Homework #3 - Lei Lei

1. For a linear two-class classifier, show that $\underline{w}$ is orthogonal to the decision boundary H.
   Hint: if $\underline{x}_1$ and $\underline{x}_2$ are on the decision boundary, $g(\underline{x}_1) = g(\underline{x}_2) = 0$.

1. $\because$ $g(\underline{x})$ is a linear two-class classifier

$\therefore$ $g(\underline{x}) = w_0 + \underline{w}^T \underline{x}$

Let $\underline{x}_1$ and $\underline{x}_2$ are on the decision boundary.

$\therefore$ $g(\underline{x}_1) = g(\underline{x}_2) = 0$

$\therefore$ $g(\underline{x}_1) = w_0 + \underline{w}^T \underline{x}_1 = 0$

$g(\underline{x}_2) = w_0 + \underline{w}^T \underline{x}_2 = 0$

$\therefore$ $w_0 + \underline{w}^T \underline{x}_1 = w_0 + \underline{w}^T \underline{x}_2 = 0$

$\therefore$ $\underline{w}^T \underline{x}_1 - \underline{w}^T \underline{x}_2 = 0$

$\therefore$ $\underline{w}^T (\underline{x}_1 - \underline{x}_2) = 0$

$\because$ $\underline{x}_1, \underline{x}_2$ are on the decision boundary

$\therefore$ $\underline{x}_1 - \underline{x}_2$ is on the decision boundary

$\because$ $\underline{w}^T (\underline{x}_1 - \underline{x}_2) = 0$

$\therefore$ $\underline{w} \perp \underline{x}_1 - \underline{x}_2$

$\therefore$ $\underline{w}$ is orthogonal to the decision boundary H.

2. Let $p(\underline{x})$ be a scalar function of a D-dimensional vector $\underline{x}$, and $f(p)$ be a scalar function of p. Prove that:

$$\nabla_{\underline{x}} f\left[p(\underline{x})\right] = \left[\frac{d}{dp} f(p)\right] \nabla_{\underline{x}} p(\underline{x})$$

i.e., prove that the chain rule applies in this way. [**Hint:** you can show it for the i^{th} component of the gradient vector, for any i. It can be done in a couple lines.]

2. $\because \nabla_{\underline{x}} f[p(\underline{x})] = \frac{\partial f[p(\underline{x})]}{\partial x_i} = \frac{d f[p(\underline{x})]}{d p(\underline{x})} \cdot \frac{\partial p(\underline{x})}{\partial x_i}$

$\because$ For any i, $\frac{\partial f[p(\underline{x})]}{\partial x_i} = \frac{d f[p(\underline{x})]}{d p(\underline{x})} \cdot \frac{\partial p(\underline{x})}{\partial x_i}$

$\therefore \nabla_{\underline{x}} f[p(\underline{x})] = \begin{bmatrix} \frac{\partial f[p(\underline{x})]}{\partial x_1} \\ \vdots \\ \frac{\partial f[p(\underline{x})]}{\partial x_D} \end{bmatrix} = \frac{d f[p(\underline{x})]}{d p(\underline{x})} \cdot \begin{bmatrix} \frac{\partial p(\underline{x})}{\partial x_1} \\ \vdots \\ \frac{\partial p(\underline{x})}{\partial x_D} \end{bmatrix}$

$\therefore \nabla_{\underline{x}} f[p(\underline{x})] = \left[\frac{d}{dp} f(p)\right] \nabla_{\underline{x}} p(\underline{x})$

3. Find the following gradients:

a. $\nabla_{\underline{x}}\left(\underline{x}^T\underline{x}\right)$

b. $\nabla_{\underline{x}}\left[\left(\underline{x}^T\underline{x}\right)^3\right]$

c. $\nabla_{\underline{w}}||\underline{w}||_2^2$

d. $\nabla_{\underline{w}}\left\|\underline{w}\right\|_2$

e. $\nabla_{\underline{w}}||\underline{\underline{M}}\,\underline{w}||_2^2$

f. $\nabla_{\underline{w}}\left\|\underline{\underline{M}}\underline{w}-\underline{b}\right\|_2$

3. (a) Let $\underline{x}=(x_1\ x_2\ \cdots\ x_n)$

$\nabla_{\underline{x}}(\underline{x}^T\underline{x})=\frac{\partial\,\underline{x}^T\underline{x}}{\partial\underline{x}}=\frac{\partial}{\partial\underline{x}}(x_1^2+x_2^2+\cdots+x_n^2)$

$\because \frac{\partial}{\partial x_i}x_i^2=2x_i$

$\therefore \frac{\partial}{\partial\underline{x}}\cdot\underline{x}^T\underline{x}=2\underline{x}$

$\therefore \nabla_{\underline{x}}(\underline{x}^T\underline{x})=2\underline{x}$

(b) Let $f(p)=p^3\quad p(\underline{x})=\underline{x}^T\underline{x}$

According to the result from problem 2.

$\nabla_{\underline{x}}f[p(\underline{x})]=\frac{df(p)}{dp}\nabla_{\underline{x}}p(\underline{x})$

$\therefore \nabla_{\underline{x}}(\underline{x}^T\underline{x})^3=\frac{dp^3}{dp}\cdot\nabla_{\underline{x}}\underline{x}^T\underline{x}=3p^2\cdot 2\underline{x}=3(\underline{x}^T\underline{x})^2\cdot 2\underline{x}=6(\underline{x}^T\underline{x})^2\underline{x}$

$\therefore \nabla_{\underline{x}}(\underline{x}^T\underline{x})^3=6(\underline{x}^T\underline{x})^2\underline{x}$

(c) $\nabla_{\underline{w}}||\underline{w}||_2^2=\frac{\partial||\underline{w}||_2^2}{\partial\underline{w}}=\frac{\partial\,\underline{w}^T\underline{w}}{\underline{w}}=2\underline{w}$

(d) $\nabla_{\underline{w}}||\underline{w}||_2=\frac{\partial||\underline{w}||_2}{\partial\underline{w}}=\frac{\partial\sqrt{\underline{w}^T\underline{w}}}{\partial\underline{w}}$

Let $f(p)=p^{\frac{1}{2}}\quad p(\underline{w})=\underline{w}^T\underline{w}$

According to the result from problem 2.

$$\nabla_{\underline{w}} \|\underline{w}\|_2 = \frac{d\, p^{\frac{1}{2}}}{dp} \cdot \frac{\partial\, \underline{w}^T\underline{w}}{\partial \underline{w}} = \frac{1}{2} \cdot p^{-\frac{1}{2}} \cdot 2\underline{w}$$

$$= \frac{1}{2} \cdot (\underline{w}^T\underline{w})^{-\frac{1}{2}} \cdot 2\underline{w} = \frac{\underline{w}}{\|\underline{w}\|_2}$$

$$\therefore \nabla_{\underline{w}} \|\underline{w}\|_2 = \frac{\underline{w}}{\|\underline{w}\|_2}$$

(e) $$\nabla_{\underline{w}} \|\underline{\underline{M}}\underline{w}\|_2^2 = \frac{\partial\, (\underline{\underline{M}}\underline{w})^T(\underline{\underline{M}}\underline{w})}{\partial \underline{w}} = \frac{d(\underline{\underline{M}}\underline{w})^T(\underline{\underline{M}}\underline{w})}{d(\underline{\underline{M}}\underline{w})} \frac{\partial\, \underline{\underline{M}}\underline{w}}{\partial \underline{w}}$$

$$= 2(\underline{\underline{M}}\underline{w}) \cdot \underline{\underline{M}}$$

$$\therefore \nabla_{\underline{w}} \|\underline{\underline{M}}\underline{w}\|_2^2 = 2(\underline{\underline{M}}\underline{w})\underline{\underline{M}}$$

(f) $$\nabla_{\underline{w}} \|\underline{\underline{M}}\underline{x} - \underline{b}\|_2 = \frac{\partial \sqrt{(\underline{\underline{M}}\underline{w} - \underline{b})^T(\underline{\underline{M}}\underline{w} - \underline{b})}}{\partial \underline{w}}$$

Let $t(p) = \sqrt{p^T p}$ $\quad p(\underline{w}) = \underline{\underline{M}}\underline{w} - \underline{b}$

$$\therefore \nabla_{\underline{w}} \|\underline{\underline{M}}\underline{x} - \underline{b}\|_2 = \frac{d\sqrt{p^T p}}{dp} \cdot \frac{\partial(\underline{\underline{M}}\underline{w} - \underline{b})}{\partial \underline{w}}$$

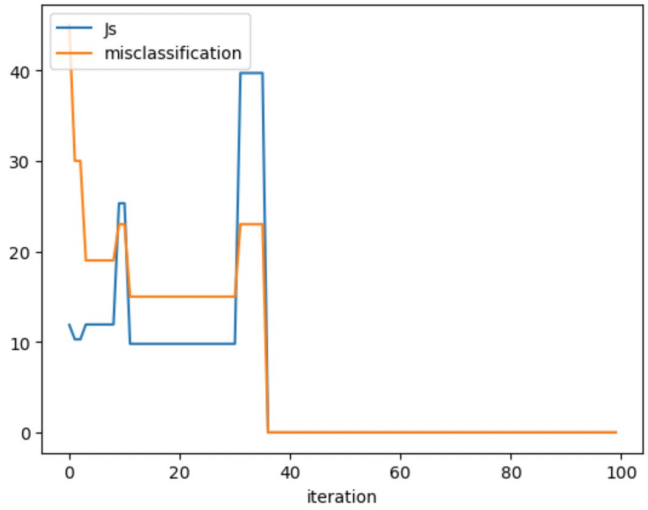
$$\frac{d\sqrt{p^T p}}{dp} = \frac{d\sqrt{p^T p}}{p^T p} \cdot \frac{\partial\, p^T p}{p} = \frac{1}{2}(p^T p)^{-\frac{1}{2}} \cdot 2p = \frac{p}{(p^T p)^{\frac{1}{2}}}$$

$$\therefore \nabla_{\underline{w}} \|\underline{\underline{M}}\underline{x} - \underline{b}\|_2 = \frac{p}{\|p\|_2} \cdot \underline{\underline{M}}$$

$$= \frac{\underline{\underline{M}}\underline{w} - \underline{b}}{\|\underline{\underline{M}}\underline{w} - \underline{b}\|_2} \cdot \underline{\underline{M}}$$

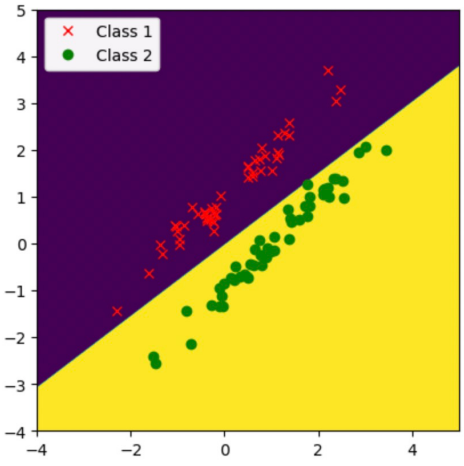
4. (a)Dataset1
(1) Sequential GD
i. i1 reach. Data is linearly separable
Weight matrix is: [0.1, -2.41605795,3.17014112]
Min J is: 0

ii.

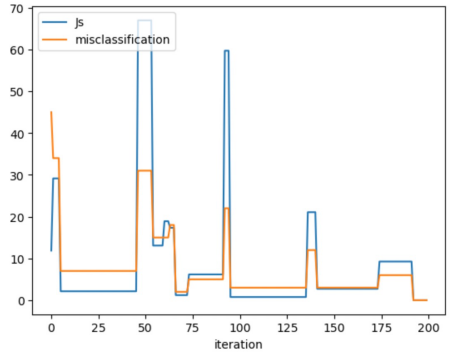


iii.
Error rate for train dataset 1: 0.00%
Error rate for test dataset 1: 0.00%

iv.



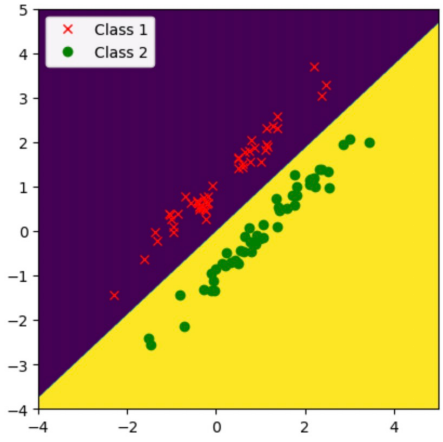
(2) SGD variant1
i. i1 reach. Data is linearly separable
Weight matrix is: [0.1 -3.20040359 4.41997576]
Min J is: 0
ii.



iii.

Error rate for train dataset 1: 0.00%
Error rate for test dataset 1: 0.00%

iv.



(b)Dataset2

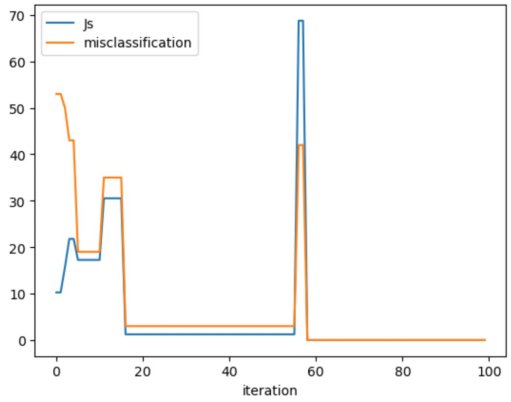
(1) Sequential GD

i. i1 reach. Data is linearly separable

Weight matrix is: [-0.9 -0.22588065 3.09050735]

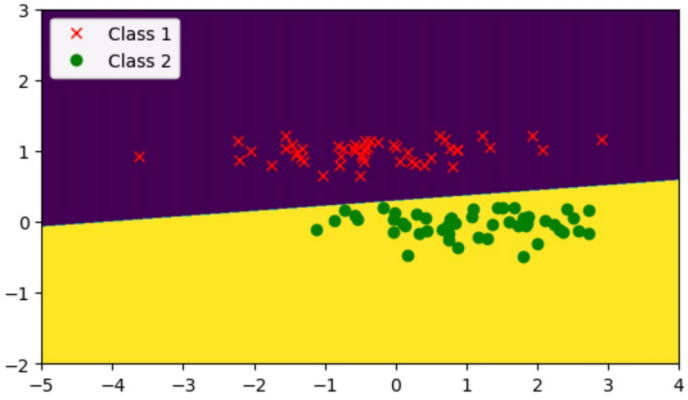
Min J is: 0

ii.

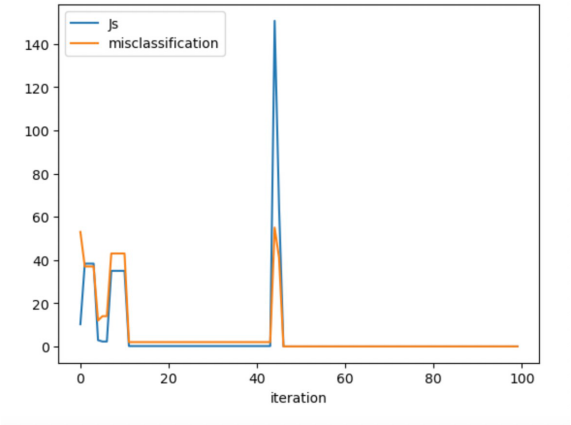


iii.
Error rate for train dataset 2: 0.00%
Error rate for test dataset 2: 2.00%

iv.

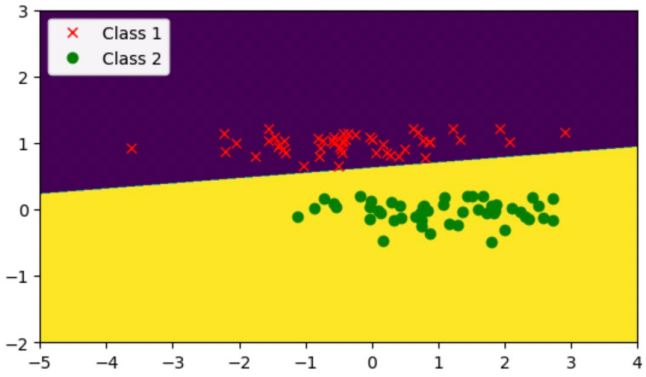


(2) SGD variant1
i. i1 reach. Data is linearly separable
Weight matrix is: [-1.9 -0.24078341 3.06009372]
Min J is: 0
ii.

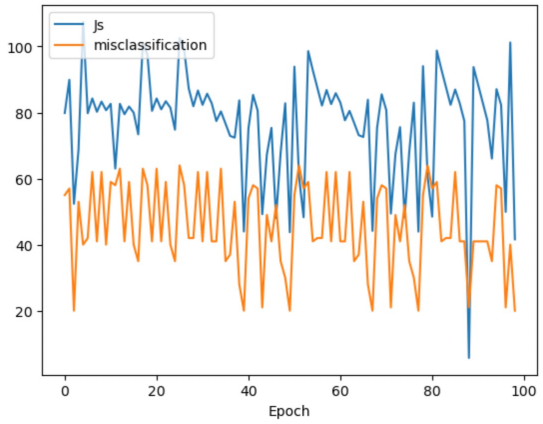


iii.
Error rate for train dataset 2: 0.00%
Error rate for test dataset 2: 3.00%

iv.

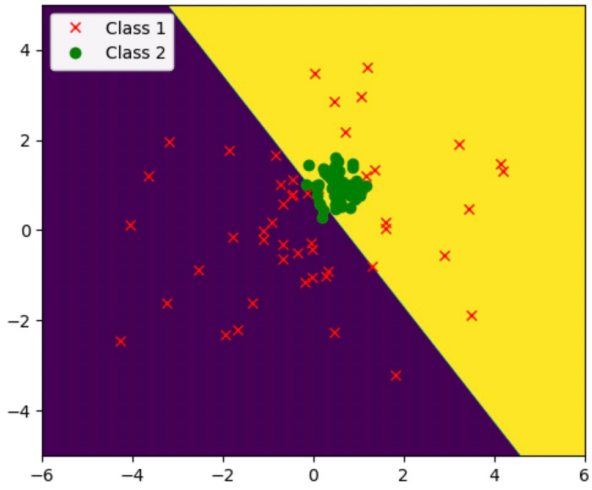


(c)Dataset3
(1) Sequential GD
i. i2 reach
Weight matrix is: [0.1 -0.14944965 -0.11672786]
Min J is: 5.643994017248471
ii.

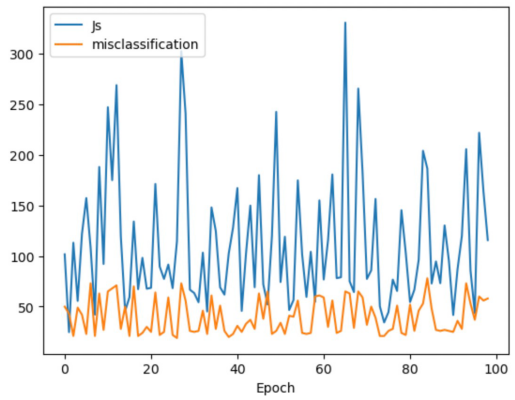


iii.
Error rate for train dataset 3: 23.00%
Error rate for test dataset 3: 25.00%

iv.

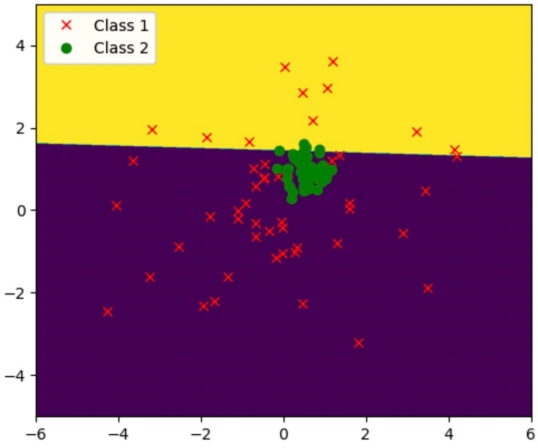


(2) SGD variant1
i. i2 reach
Weight matrix is: [0.1 -0.00202323 -0.06969695]
Min J is: 2.395592804014518
ii.



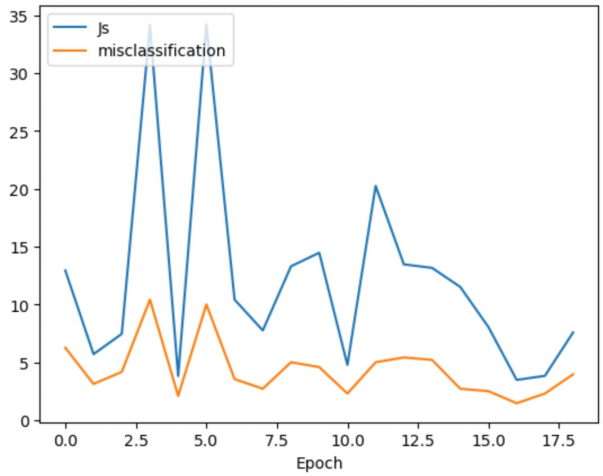
iii.
Error rate for train dataset 3: 65.00%
Error rate for test dataset 3: 68.00%

iv.



(d)BC data
(1) Sequential GD
i. i2 reach
Weight matrix is: [-9.9 0.5017636 6.49740265 0.62713144 3.19182841 -2.10446458
-1.14895794 6.11455833 6.04760966 0.0299059 -3.09329639 2.29076901
-0.11159452 -0.33217937 5.84373329 0.36845115 -3.53841119 -3.54586243
-0.96518394 2.30119468 -3.70253296 2.14138555 8.19515425 1.90574421
6.16414975 1.04743463 2.3097488 3.49591765 4.74253323 3.86528191
-0.24946351]
Min J is: 3.4741884847744737

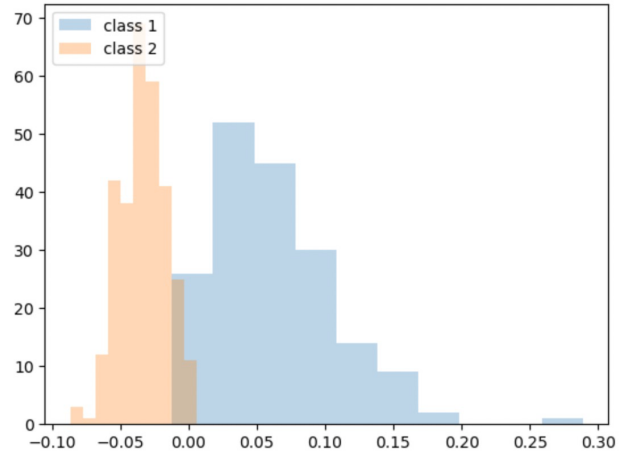
ii.



iii.
Error rate for train dataset : 1.4583333333333333 %
Error rate for test dataset : 4.49438202247191 %

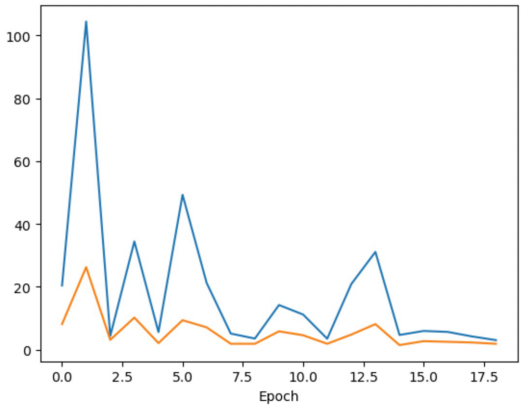
iv.

No, the data is not linearly separable because it can't be totally classified in 10000 iteration and there still exist some points across the boundary.



(2) SGD variant1
i. i2 reach
Weight matrix is: [-9.9 0.34538201 5.85956462 0.46216348 3.04439382 -2.2735228
-2.02382794 5.25627691 3.36506732 -0.16216568 -2.96547847 1.58327834
-0.57130751 0.27344041 5.72551543 0.27151865 -3.26369525 -3.09442752
-3.0815444 2.56545637 -4.04956398 2.471221 8.11519134 2.46610604
7.08892095 1.49724016 2.78612184 5.54279396 5.24020285 4.7215487
0.46984436]
Min J is: 2.7210562544169994

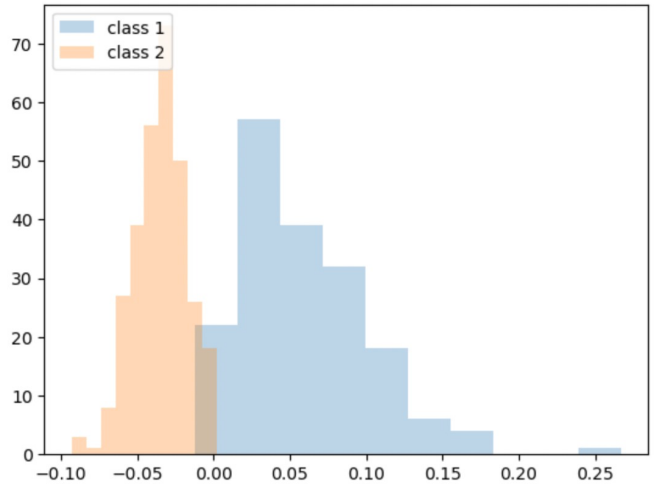
ii.



iii.

Error rate for train dataset : 1.6666666666666667 %
Error rate for test dataset : 3.3707865168539324 %

iv.



5. In lecture, we defined the criterion function for a 2-class Perceptron Learning problem (in augmented space) as:

$$J(\underline{w}) = -\sum_{n=1}^{N} [\![\underline{w}^T z_n \, \underline{x}_n \leq 0]\!] \, \underline{w}^T z_n \, \underline{x}_n$$

(a) Rewrite the criterion function using $max\{\cdot\}$ function and the rectified linear unit (ReLu) function defined by $ReLu(x) = max(0, x)$

(b) Consider replacing the ReLu(.) function with the softmax(.) function defined by $softmax(x) = ln(1 + exp(x))$ in this criterion function to produce a new criterion function $J_L(\underline{w})$. This results in logistic regression.

a. Plot ReLu(x) function with the softmax(x) vs x. Discuss the change in the criterion function when this softmax function is used. Specifically, for perceptron learning, the criterion function penalizes only errors with a loss proportional to the distance from the decision boundary. Is this still the case?

b. Find the gradient of $J_L(\underline{w})$. You may utilize the sigmoid function $\sigma(v) = e^v/(1 + e^v) = 1/(1 + e^{-v})$ in expressing the gradient.

(c) Repeat problem 4(d) (i.e., the breast cancer data) using this Logistic regression and compare the results to that obtained with perceptron learning.

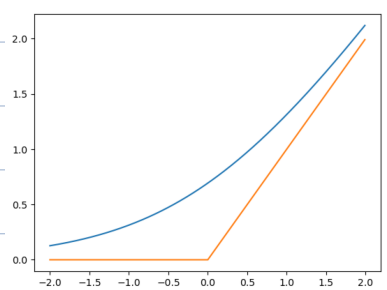
5. (a) set $[\![\underline{w}^T z_n \underline{x}_n \leq 0]\!]$ equal to 1 $[\![\underline{w}^T z_n \underline{x}_n > 0]\!]$ equal to 0

$$\therefore J(w) = \begin{cases} -\sum_{n=1}^{N} 0 \cdot \underline{w}^T z_n \underline{x}_n \\ -\sum_{n=1}^{N} 1 \cdot \underline{w}^T z_n \underline{x}_n \end{cases} = \begin{cases} 0 \\ -\sum_{n=1}^{N} \underline{w}^T z_n \underline{x}_n \end{cases}$$

$$\therefore J(w) = \sum_{n=1}^{N} \max(0, -\underline{w}^T z_n \underline{x}_n)$$

$$= \sum_{n=1}^{N} \text{Relu}(-\underline{w}^T z_n \underline{x}_n)$$

(b) a.



b. $J_L(w) = \sum_{n=1}^{N} \text{softmax}(-\underline{w}^T z_n \underline{x}_n)$

$$= \sum_{n=1}^{N} \ln(1 + \exp(-\underline{w}^T z_n \underline{x}_n)$$

$$\nabla J_L(\underline{w}) = \sum_{n=1}^{N} \nabla_{\underline{w}} \ln\left(1+\exp\left(-\underline{w}^T \underline{z}_n \underline{x}_n\right)\right)$$

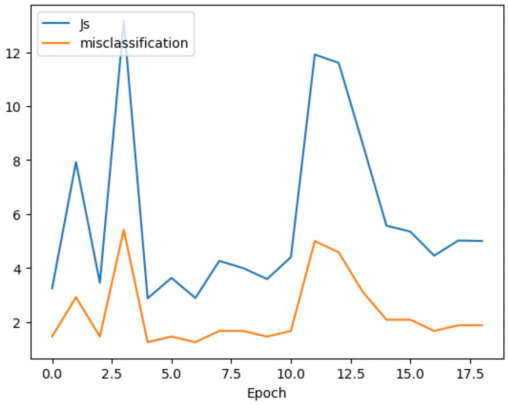
$$= \sum_{n=1}^{N} \left[\frac{1}{1+\exp\left(-\underline{w}^T \underline{z}_n \underline{x}_n\right)} \cdot \frac{\partial\left(1+\exp\left(-\underline{w}^T \underline{z}_n \underline{x}_n\right)\right)}{\partial \underline{w}} \right]$$

$$= \sum_{n=1}^{N} \left[\frac{1}{1+\exp\left(-\underline{w}^T \underline{z}_n \underline{x}_n\right)} \cdot \exp\left(-\underline{w}^T \underline{z}_n \underline{x}_n\right) \cdot \underline{z}_n \underline{x}_n \right]$$

$$= \sum_{n=1}^{N} \left[\sigma\left(-\underline{w}^T \underline{z}_n \underline{x}_n\right) \cdot \underline{z}_n \underline{x}_n \right]$$

5.(c)
Sequtial GD:
i. i2 reach
Weight matrix is: [-10.23434489 0.40360165 6.49344559 0.52599664 3.05415421
 -2.172346 -1.1638993 5.99377854 5.89871147 0.03638436
 -3.12607942 2.03257688 -0.44541542 -0.67006477 5.64161758
 0.14892003 -3.20418973 -3.56874944 -1.19027039 2.37559059
 -3.50258556 2.11994398 8.30089006 1.86583569 6.19116042
 1.1053723 2.84370102 3.80146479 4.8881845 4.20184781
 -0.01394843]
Min J is: 2.0739478716431825

ii.

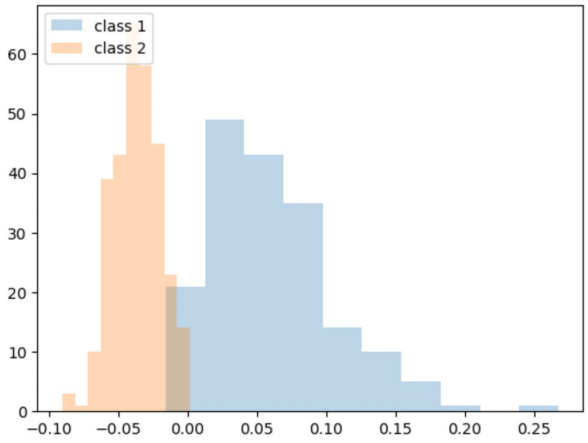


iii.

Error rate for train: 1.0416666666666665 %

Error rate for test: 3.3707865168539324 %

Iv:



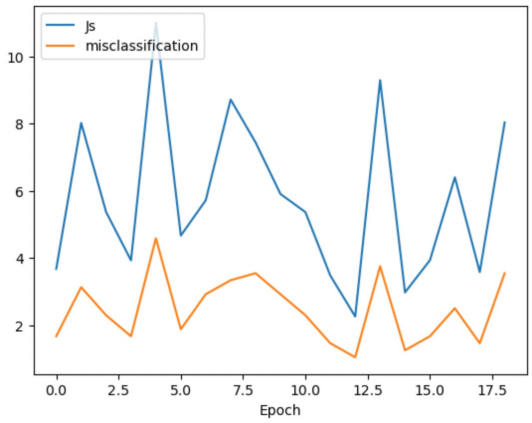
SGD variant 1

i. i2 reach

Weight matrix is: [-10.02345692 0.35188315 6.63721107 0.46105787 2.95586491
 -2.20427504 -1.72317507 5.64261971 5.21481623 0.23376044
 -3.18676316 2.00343102 -0.58377632 -0.59289793 5.81247235
  0.28742529 -3.19955481 -3.27488414 -1.41328204 2.3167574
 -3.7074916   2.20742895 8.64000369 1.97033964 6.40868743
  1.2968597   2.63911008 4.01138392 4.93147716 4.45201502
  0.01783195]

Min J is: 2.253363585563639

ii.

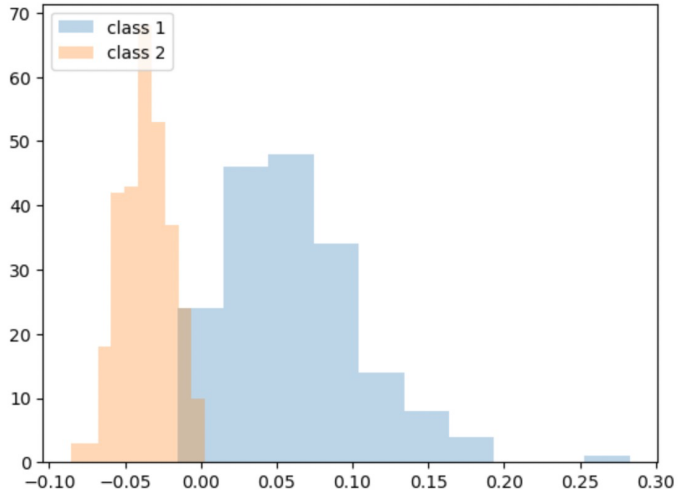


iii.

Error rate for train: 1.0416666666666665 %

Error rate for test: 3.3707865168539324 %

Iv:



Conclusion: the performance of Logistic regression is a little bit much better than the perceptron learning.