, w3, w4, w5] = [-0.4, 0.2, 0.2, 0.2, 0.2, 0.2]

For learning rate = 0.8 :
  10 epochs Done!
  Final classifier : [w0, w1, w2, w3, w4, w5] = [-5.4, 2.6, 1.8, 2.6, 1.8, 2.6]

```

Computer Assignment 4-3

- 語言格式: Matlab (R2018a)
- 程式流程
 1. 讀入 dataset
 2. 將每筆 examples 分類 label
 3. 實作 perceptron learning
- 程式說明與截圖輸出
 1. 首先紀錄要隨機增加的 attributes 個數 **N**，以及建立一個 **sum_epoch**，由於每次跑出來的結果不一樣，因此我每組 **N** 都跑 1000 次再去取平均，取平均之前的 epoch 就儲存於 **sum_epoch** 之中。

```

75 %%% 4.3 %%%
76 fprintf('Computer Assignment 4.3\n\n')
77 N = [1 5 10 15 20];
78
79 sum_epoch = zeros(length(N), 1);

```

2. 利用 **for** 迴圈來執行 5 組不同的 **N**，**for k = 1:1000** 表示跑了 1000 次模擬，其中 line 88~91 隨機產生 0/1 的 attributes，並加於原 data 之後。

```

80 for n = 1:length(N)
81     for k = 1:1000
82         new_data = zeros(20, 6+N(n));
83         new_classifier = zeros(20, 6+N(n));
84         new_w = 0.2*ones(1, 6+N(n));
85         epochs = 0;
86         hold = 1;
87
88         for i = 1:20
89             temp = round(unifrnd(0,1,1,N(n)));
90             new_data(i, :) = [data(i, :) temp];
91         end

```

3. 其餘 perceptron learning 部分都與前一題施行方式相同，差別在於累計每次新算出的 epoch，總共累計 1000 次。

4. 印出增加不同 attributes 數量 N 時，平均 epoch 數，並且作圖

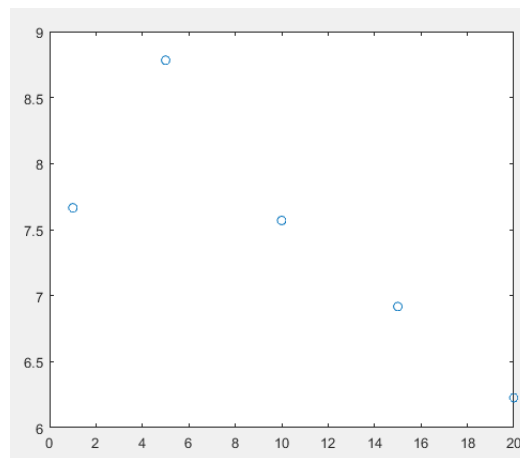
```

118 - for i = 1:length(N)
119 -     fprintf('With %d attributes added\n', N(i))
120 -     fprintf('\tAverage epochs : %.3f\n\n', sum_epoch(i, 1)/1000)
121 - end
122
123 - plot(N, sum_epoch(:, 1)/1000, 'o');

```

5. 最後輸出結果如下:

其中 y 軸為 epoch 數，若乘上 20 則表示 the number of example-presentations (the number of training epochs to achieve zero error rate times the number of examples in the training set)



Computer Assignment 4.3

With 1 attributes added
Average epochs : 7.665

With 5 attributes added
Average epochs : 8.783

With 10 attributes added
Average epochs : 7.569

With 15 attributes added
Average epochs : 6.917

With 20 attributes added
Average epochs : 6.227

6. 可以發現無論 N 值為多少，其 epoch 大都分布於 6-9 之間，可以知道 learning rate 的設定並非一味的增加或減少，而是根據不同的 data 會有不同適用的 learning rate。