

Machine Learning Fundamental HW3

資工所碩一 R08922143 賴振東

Nonlinear Transformation

作業三 | Coursera

course

ra.org/learn/htumlone-algorithmicfoundations/exam/1avuB/zuo-ye-san?redirectToCover=true

無痕模

Linux Comma...

HackMD - Mar...

GitHub

Gmail

YouTube

course

era

探索

您想學習什麼？

Chen Tung Lai

機器學習基石下 (Machine Learning Foundations)---Algc

第 4 週

作業三

上一個

下

Nonlinear Transformation

視頻: Quadratic Hypothesis 23 min

視頻: Nonlinear Transform 9 min

視頻: Price of Nonlinear Transform 15 min

視頻: Structured Hypothesis Sets 9 min

測驗: 作業三 20 個問題

測驗 • 40 MIN

作業三

用戶取得了進展
最近已有 145 位學生完成了此作業

提交您的作業

截止時間 2月3日 15:59 CST 答題次數 3/8 hours

再試

收到成績

通過條件 75% 或更高

成績 100%

查看反饋

我們會保留您的最高分數

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

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

$$\text{prove } \text{err}(w) = \max(0, -y w^T x)$$

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

$$\text{case 1: } y_n = \text{sign}(w_t^T x_n)$$

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

$$\because y_n = \text{sign}(w_t^T x_n) \therefore -y w^T x \leq 0$$

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

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

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

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

$$\therefore -y w^T x > 0$$

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

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

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

$$\Rightarrow \text{err}(w) \text{ of PLA is } \max(0, -y w^T x)$$

$$3. E(u+\Delta u, v+\Delta v)$$

by Taylor Series

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

$$\text{当 } \Delta(u, v) \rightarrow (0, 0)$$

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

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

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

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

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

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

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

当 y 擴展為 k -class, $y \in \{1, 2, \dots, k\}$

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

為猜 $y=1 \sim k$ 其中某一个的機率



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

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

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

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

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

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

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

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

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

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

$$\text{找 } \min_w \frac{A}{N+K} \Leftrightarrow \text{找 } \frac{\partial A}{\partial w} = 0 \quad \hat{=} \quad w$$

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

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

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

$$6. \quad w_{\text{reg}} = \arg \min_w \frac{\lambda}{N} \|w\|^2 + \frac{1}{N} \|Xw - y\|^2$$

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

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

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

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

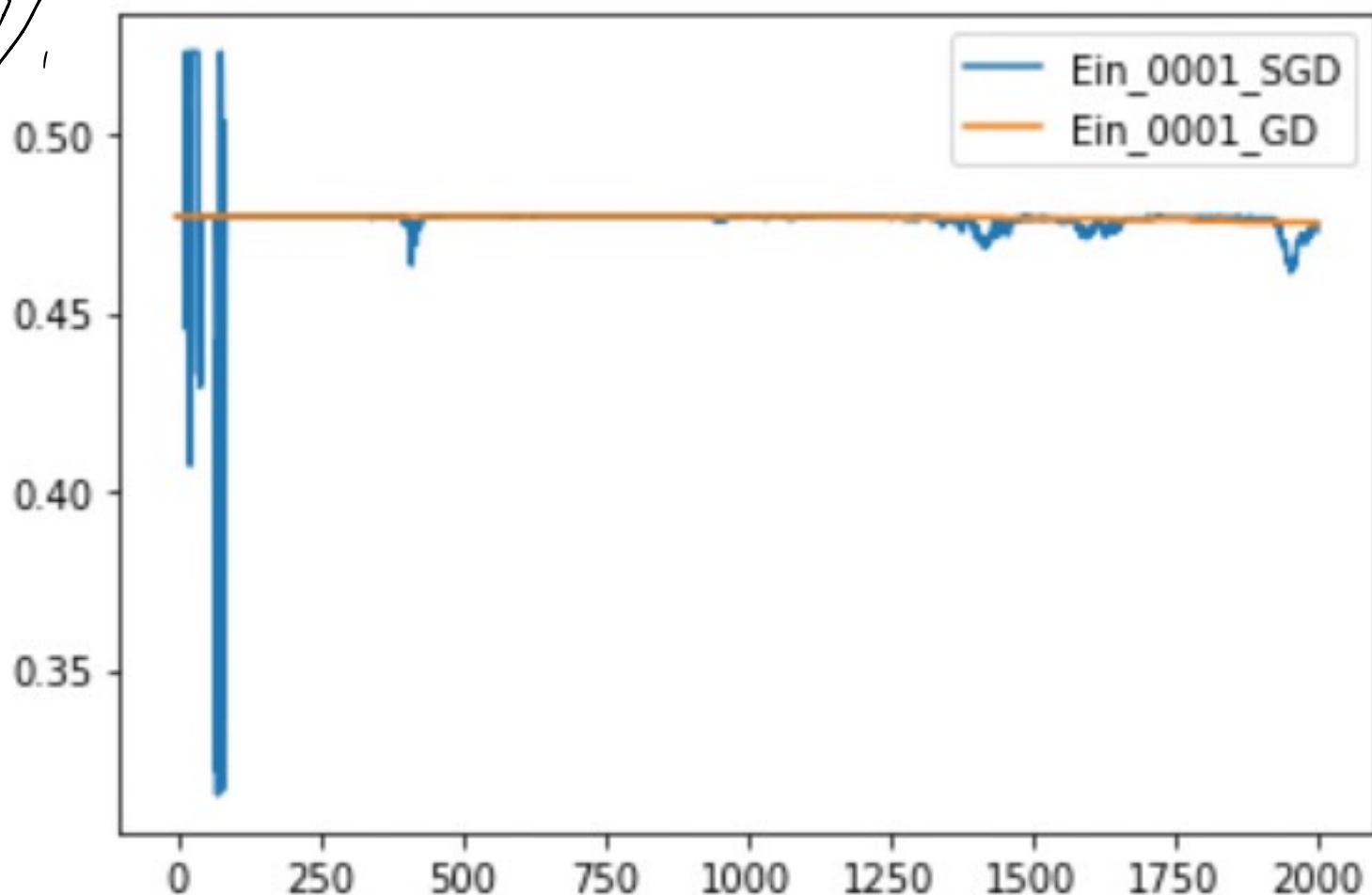
$$\Rightarrow w = (X^T X + \lambda I)^{-1} X^T y$$

$$= (X^T X + \tilde{X}^T \tilde{X})^{-1} (X^T y - \tilde{X}^T \tilde{y}) \quad (\text{by 前題})$$

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

※

7.



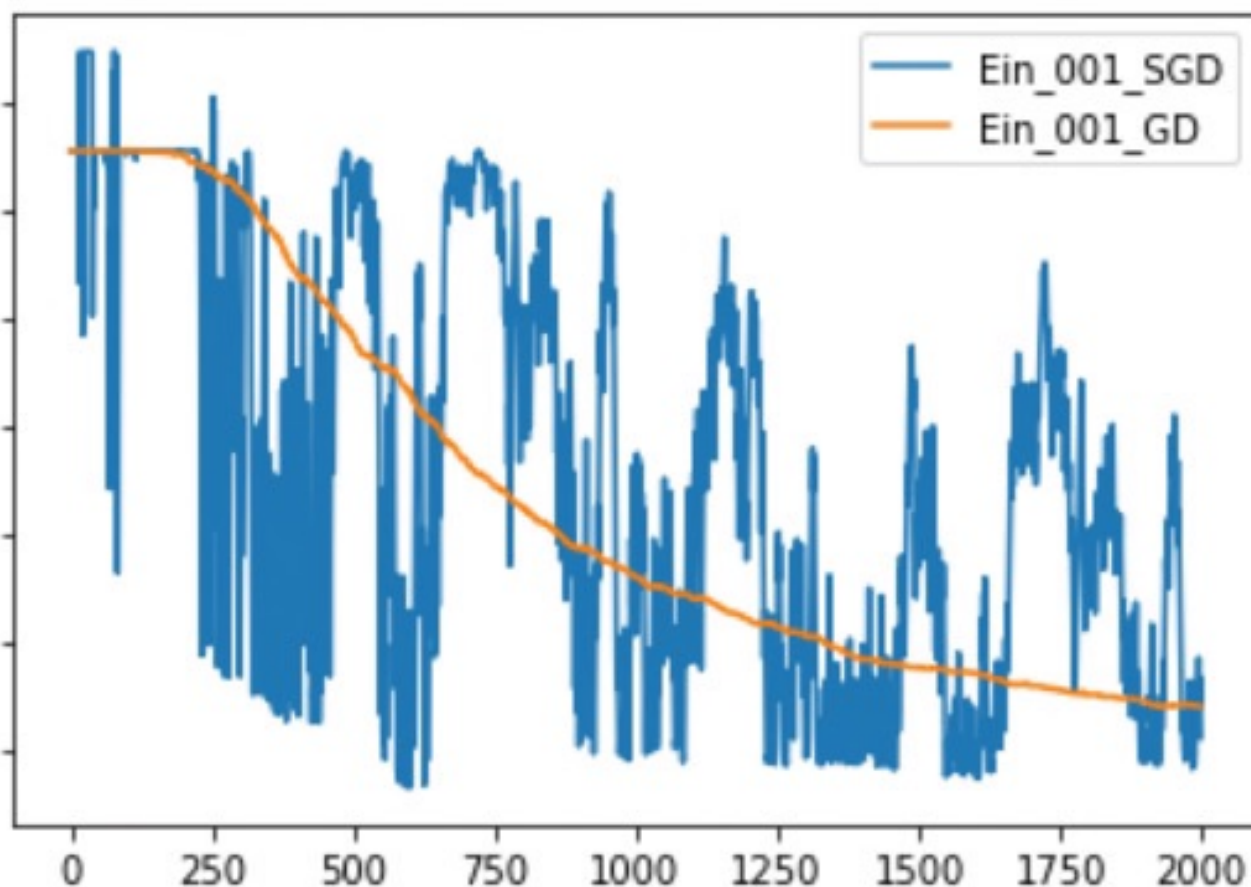
```

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 $E_{in}(wt)$ ，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。



```

eta = 0.01
=====GD=====
Ein = 0.197
Eout = 0.22
=====SGD=====
Ein = 0.187
Eout = 0.205333333333333334

```

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 E_{in} 都有顯著的下降，甚至 SGD 的 E_{in} 、 E_{out} 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 E_{in} 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 E_{in} 為 0.35，遠高於同樣更新次數的 GD 的 E_{in} ，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 E_{in} 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。