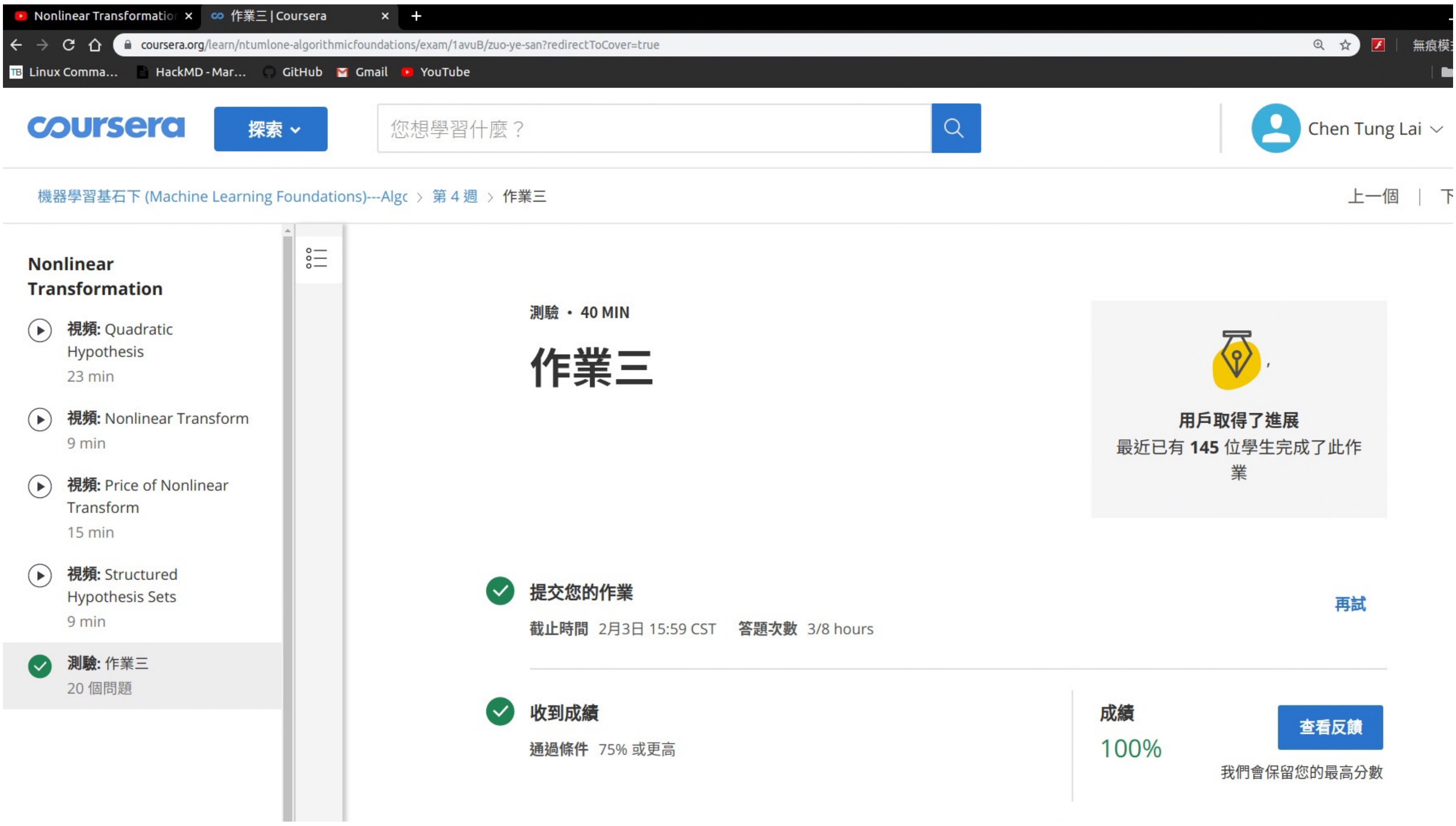
Machine Learning Fundamental HW3

資工所碩一 R08922143 賴振東



2. SGD: $w_{t+1} \leftarrow w_t + \eta(-\nabla err(w))$

PLA: $w_{t+1} \leftarrow w_t + 1 \cdot [y_n \neq sign(w_t^T x_n)](y_n x_n)$

prove $err(w) = \max(0, -yw^T x)$

$\Rightarrow$ prove $[y_n \neq sign(w_t^T x_n)](y_n x_n) = -\nabla err(w) = -\nabla \max(0, -yw^T x)$

case 1: $y_n = sign(w_t^T x_n)$

$\Rightarrow [y_n \neq sign(w_t^T x_n)](y_n x_n) = 0$

$\because y_n = sign(w_t^T x_n) \quad \therefore -yw^T x < 0$

$\Rightarrow -\nabla \max(0, -yw^T x) = 0$

case 2: $y_n \neq sign(w_t^T x_n)$

$\Rightarrow [y_n \neq sign(w_t^T x_n)](y_n x_n) = \begin{cases} x_n, & \text{if } y_n > 0 \\ -x_n, & \text{if } y_n < 0 \end{cases}$

$\because y_n \neq sign(w_t^T x_n)$

$\therefore -yw^T x > 0$

$\Rightarrow \max(0, -yw^T x) = -yw^T x$

$\Rightarrow -\nabla \max(0, -y_n w_t^T x_n) = y_n x_n = \begin{cases} x_n, & \text{if } y_n > 0 \\ -x_n, & \text{if } y_n < 0 \end{cases}$

by case 1, 2. $\nabla err(w) = \nabla \max(0, -yw^T x)$

$\Rightarrow$ err(w) of PLA 為 $\max(0, -yw^T x)$

\#

3、$E(u+\Delta u, v+\Delta v)$

by Taylor Series

$$= E(u,v) + \nabla E(u,v)\Delta(u,v) + \frac{1}{2}\nabla^2 E(u,v)\Delta(u,v)^2 + \cdots$$

当 $\Delta(u,v) \rightarrow (0,0)$

$$\Rightarrow 0 = 0 + \nabla E(u,v)\Delta(u,v) + \frac{1}{2}\nabla^2 E(u,v)\Delta(u,v)^2.$$

$$\Rightarrow \frac{\partial E}{\partial \Delta(u,v)} = \nabla E(u,v) + \nabla^2 E(u,v)\Delta(u,v) = 0.$$

又 $\because H = \nabla^2 E(u,v)$ 為正定矩陣

$$\Rightarrow \Delta(u,v) = \frac{\nabla E(u,v)}{\nabla^2 E(u,v)}$$

$= (\nabla^2 E(u,v))^{-1}\nabla E(u,v)$ 可使 $E(u+\Delta u, v+\Delta v)$ 有最小值

4. 当 $y \in \{-1, +1\}$. $\Rightarrow$ $\text{likelihood}(h) \propto \prod_{n=1}^{N} h(y_n x_n)$

$$\Rightarrow \max_h \prod_{n=1}^{N} h(y_n x_n)$$

$$\Rightarrow \max_w \prod_{n=1}^{N} \theta(y_n w^T x_n)$$

$$\Rightarrow \min_w \frac{1}{N}\sum_{n=1}^{N} -\ln(y_n w^T x_n)$$

当 y 擴展為 K-class, $y \in \{1, 2, \cdots K\}$.

$$\Rightarrow h_y(x) = (\exp(w_y^T x)) / (\textstyle\sum_{k=1}^{K} \exp(w_k^T x))$$

為猜 $y=1\sim k$ 其中某一个的机率

$$\therefore \quad h_w(x) = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_k(x) \end{bmatrix} = \frac{1}{\sum_{k=1}^{K} \exp(w_k^T x)} \begin{bmatrix} \exp(w_1^T x) \\ \vdots \\ \exp(w_k^T x) \end{bmatrix}$$

$$\Rightarrow \max_{h} \prod_{n=1}^{N} h(y_n x_n)$$

$$\Rightarrow \max_{w} \prod_{n=1}^{N} \frac{\exp(w_{y_n}^T x_n)}{\sum_{k=1}^{K} \exp(w_k^T x_n)}$$

取 ln

$$\Rightarrow \max_{w} \frac{1}{N} \sum_{n=1}^{N} \ln\left[\exp(w_{y_n}^T x_n) - \sum_{k=1}^{K} \exp(w_k^T x_n)\right]$$

$$= \min_{w} \frac{1}{N} \sum_{n=1}^{N} \left\{ \ln\left[\sum_{k=1}^{K} \exp(w_k^T x_n) - \exp(w_{y_n}^T x_n)\right]\right\}$$

$$= \min_{w} \frac{1}{N} \sum_{n=1}^{N} \left\{ \ln\left[\sum_{k=1}^{K} \exp(w_k^T x_n)\right] - w_{y_n}^T x_n \right\}$$

#

5. let $A = \sum_{n=1}^{N} (y_n - w^T x_n)^2 + \sum_{k=1}^{K} (\tilde{y}_k - w^T \tilde{x}_k)^2$

$$= \left\| \begin{bmatrix} y_1 - w^T x_1 \\ \vdots \\ y_N - w^T x_n \end{bmatrix} \right\|^2 + \left\| \begin{bmatrix} \tilde{y}_1 - w^T \tilde{x}_1 \\ \vdots \\ \tilde{y}_k - w^T x_k \end{bmatrix} \right\|^2$$

$$= |Xw - y|^2 + |\tilde{X}w - \hat{y}|^2$$

$$= w^T X^T X w - 2w^T X^T y + y^T y + w^T \tilde{X}^T \tilde{X} w - 2w^T \tilde{X}^T y + y^T y$$

找 $\min_{w} \frac{A}{N+k} \Leftrightarrow$ 找 $\frac{\partial A}{\partial w} = 0$ 之 w

$$\Rightarrow \frac{\partial A}{\partial w} = 2X^T X w - 2X^T y + 2\tilde{X}^T \tilde{X} w - 2\tilde{X}^T \tilde{y} = 0$$

$$\Rightarrow (\cancel{2}X^T X + \cancel{2}\tilde{X}^T \tilde{X})w - \cancel{2}X^T y - \cancel{2}\tilde{X}^T \tilde{y} = 0$$

$$\Rightarrow w = (X^T X + \tilde{X}^T \hat{X})^{-1} (X^T y - \tilde{X}^T \tilde{y})$$

※

6.

$$w_{reg} = \operatorname{argmin}_w \frac{\lambda}{N}|w|^2 + \frac{1}{N}|Xw - y|^2$$

$$\Leftrightarrow \min_w \left(\lambda w^T w + w^T X^T X w - 2 w^T X^T y + y^T y \right)$$

$$\Leftrightarrow \frac{\partial}{\partial w} \left(\lambda w^T w + w^T X^T X w - 2 w^T X^T y + y^T y \right) = 0$$

$$\Rightarrow 2\lambda w + 2 X^T X w - 2 X^T y = 0$$

$$\Rightarrow (\lambda I + X^T X) w = X^T y$$

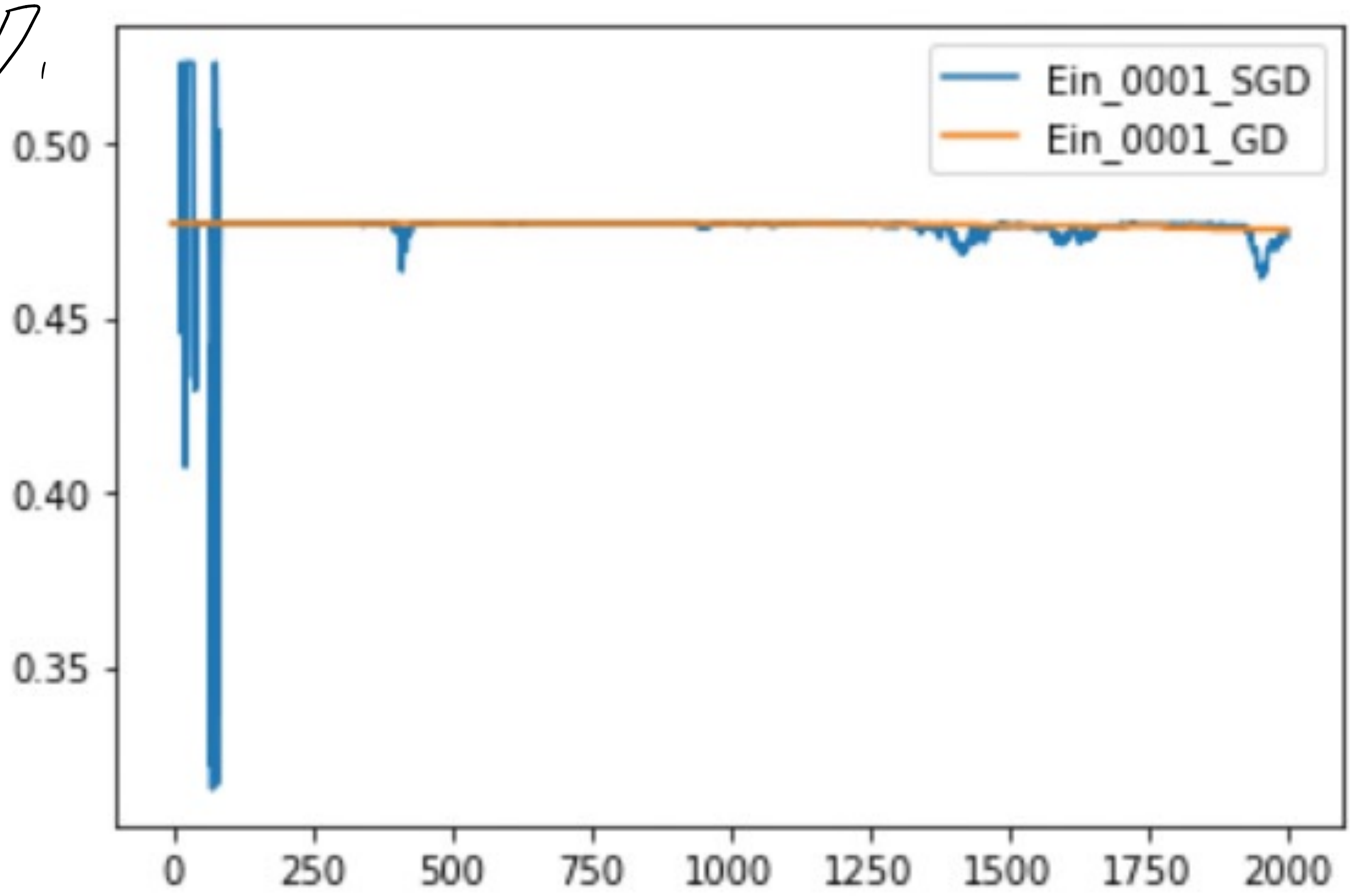
$$\Rightarrow w = (X^T X + \lambda I)^{-1} X^T y$$

$$= (X^T X + \tilde{X}^T \tilde{X})^{-1} (X^T y - \tilde{X}^T \tilde{y})$$ (by前一題)

$$\Rightarrow \begin{cases} \tilde{X}^T \tilde{X} = \lambda I \\ \tilde{X}^T \tilde{y} = 0 \end{cases} \Rightarrow \begin{cases} \tilde{X} = \sqrt{\lambda} I \\ \tilde{y} = 0 \end{cases}$$

#

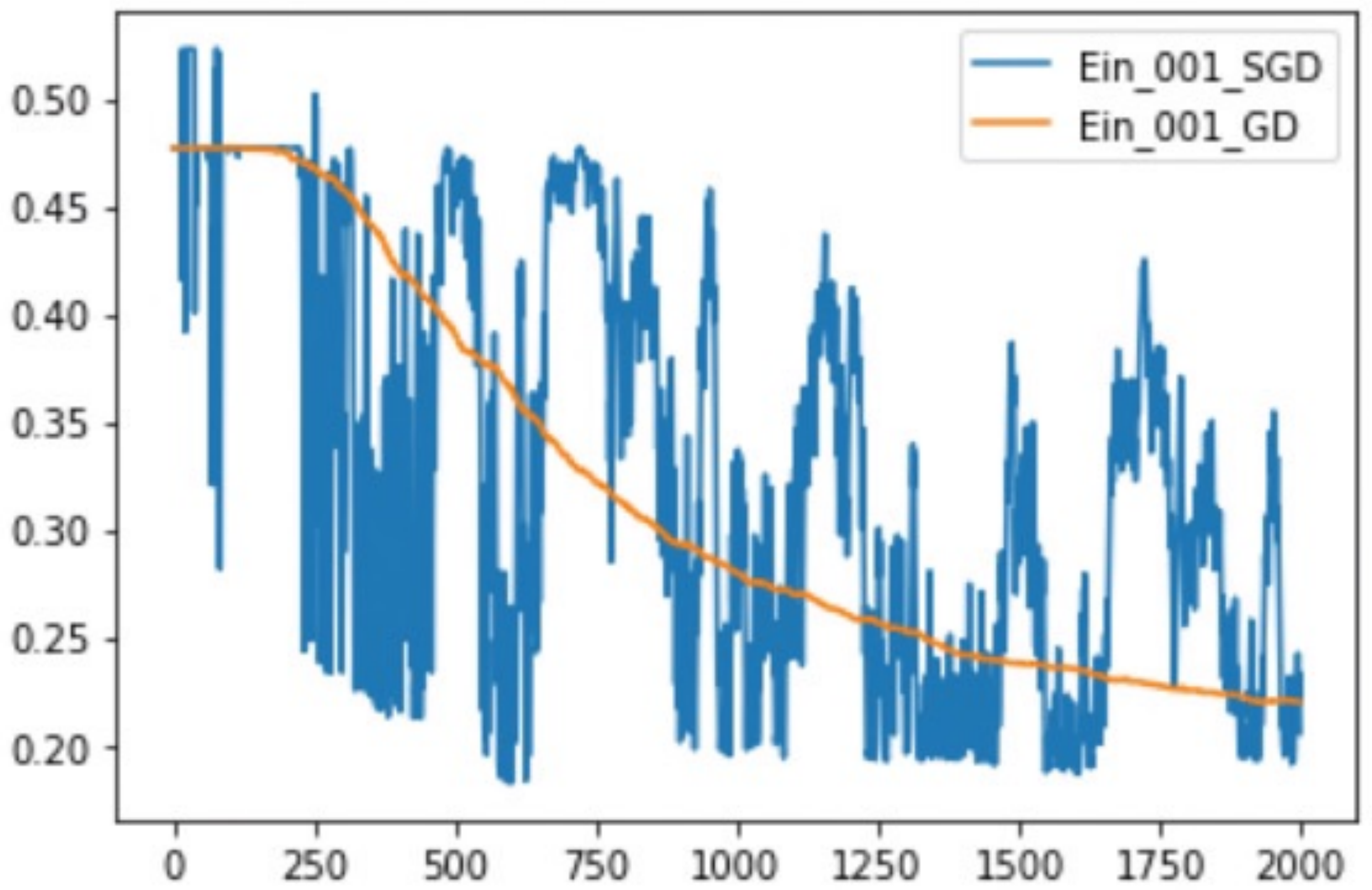
7.




```
eta = 0.001
=====GD=====
Ein = 0.466
Eout = 0.475
=====SGD=====
Ein = 0.464
Eout = 0.473
```

圖中橫軸為迭代次數，也就是 weight 更新的次數，而縱軸則為 Ein(wt)，是每一個 wt 所對應到的 error rate。
當 eta = 0.001 時，因為每一步都跨得很小(w 改變幅度不大) 所以可以看到，不論是 GD 還是 SGD，在進行了2000次更新後 error rate 都沒有顯著下降。但還是可以發現 Stochastic gradient decent 雖節省了計算量，但穩定性也可能跟著下降，可能出現更新完 weight ，error rate 反而上升的情形。

8.




```
eta = 0.01
=====GD=====
Ein = 0.197
Eout = 0.22
=====SGD=====
Ein = 0.187
Eout = 0.20533333333333334
```

當 eta = 0.01 時，每次 weight 更新跨出的 step 較大，在經過 2000次更新後 GD 與 SGD 的 Ein 都有顯著的下降，甚至 SGD 的 Ein、Eout 都較 GD 低一些。但這是否代表 SGD 不論計算量或準確率都高於 GD 呢？我想透過分析上圖就可以得出否定的答案。雖然 SGD 的 Ein 呈現下降的趨勢，但穩定度遠遠不如 GD。舉例來說，在第1962次更新後，SGD 的 Ein 為 0.35，遠高於同樣更新次數的 GD 的 Ein，若這種突然飆高的情況發生在最後一次，而且若沒有儲存先前的 weight 值，則可能得到相當差的成果。

所以我認為雖然 gradient decent 每次都要計算 N (1000)個點後才能進行一次更新，但是它可以保證 Ein 是穩定下降的，這樣的特性 SGD 無法辦到，所以 GD 仍有其必要性。或許可以試著在 1 與 N 之間找到一個性價比較高的平衡點，在減少計算量同時又不喪失太多穩定度。