Lecture 2 : What to do if my network fails to train

Create at 2022/06/01

- 上課資源：
  1. 【機器學習 2022】再探寶可夢、數碼寶貝分類器 — 淺談機器學習原理 (https://www.youtube.com/watch?v=_j9MVVcvyZI)

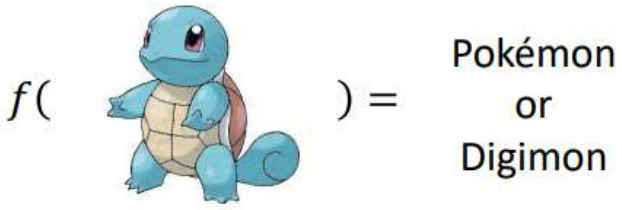
介紹

- 延伸教材：【機器學習2021】機器學習任務攻略 (https://www.youtube.com/watch?v=WeHM2xpYQpw)

如何分類是寶可夢還是數碼寶貝

Pokémon/Digimon Classifier

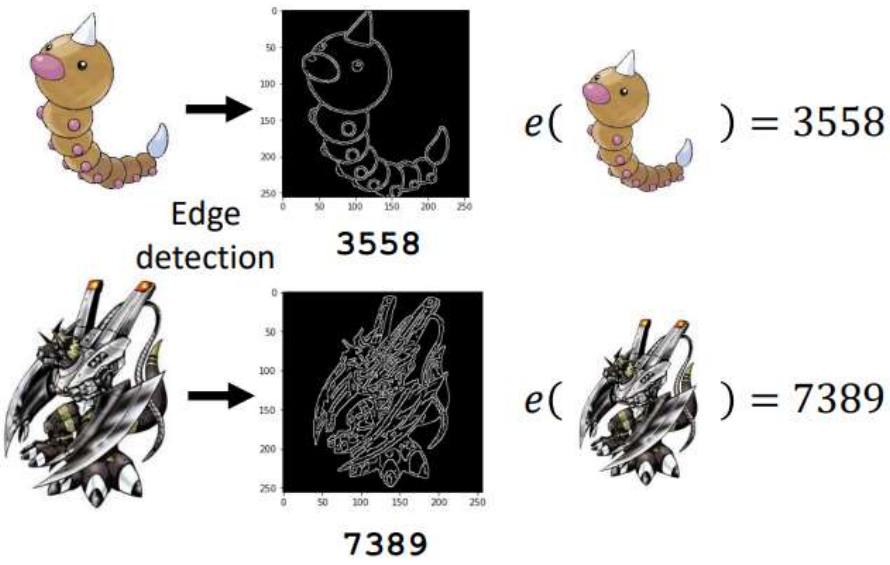
- We want to find a function .....



Determine a function with unknown parameters
(based on domain knowledge)

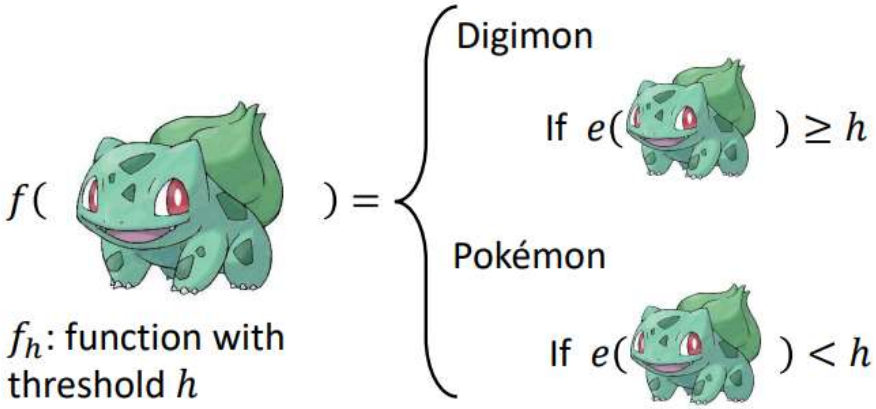
找一個有未知參數的 function

Observation



去做 edge detection，觀察出數碼寶貝的線條比較複雜

Function with Unknown Parameters



$\mathcal{H} = \{1,2,\cdots,10{,}000\}$ $|\mathcal{H}|$: number of candidate functions (model "complexity")

- 定義一個 function e，得到線條的複雜程度
- 如果線條的複雜程度超過某一個數值 h，分類為數碼寶貝
- 如果小於 h，分類為寶可夢
- 現在不知道 h 設在哪裡的時候可以準確分割寶可夢跟數碼寶貝
- 所有未知參數 h 的集合，稱為 H
- $|H|$ 為有多少可能的選擇，可能的數目稱為模型的複雜程度

Loss of a function (given data)

- Given a dataset $\mathcal{D}$

Pokémon

$$\mathcal{D} = \{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \cdots, (x^N, \hat{y}^N)\}$$

- Loss of a threshold h given data set $\mathcal{D}$

$$L(h, \mathcal{D}) = \frac{1}{N}\sum_{n=1}^{N} l(h, x^n, \hat{y}^n)$$

Error rate

$l(h, x^n, \hat{y}^n) \rightarrow \mathrm{I}(f_h(x^n) \neq \hat{y}^n)$

If $f_h(x^n) \neq \hat{y}^n$ Output 1

Otherwise Output 0

Don't like it? Of course, you can choose cross-entropy. ☺

- $Loss$ 是根據資料計算出來的
- D 裡面是有很多成對的寶可夢或數碼寶貝與正確答案
- 給定一個 h 根據 dataset D 去計算 $Loss$ 有多大
- 把 D 裡的每一筆資料都取出來，每筆資料都與 threshold 合起來，去計算 $loss$，代表這個 h 在這筆資料上表現的好壞
- $loss$ 越大表現越差，$loss$ 越小表現越好
- 定義 I 把 h 當作是 f 的參數，x^n 當作 input，如果輸出 $\hat{y}^n$ 跟正解不一樣，I 輸出 1，反之輸出 0
- 結果相加後平均起來，得到的就是 Error rate

Training Examples

- If we can collect all Pokémons and Digimons in the universe $\mathcal{D}_{all}$, we can find the best threshold h^{all}

$$h^{all} = arg \min_h L(h, \mathcal{D}_{all})$$

- We only collect some examples $\mathcal{D}_{train}$ from $\mathcal{D}_{all}$

$$\mathcal{D}_{train} = \{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \cdots, (x^N, \hat{y}^N)\}$$

$$(x^n, \hat{y}^n) \sim \mathcal{D}_{all}$$ independently and identically distributed (i.i.d.)

$$h^{train} = arg \min_h L(h, \mathcal{D}_{train})$$

- 所有的寶可夢跟數碼寶貝及合起來，稱為 D_{all}
- 也可以找到一個最好的參數，稱為 h^{all}
- h^{all} 是所有可能的 h 裡面，可以讓 D_{all} 資料集上 $Loss$ 最小的 h

- 事實上我們無法得到所有的寶可夢跟數碼寶貝
- 訓練資料集 D_{train} 有 N 筆資料
- 有了 D_{train} 可以找到一個 h，讓 Loss 最小，稱為 h^{train}

- If we can collect all Pokémons and Digimons in the universe $\mathcal{D}_{all}$, we can find the best threshold h^{all}

$$h^{all} = arg \min_h L(h, \mathcal{D}_{all})$$ 理想

- We only collect some examples $\mathcal{D}_{train}$ from $\mathcal{D}_{all}$

$$h^{train} = arg \min_h L(h, \mathcal{D}_{train})$$ 現實

We hope $L(h^{train}, \mathcal{D}_{all})$ and $L(h^{all}, \mathcal{D}_{all})$ are close.

現實 理想

- 上面是理想的狀態
- 下面是實際會發生的情況
- 希望是理想與現實非常接近

We hope $L(h^{train}, \mathcal{D}_{all})$ and $L(h^{all}, \mathcal{D}_{all})$ are close.

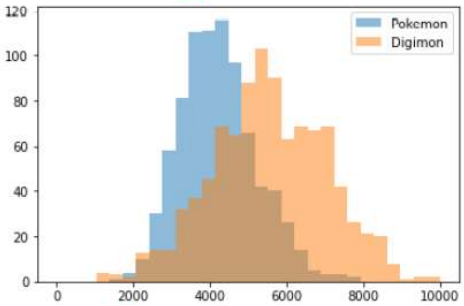
Pokémon: 819

Digimon: 971

In most applications, you cannot obtain $\mathcal{D}_{all}$.

(Testing data $\mathcal{D}_{test}$ as the proxy of $\mathcal{D}_{all}$)

All Pokémons and Digimons we know as $\mathcal{D}_{all}$



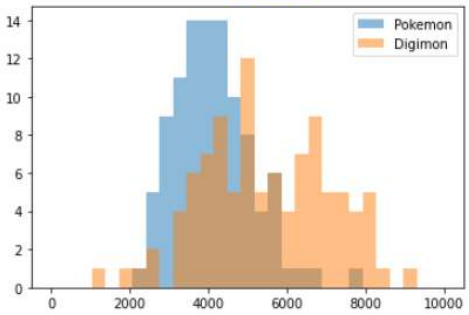
$h^{all} = 4824$

$L(h^{all}, \mathcal{D}_{all}) = 0.28$

Source of Digimon:
https://github.com/mrok273/Qiita
Source of Pokémon:
https://www.kaggle.com/kvpratama/pokemon-images-dataset/data

- 實際應用裡面，因為沒有 D_{all} 所以常見的做法是會準備一個 testing set D_{test}
- 我們會希望這個 testing set 在所有 data 裡面是有代表性的
- 假設 h^{all} 是 4824 (線條數目)，得到 $Loss$ 為 0.28

Sample 200 Pokémons and Digimons as $\mathcal{D}_{train1}$

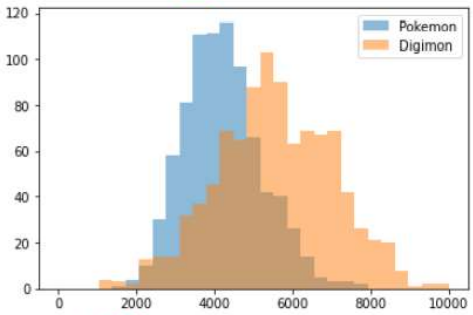


$h^{train1} = 4727$

$L(h^{train1}, \mathcal{D}_{train1}) = 0.27$

Even lower than $L(h^{all}, \mathcal{D}_{all})$?

All Pokémons and Digimons we know as $\mathcal{D}_{all}$



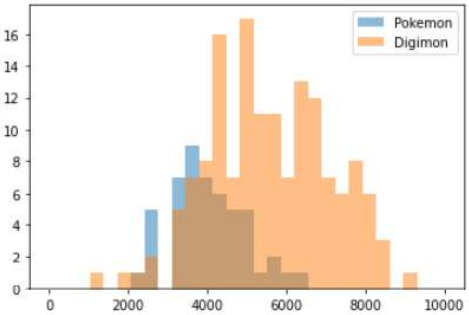
$h^{all} = 4824$

$L(h^{all}, \mathcal{D}_{all}) = 0.28$

$L(h^{train1}, \mathcal{D}_{all}) = 0.28$

- 假設準備一個 training dataset 為 D_{train1}，找到一個最好的 h 叫 h^{train1} = 4727，$Loss$ 為 0.27
- 把 h^{train1} 用在 D_{all} 發現 $Loss$ 也是0.28
- 顯然現實與理想是差不多的
- $L(h^{train1}, D_{all})$ 才是我們真正關心的，跟 sample 到的資料有很大的關係

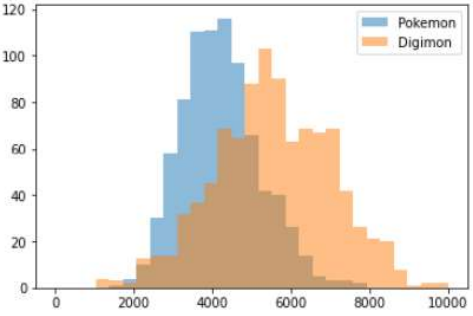
Sample 200 Pokémons and Digimons as $\mathcal{D}_{train2}$



$h^{train2} = 3642$

$L(h^{train2}, \mathcal{D}_{train2}) = 0.20$

All Pokémons and Digimons we know as $\mathcal{D}_{all}$



$h^{all} = 4824$

$L(h^{all}, \mathcal{D}_{all}) = 0.28$

$L(h^{train2}, \mathcal{D}_{all}) = 0.37$

- 抓出另外一個資料集叫做 D_{train2}
- 找到一個最好的 h 叫 h^{train2} = 3642，$Loss$ 為 0.20
- $L(h^{train2}, D_{all})$ 才是我們真正關心的
- h^{train2} 在 D_{all} 上的表現並不好
- 可以得知，如果 sample 到好的資料，理想跟現實就會比較接近

What do we want?

$L(h^{train}, \mathcal{D}_{train})$ can be smaller than $L(h^{all}, \mathcal{D}_{all})$

We want $L(h^{train}, \mathcal{D}_{all}) - L(h^{all}, \mathcal{D}_{all}) \leq \delta$

What kind of $\mathcal{D}_{train}$ fulfill it?

$$\forall h \in \mathcal{H}, |L(h, \mathcal{D}_{train}) - L(h, \mathcal{D}_{all})| \leq \delta/2$$

$\mathcal{D}_{train}$ is a good proxy of $\mathcal{D}_{all}$ for evaluating loss L given any h.

$$L(h^{train}, \mathcal{D}_{all}) \leq L(h^{train}, \mathcal{D}_{train}) + \delta/2$$

$$\leq L(h^{all}, \mathcal{D}_{train}) + \delta/2 \qquad h^{train} = arg\min_h L(h, \mathcal{D}_{train})$$

$$\leq L(h^{all}, \mathcal{D}_{all}) + \delta/2 + \delta/2 = L(h^{all}, \mathcal{D}_{all}) + \delta$$

- 希望 h^{train} 在 D_{all} 上的 $Loss$ 減去 h^{all} 在 D_{all} 上的 $Loss$ 小於等於某個數值 δ
- $L(h^{all}, D_{all})$ 一定會比 $L(h^{train}, D_{all})$ 小，因為 h^{all} 是 D_{all} 上找出來最好的 threshold
- 什麼樣的 D_{train} 可以讓理想跟現實很接近
  - 窮舉所有可能的 h，對 h 而言，在 D_{train} 上算出的 $Loss$ 跟 在 D_{all} 上算出的 $Loss$ 小於等於 $\delta/2$，上面的狀況就會成立，這意味著 D_{train} 跟 D_{all} 很像

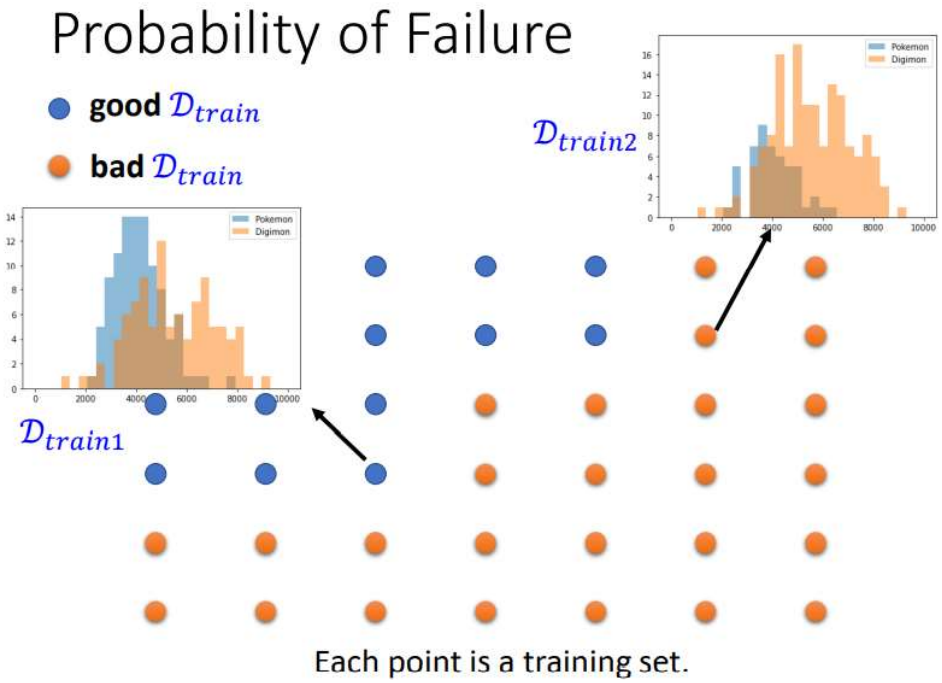
We want to sample **good** $\mathcal{D}_{train}$ $\varepsilon = \delta/2$

$$\forall h \in \mathcal{H}, |L(h, \mathcal{D}_{train}) - L(h, \mathcal{D}_{all})| \leq \varepsilon$$

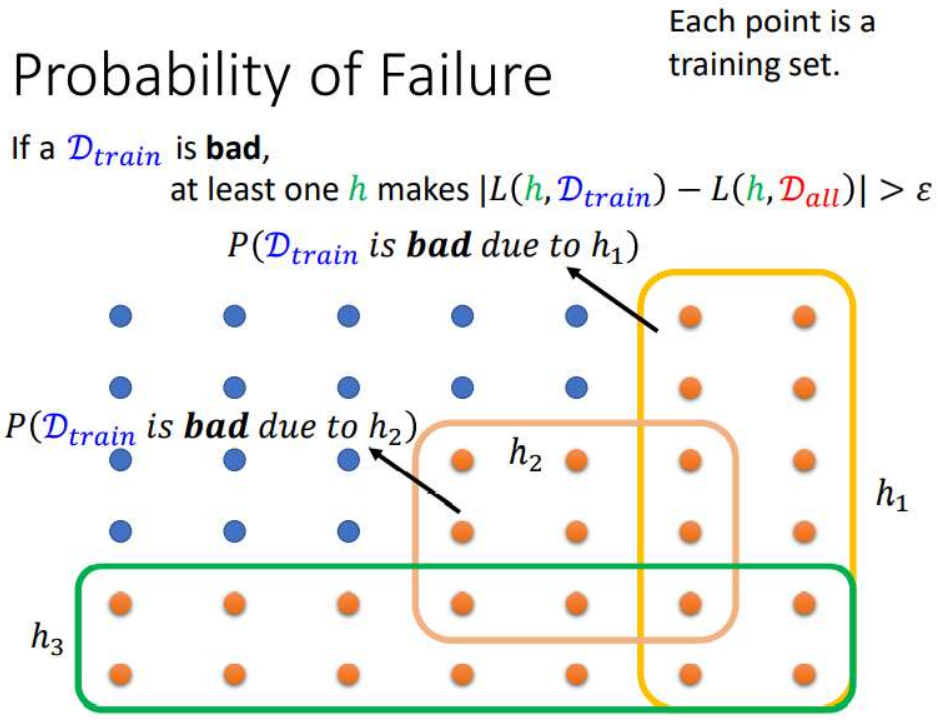
What is the probability of sampling **bad** $\mathcal{D}_{train}$?

目標是 sample 到一個好的 training data，在 D_{train} 上算出的 $Loss$ 減去在 D_{all} 上算出的 $Loss$ 小於等於一個很小的數值 ε

一般化的原理，可以用在很多不同的情境



估測找到壞的訓練資料集的機率有多大

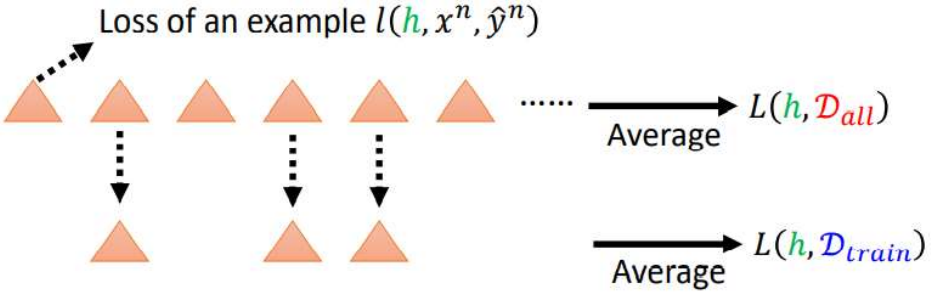


- 如果可以找到一個在 D_{train} 上算出的 $Loss$ 減去在 D_{all} 上算出的 $Loss$ 大於 ε，那這個訓練資料就是不好的
- 所以每一個壞的資料集，一定有至少一個 h 把它弄壞

$$P(\mathcal{D}_{train} \text{ is } \textbf{bad}) = \bigcup_{h \in \mathcal{H}} P(\mathcal{D}_{train} \text{ is } \textbf{bad} \text{ due to } h)$$

$$\leq \sum_{h \in \mathcal{H}} P(\mathcal{D}_{train} \text{ is } \textbf{bad} \text{ due to } h)$$

$$|L(h, \mathcal{D}_{train}) - L(h, \mathcal{D}_{all})| > \varepsilon \qquad L(h, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^{N} l(h, x^n, \hat{y}^n)$$



- 因為壞的資料集重複的 dataset 難以考慮，所以直接用相加的方式取得 upper bound

Hoeffding's Inequality:

$$P(\mathcal{D}_{train} \text{ is bad due to } h) \leq 2exp(-2N\varepsilon^2)$$

- The range of loss L is $[0,1]$
- N is the number of examples in $\mathcal{D}_{train}$

$$P(\mathcal{D}_{train} \text{ is } \textbf{bad}) = \bigcup_{h \in \mathcal{H}} P(\mathcal{D}_{train} \text{ is } \textbf{bad} \text{ due to } h)$$

$$\leq \sum_{h \in \mathcal{H}} P(\mathcal{D}_{train} \text{ is } \textbf{bad} \text{ due to } h)$$

$$\leq \sum_{h \in \mathcal{H}} 2exp(-2N\varepsilon^2)$$

$$= |\mathcal{H}| \cdot 2exp(-2N\varepsilon^2)$$

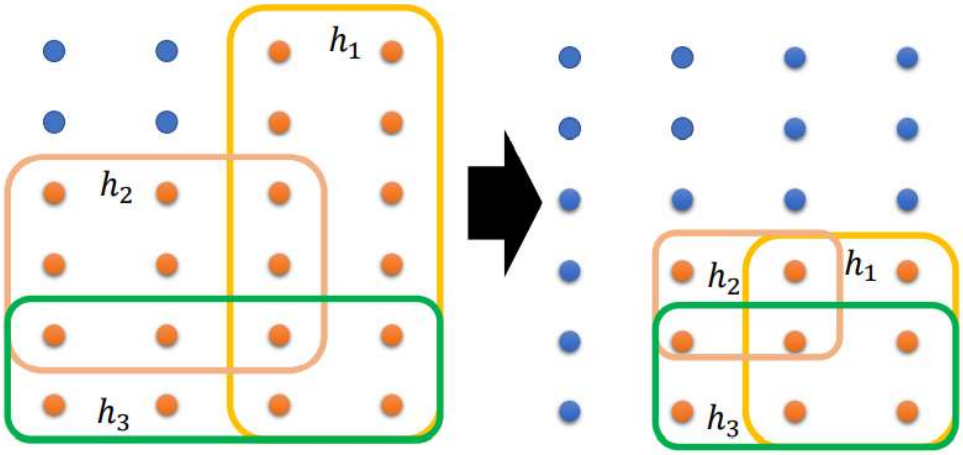
How to make $P(\mathcal{D}_{train} \text{ is bad})$ smaller?

Larger N and smaller $|\mathcal{H}|$

- $P(D_{train}$ is bad$) \leq |H| * 2exp(-2N\varepsilon^2)$
- 讓 N 越大，sample 到壞資料的機率越低
- 讓 H 越小，sample 到壞資料的機率越低

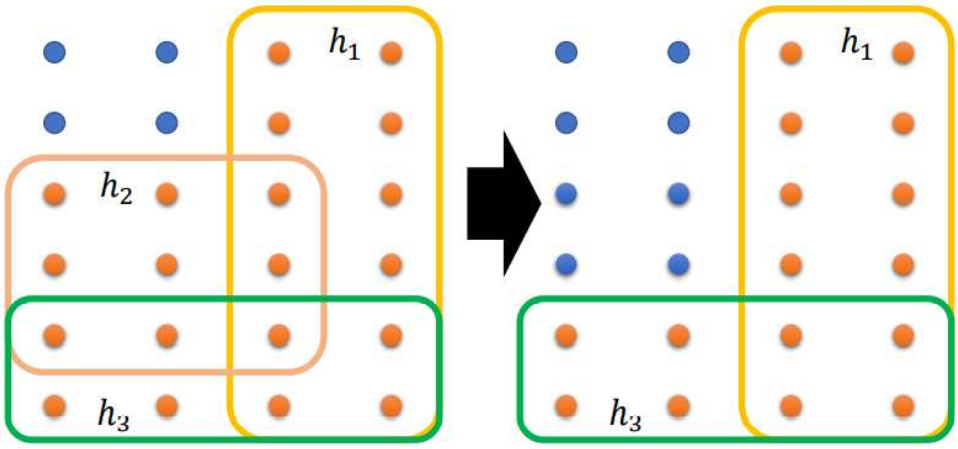
$$P(\mathcal{D}_{train} \textit{ is } \textbf{bad}) \leq |\mathcal{H}| \cdot 2exp(-2N\varepsilon^2)$$

Larger N



- 把 N 調大，讓每一個 h 可以弄壞的 training data 變少，可以弄壞的難度增加

Smaller $|\mathcal{H}|$



- 把 $|H|$ 變小，可以讓差的 dataset sample 到的機率變小

Example

$$\mathcal{H} = \{1,2,\cdots,10{,}000\}$$
$$\mathcal{D}_{train} = \{(x^1,\hat{y}^1),(x^2,\hat{y}^2),\cdots,(x^N,\hat{y}^N)\}$$
$$\forall h \in \mathcal{H}, |L(h,\mathcal{D}_{train}) - L(h,\mathcal{D}_{all})| \leq \varepsilon$$

$$P(\mathcal{D}_{train} \text{ is } \boldsymbol{bad}) \leq |\mathcal{H}| \cdot 2exp(-2N\varepsilon^2)$$

$|\mathcal{H}| = 10000, N = 100, \varepsilon = 0.1$ Usually happen QQ

$$P(\mathcal{D}_{train} \text{ is } bad) \leq 2707$$

$|\mathcal{H}| = 10000, N = 500, \varepsilon = 0.1$

$$P(\mathcal{D}_{train} \text{ is } bad) \leq 0.91$$

$|\mathcal{H}| = 10000, N = 1000, \varepsilon = 0.1$

$$P(\mathcal{D}_{train} \text{ is } bad) \leq 0.00004$$

增加 sample 到的資料數目 N 是很有效的

Example

$$P(\mathcal{D}_{train} \text{ is } \boldsymbol{bad}) \leq |\mathcal{H}| \cdot 2exp(-2N\varepsilon^2)$$

If we want $P(\mathcal{D}_{train} \text{ is } bad) \leq \delta$

How many training examples do we need?

$$|\mathcal{H}| \cdot 2exp(-2N\varepsilon^2) \leq \delta \Rightarrow N \geq \frac{log(2|\mathcal{H}|/\delta)}{2\varepsilon^2}$$

$|\mathcal{H}| = 10000, \delta = 0.1, \varepsilon = 0.1$

$$\Rightarrow N \geq 610$$

計算出 $N \geq 610$

Model Complexity

$$P(\mathcal{D}_{train} \text{ is } \textbf{bad}) \leq |\mathcal{H}| \cdot 2exp(-2N\varepsilon^2)$$

Why don't we simply use a very small $|\mathcal{H}|$?

"$\mathcal{D}_{train}$ is **good**" means ... 理想崩壞

$$\forall h \in \mathcal{H}, |L(h, \mathcal{D}_{train}) - L(h, \mathcal{D}_{all})| \leq \varepsilon$$

Larger loss

$$L(h^{train}, \mathcal{D}_{all}) - L(h^{all}, \mathcal{D}_{all}) \leq \delta \quad \varepsilon = \delta/2$$

$$h^{all} = arg \min_{h \in \mathcal{H}} L(h, \mathcal{D}_{all})$$

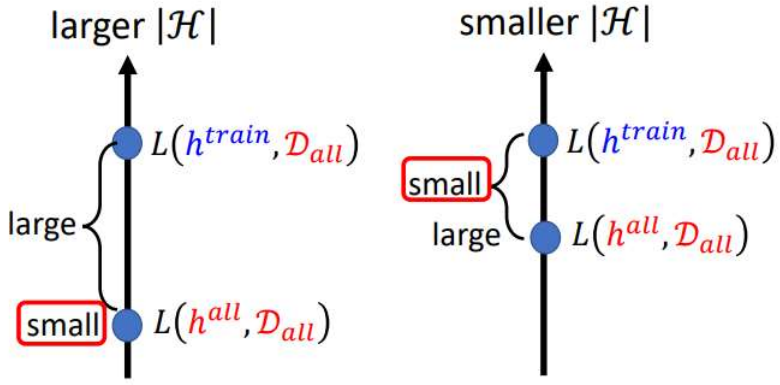
fewer candidates

- 當我們說一個 training set 是好的，意思是可以找到一個 h^{train} ，並且在 D_{all} 上不管是用 h^{train} 算還是 h^{all} 上算，他們的 $Loss$ 差距都會小於 δ
- 當我們把 H 弄得很小的時候，可以選擇的 h 就會很有限，就沒有辦法讓 h^{all} 在 D_{all} 上算出來的 $Loss$ 真的很小，意味著**理想崩壞**

Tradeoff of Model Complexity

Larger N and smaller $|\mathcal{H}|$ ➡ $L(h^{train}, \mathcal{D}_{all}) - L(h^{all}, \mathcal{D}_{all}) \leq \delta$

Smaller $|\mathcal{H}|$ ➡ Larger $L(h^{all}, \mathcal{D}_{all})$



魚與熊掌可以兼得嗎？ Yes, **Deep Learning.**

- 選擇比較大的 H，理想跟現實的差距就會比較明顯
- 選擇比較小的 H，理想跟現實的差距比較小，但是理想已經崩壞，所以與理想接近也沒什麼用

- 課程網頁 (https://speech.ee.ntu.edu.tw/~hylee/ml/2022-spring.php)

tags: 2022 李宏毅_機器學習