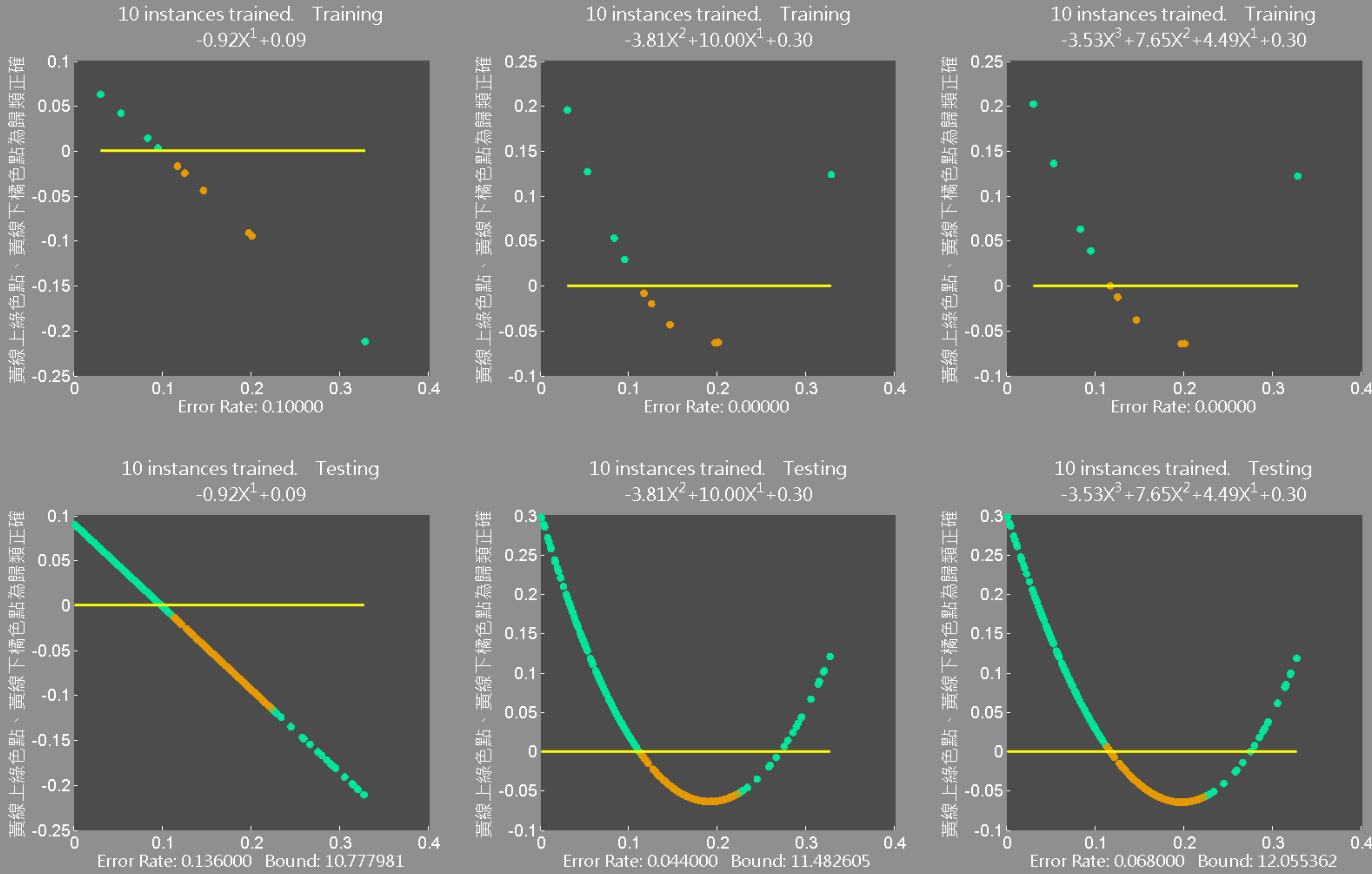
# 此份文件包含

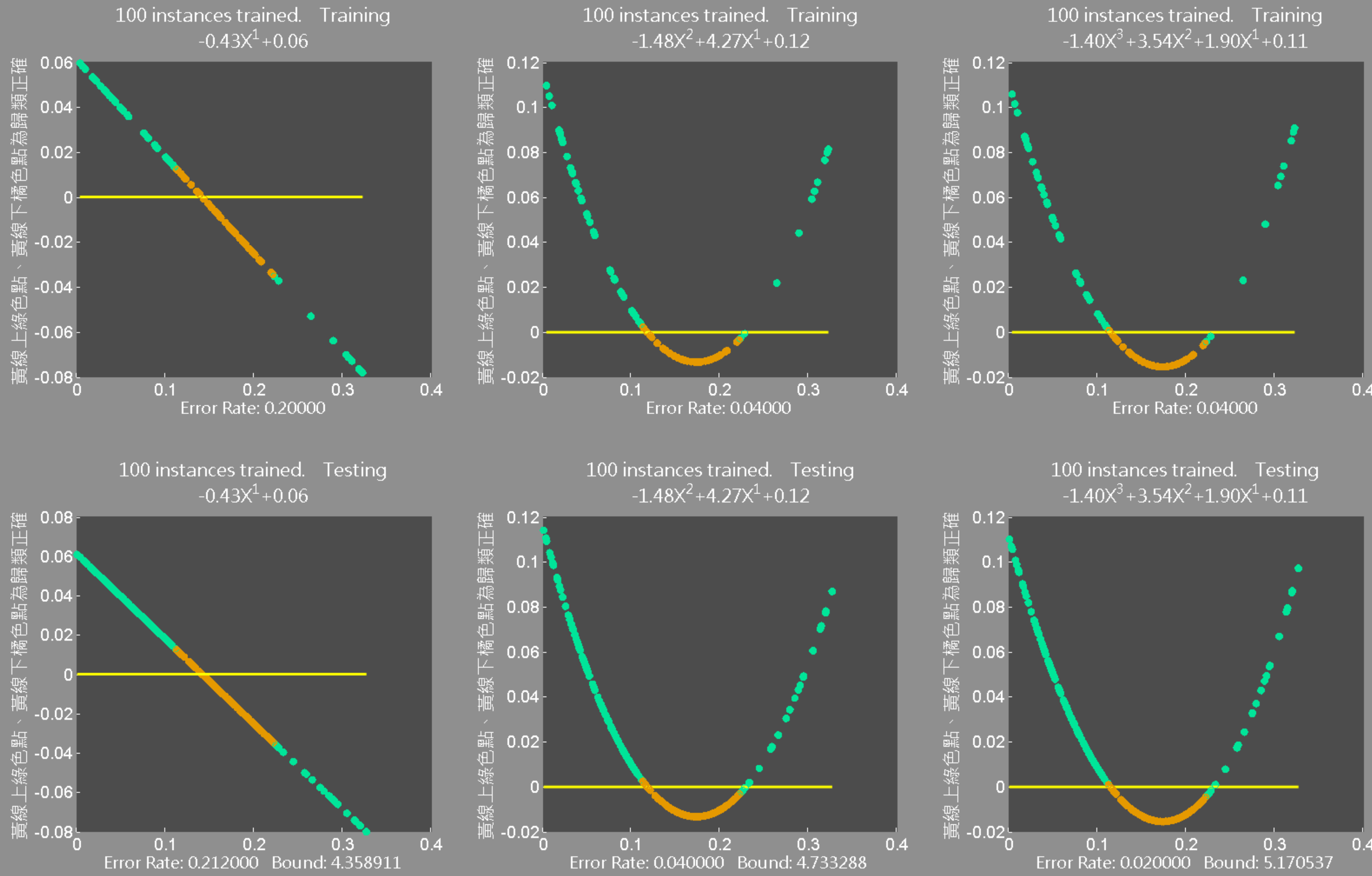
訓練和測試結果

實作細節

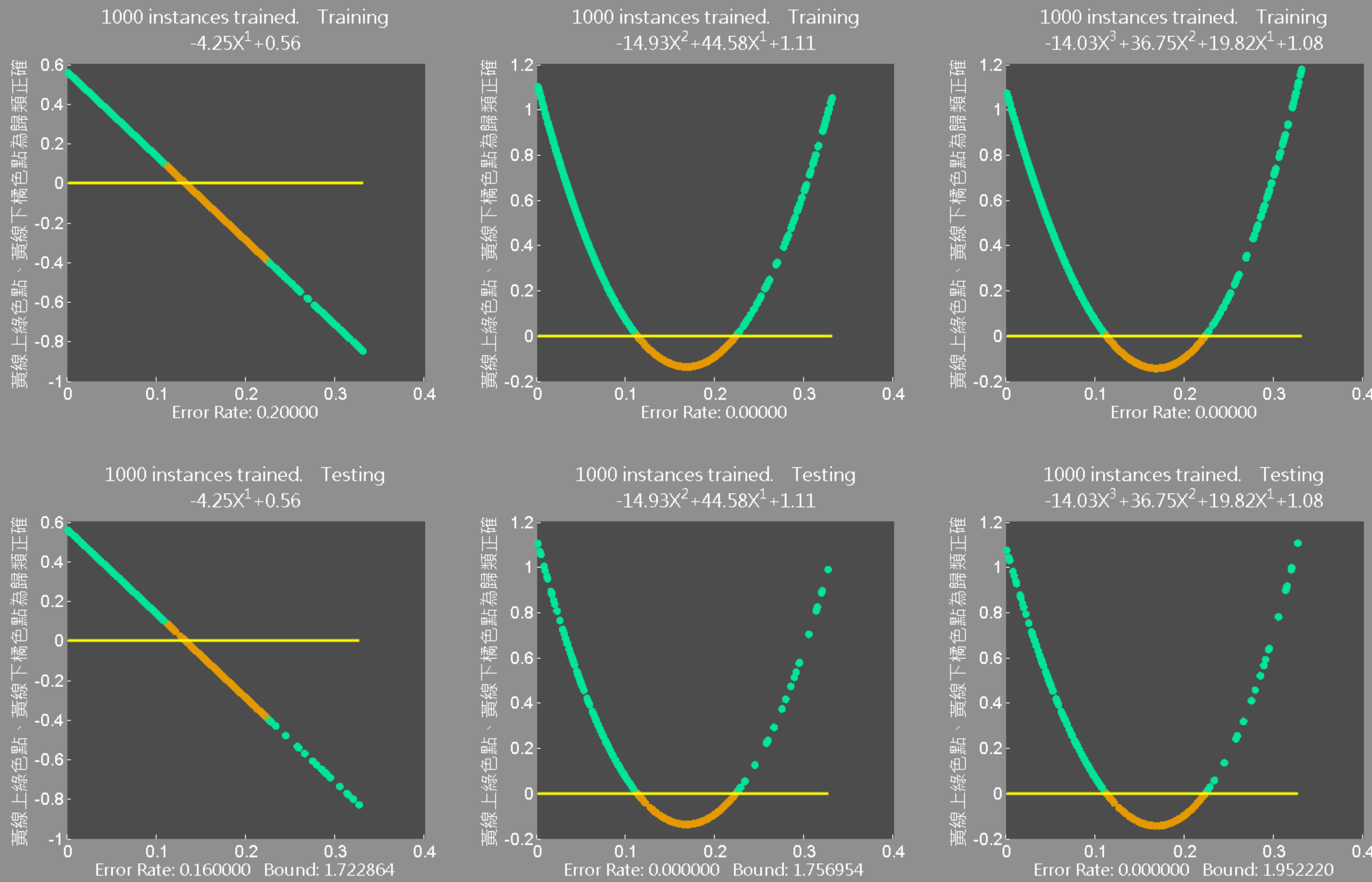
Bound

討論，關於Bound


10 instances trained. Training
-0.92X^1+0.09
Error Rate: 0.10000
10 instances trained. Training
-3.81X^2+10.00X^1+0.30
Error Rate: 0.00000
10 instances trained. Training
-3.53X^3+7.65X^2+4.49X^1+0.30
Error Rate: 0.00000
10 instances trained. Testing
-0.92X^1+0.09
Error Rate: 0.136000 Bound: 10.777981
10 instances trained. Testing
-3.81X^2+10.00X^1+0.30
Error Rate: 0.044000 Bound: 11.482605
10 instances trained. Testing
-3.53X^3+7.65X^2+4.49X^1+0.30
Error Rate: 0.068000 Bound: 12.055362
黃線上綠色點、黃線下橘色點為歸類正確



100 instances trained. Training
-0.43X^1+0.06
Error Rate: 0.20000
100 instances trained. Training
-1.48X^2+4.27X^1+0.12
Error Rate: 0.04000
100 instances trained. Training
-1.40X^3+3.54X^2+1.90X^1+0.11
Error Rate: 0.04000
100 instances trained. Testing
-0.43X^1+0.06
Error Rate: 0.212000 Bound: 4.358911
100 instances trained. Testing
-1.48X^2+4.27X^1+0.12
Error Rate: 0.040000 Bound: 4.733288
100 instances trained. Testing
-1.40X^3+3.54X^2+1.90X^1+0.11
Error Rate: 0.020000 Bound: 5.170537
黃線上綠色點、黃線下橘色點為歸類正確

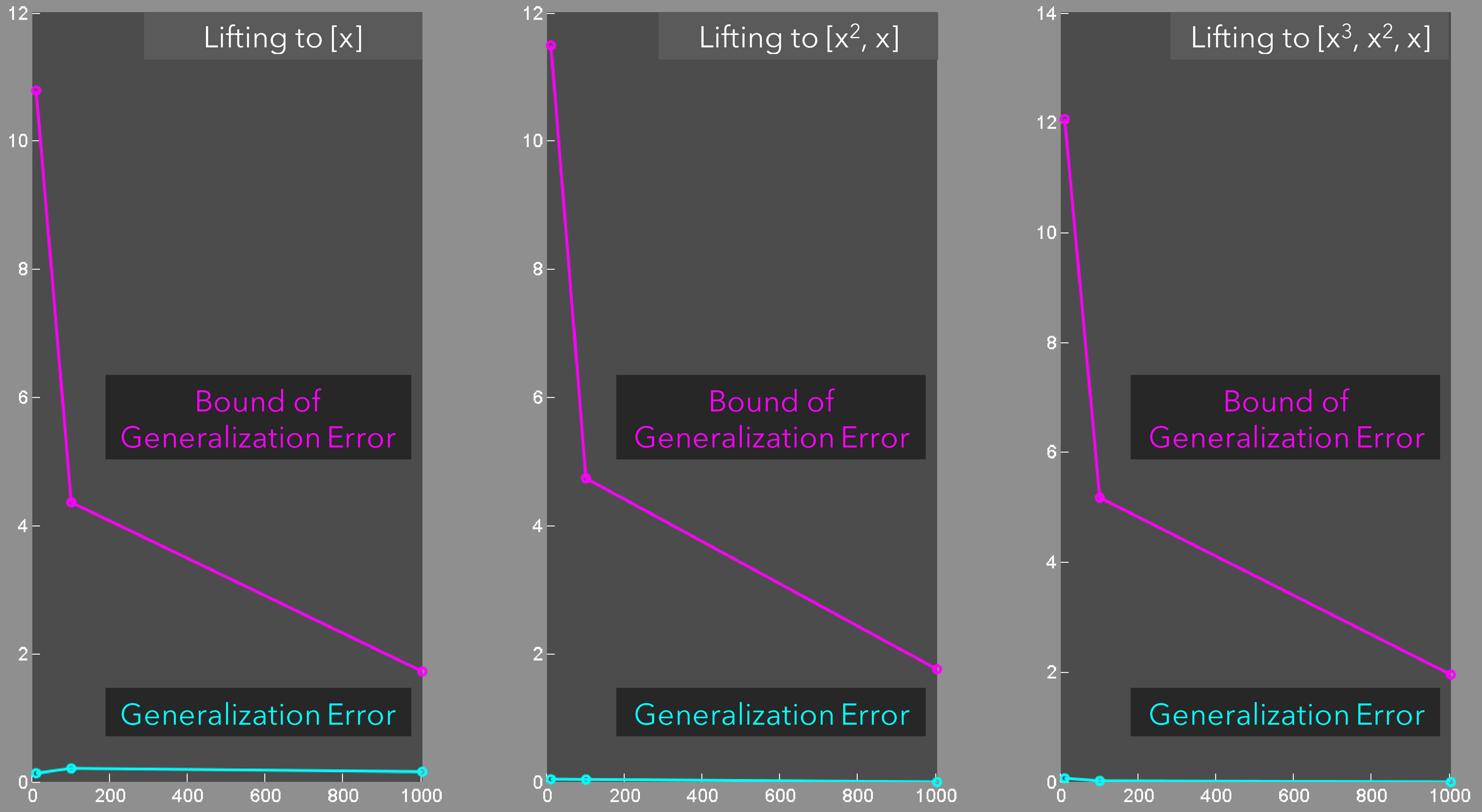


1000 instances trained. Training
-4.25X^1+0.56
黃線上綠色點、黃線下橘色點為歸類正確
Error Rate: 0.20000
1000 instances trained. Training
-14.93X^2+44.58X^1+1.11
黃線上綠色點、黃線下橘色點為歸類正確
Error Rate: 0.00000
1000 instances trained. Training
-14.03X^3+36.75X^2+19.82X^1+1.08
黃線上綠色點、黃線下橘色點為歸類正確
Error Rate: 0.00000
1000 instances trained. Testing
-4.25X^1+0.56
黃線上綠色點、黃線下橘色點為歸類正確
Error Rate: 0.160000 Bound: 1.722864
1000 instances trained. Testing
-14.93X^2+44.58X^1+1.11
黃線上綠色點、黃線下橘色點為歸類正確
Error Rate: 0.000000 Bound: 1.756954
1000 instances trained. Testing
-14.03X^3+36.75X^2+19.82X^1+1.08
黃線上綠色點、黃線下橘色點為歸類正確
Error Rate: 0.000000 Bound: 1.952220

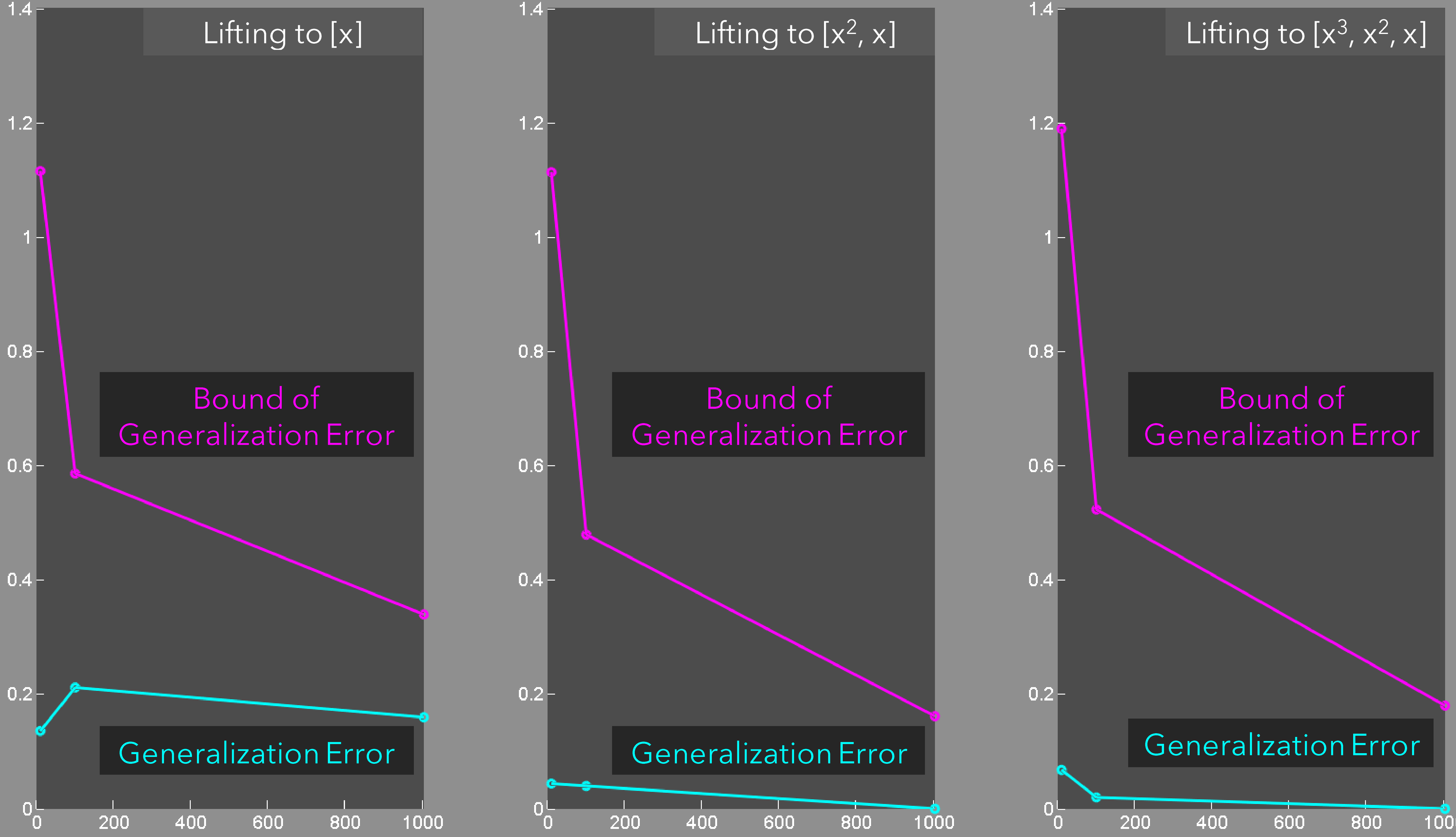
# Perceptron Learning Algorithm

- 主要演算法位於 perceptron_train.m 中
  - η的選取：
    - 固定η為0.001可以得到不錯的結果
    - 初始設為1(與資料尺度有關)，每次迴圈依次減半也能得到不錯的結果
  - 聊勝於無：
    - 在最後，以0,1-loss function再調整一下bias，可以使empirical error 再下降一些

吳尚鴻老師的教材所提供的VC bound計算方式，較鬆(looser)



維基百科所提供的VC bound計算方式，較緊(tighter)



- 測試資料用2維的PLA就可以完美分開，因此3維的bound較對應的2維的bound高，因為其VC dimension較高。
- 2維的bound也比1維的高，因為其VC dimension較高。奇怪的是，即使2維的empirical error較低，依然如此，因為bound實在太不準了。
- 這個bound非常鬆，時常>1，並且在資料完美可分的情況下，可以比實際的generalization error高出幾個數量級

這裡的維度指的是lifting function所在的維度

例如2維指的是lifting function為$[x^2, x]$

# 既然如此，那麼這個bound有什麼意義呢?

- 中國一位部落客認為
  - VC Bound对数据分布、目标函数、备选函数集、学习算法都没有要求，它牺牲了部分精确性，换来了无所不包的一般性。这使得VC Bound具有哲学意义上的指导性。

取自 http://my.oschina.net/findbill/blog/214849

# 關於這個 bound

- 使用者可以調整的變數
  - ε : 限制 empirical error 和 generalization error 的差距在ε以內
  - δ : 如上條件不發生的機率為δ (δ=0.1稱為有90%的信心水準)
  - N : 用於訓練的樣本資料數
  - 固定上述變數的其中2個，藉由VC bound可以得到另一個

# 不同教材所提到的bound

$$\varepsilon = 2\sqrt{\frac{32}{N}\left(VC(\mathcal{H})\log\frac{Ne}{VC(\mathcal{H})} + \log\frac{4}{\delta}\right)}$$

## 吳尚鴻老師的教材

來自ML2014講義

$$\varepsilon = \sqrt{\frac{8}{N}\ln\left(\frac{4(2N)^{d_{VC}}}{\delta}\right)}$$

## 林軒田老師的教材

來自《機器學習基石》講義

$$\varepsilon = \sqrt{\frac{h(\log(2N/h)+1) - \log(\eta/4)}{N}}$$

## 維基百科

http://en.wikipedia.org/wiki/VC_dimension

h 為 VC dimension，η 即 δ

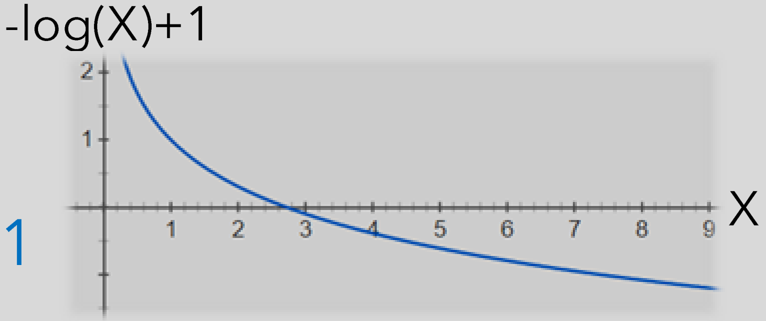
# 不同教材所提到的bound

$$2\sqrt{\frac{32}{N}\left(VC(\mathcal{H})\log\frac{Ne}{VC(\mathcal{H})}+\log\frac{4}{\delta}\right)}$$

$$\sqrt{\frac{8}{N}\ln\left(\frac{4(2N)^{d_{vc}}}{\delta}\right)}$$

$$\sqrt{\frac{h(\log(2N/h)+1)-\log(\eta/4)}{N}}$$

- 讓我們先聚焦於根號中的變數部分，即綠色框框罩住的地方。
- 林軒田老師的ε
  - 缺省了 -log(VC(H))+1




- 吳尚鴻老師的ε和維基百科的ε比起來，綠色框框內的變數因次完全相同，差異僅在log內的N與2N

# 不同教材所提到的bound

$$2\sqrt{\boxed{\frac{32}{N}}\left(VC(\mathcal{H})\log\frac{Ne}{VC(\mathcal{H})}+\log\frac{4}{\delta}\right)}$$

$$\sqrt{\boxed{\frac{8}{N}\ln}\left(\frac{4(2N)^{d_{vc}}}{\delta}\right)}$$

$$\sqrt{\frac{h(\log(2N/h)+1)-\log(\eta/4)}{N}}\Bigg)$$

- 常數係數方面
  - 以吳尚鴻老師給出的最為寬鬆
  - 而維基百科沒有任何常數係數，最緊