BME 548L Homework 2

Libo Zhang (NetID: lz200)
Email: libo.zhang@duke.edu

Problem 1 (a):

$$X = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{matrix} (X_1) \\ (X_2) \\ (X_3) \end{matrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}_{3\times1} = \begin{bmatrix} +1 \\ -1 \\ +1 \end{bmatrix} \quad W = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}_{3\times1}$$

I use np.linalg.pinv(X) to compute the pseudo-inverse, I get

$$X^{+} = \begin{bmatrix} 3.3333e-01 & 3.3333e-01 & 3.3333e-01 \\ -5.0000e-01 & -3.1352e-17 & 5.0000e-01 \\ 0.0000e+00 & 0.0000e+00 & 0.0000e+00 \end{bmatrix} \approx \begin{bmatrix} 0.33 & 0.33 & 0.33 \\ -0.5 & 0 & +0.5 \\ 0 & 0 & 0 \end{bmatrix}_{3\times3}$$

$$W = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix} = X^{+}Y = X^{+}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0.33 & 0.33 & 0.33 \\ -0.5 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}_{3\times3} \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}_{3\times1}$$

$$\therefore W = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix}_{3\times1} = \begin{bmatrix} 0.33 \\ 0 \\ 0 \end{bmatrix}_{3\times1}. \quad \begin{cases} b = 0.33 \text{ (offset term)} \\ w_1 = 0 \\ w_2 = 0 \end{cases}$$

Problem 1(b):

$$\hat{X} = \begin{array}{c} \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}_{3\times3} \\ (x_1) \ (x_2) \ (x_3) \end{array}$$

we know from 1(a) that

$$W = \begin{bmatrix} b \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} a_{33} \\ 0 \\ 0 \end{bmatrix}, \quad \text{So } W^T = [a_{33} \ \ 0 \ \ 0]_{1\times3}$$

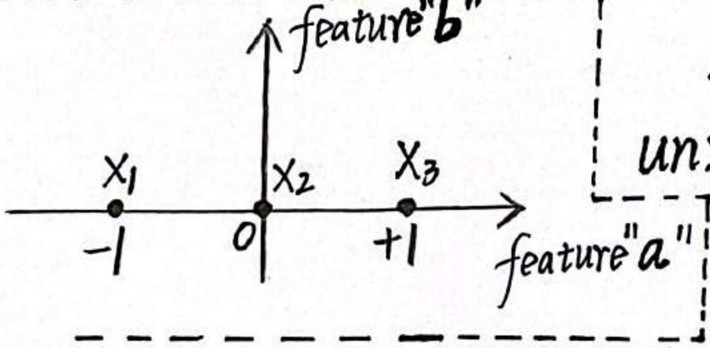
$$y^* = [y_1^* \ \ y_2^* \ \ y_3^*]_{1\times3} = \text{sign}(W^T \hat{X})$$

$$= \text{sign}\left([a_{33} \ \ 0 \ \ 0]_{1\times3} \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}_{3\times3} \right)$$

$$= \text{sign}([a_{33} \ \ a_{33} \ \ a_{33}]_{1\times3}) = [+1 \ \ +1 \ \ +1]_{1\times3}$$

$\therefore \ y^* = [y_1^* \ \ y_2^* \ \ y_3^*] = [+1 \ \ +1 \ \ +1]$, $\begin{cases} y_1^* = y_1, \ y_3^* = y_3, \text{ Correct.} \\ y_2^* \neq y_2, \text{ Wrong.} \end{cases}$

2 predictions are correct, while 1 prediction is wrong, So the optimal weights W did not do well at classifying the original 3 data points.



I think the classification attempt is unsuccessful because 3 data points are on the same line of the first feature ("a"), and the second feature ("b") is not contributing any additional information at all.

Problem 1 (C):

$$W_1 = \begin{bmatrix} 3 & 0 & 0 \\ -1 & 3 & 0 \\ 0 & -1 & 3 \\ 0 & 0 & -1 \end{bmatrix}_{4\times3}$$

$W_1 x_1 = [3\ -4\ 1\ 0]^T$

$W_1 x_2 = [3\ -1\ 0\ 0]^T$

$W_1 x_3 = [3\ 2\ -1\ 0]^T$

Then we can build a new 3×4 data matrix $X_c \in \mathbb{R}^{3\times4}$ with each row having a data point.

$$\begin{array}{l} X_c = \begin{bmatrix} 3 & -4 & 1 & 0 \\ 3 & -1 & 0 & 0 \\ 3 & 2 & -1 & 0 \end{bmatrix}_{3\times4} \end{array} \begin{array}{l} (x_1) \\ (x_2) \\ (x_3) \end{array}$$

With the help of np.linalg.pinv(X_c), we can calculate the pseudo-inverse of X_c,

$$X_c^\dagger \text{[redacted]} \approx \begin{bmatrix} 0.06 & 0.11 & 0.16 \\ -0.15 & -0.004 & 0.145 \\ 0.044 & -0.011 & -0.066 \\ 0 & 0 & 0 \end{bmatrix}_{4\times3}$$

Then we can find the weights $W_2 \in \mathbb{R}^{4\times1}$ by $W_2 = X_c^\dagger \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}_{3\times1}$

$\therefore W_2$ [redacted] $\approx [0.10989, -0.00366, -0.01099, 0]^T$.

$$\therefore y^* = [y_1^*, y_2^*, y_3^*]_{1\times3} = \text{sign}(W_2^T(\hat{X}_c)) = \text{sign}(W_2^T \underbrace{(X_c)^T}_{\text{Column-based } x_1, x_2, x_3.})$$

$$= \text{sign}\left(W_2^T \begin{bmatrix} 3 & 3 & 3 \\ -4 & -1 & 2 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}_{4\times3} \right) = \text{sign}([0.33, 0.33, 0.33])$$

$$\therefore y^* = [y_1^*, y_2^*, y_3^*] = [+1, +1, +1] \Rightarrow \begin{cases} y_1^* = y_1, \text{ Correct.} \\ y_2^* \neq y_2, \text{ wrong.} \\ y_3^* = y_3, \text{ Correct.} \end{cases}$$

Therefore, it is still impossible (unable) to determine a W_2 that can correctly classify the three points.

Problem 1(d): $W_1 = \begin{bmatrix} 3 & 0 & 0 \\ -1 & 3 & 0 \\ 0 & -1 & 3 \\ 0 & 0 & -1 \end{bmatrix}_{4\times3}$ $X_c = \begin{array}{c} \begin{bmatrix} 3 & -4 & 1 & 0 \\ 3 & -1 & 0 & 0 \\ 3 & 2 & -1 & 0 \end{bmatrix} \end{array}\begin{array}{l} (x_1) \\ (x_2) \\ (x_3) \end{array}$ row-based

After adding ReLU(), compared with 1(c), we can build a new 3x4 data matrix $X_d \in \mathbb{R}^{3\times4}$ from X_c, with each row having a data point.

$X_d = \mathrm{ReLU}(X_c) = \begin{bmatrix} 3 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 \end{bmatrix}_{3\times4}\begin{array}{l} (x_1) \\ (x_2) \\ (x_3) \end{array}$ $f(x) = \mathrm{ReLU}(x) = \max(0, x)$.

Find the pseudo-inverse of X_d with np.linalg.pinv(X_d),

$X_d^+ \approx \begin{bmatrix} 0 & 0.33 & 0 \\ 0 & -0.5 & 0.5 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{4\times3}$

Then we can find the weights by

$W_2 = X_d^+ \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}_{3\times1}$, where $W_2 \in \mathbb{R}^{4\times1}$.

$\therefore W_2 \approx [-0.333, 1.0, 2.0, 0]^T$. Then we can perform classification task:

$y^* = [y_1^*, y_2^*, y_3^*] = \mathrm{sign}(W_2^T \mathrm{ReLU}(\hat{X}_d))$, $\hat{X}_d$ has column-based x_1, x_2, x_3

$= \mathrm{sign}(W_2^T \mathrm{ReLU}((X_d)^T))$

$= \mathrm{sign}(W_2^T \begin{bmatrix} 3 & 3 & 3 \\ 0 & 0 & 2 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{4\times3})$

$\approx \mathrm{sign}([1.0, -1.0, 1.0])$

$= [+1, -1, +1]$.

$\therefore \begin{cases} y_1^* = y_1 \\ y_2^* = y_2 \\ y_3^* = y_3 \end{cases}$ All classifications are Correct!

Therefore, here it is possible (able) to determine a W_2 that can accurately classify the 3 points contained in the columns of $\hat{X}$, and

$W_2 = [-0.333, 1, 2, 0]^T$.

$W_2 \in \mathbb{R}^{4\times1}$

Problem 2 (a):

Without loss of generality (WOLOG), we can assume that here we have m features in total, So that $\underline{w} \in \mathbb{R}^{m\times1}$, $\underline{x_n} \in \mathbb{R}$ [redacted] $(m\times1)$, and for all N data points $\underline{X} = \begin{bmatrix} \underline{x_1}^T \\ \underline{x_2}^T \\ \vdots \\ \underline{x_n}^T \\ \vdots \\ \underline{x_N}^T \end{bmatrix}$, $\underline{X} \in \mathbb{R}^{N\times m}$. In problem 2(a), to compute the gradient of $L_{in}(\underline{w}) = \frac{1}{N}\sum_{n=1}^{N} \ln(1+e^{-y_n \underline{w}^T \underline{x_n}})$, we start with scalar examples, so WOLOG, we can set $m=2$ here.

$$L = L_{in}(\underline{w}) = \frac{1}{N}\cdot\sum_{n=1}^{N} \ln(1+e^{(-y_n)\cdot(w_1\cdot x_{n1} + w_2\cdot x_{n2})}), \quad \underline{x_n} = [x_{n1} \ x_{n2}]^T$$ [redacted]

This is because $\underline{w} = [w_1 \ w_2 \cdots w_m]^T$, $\underline{w} \in \mathbb{R}^{m\times1}$, $m=2$ and

$\underline{x_n} = [x_{n1} \ x_{n2} \cdots x_{nm}]^T$, $\underline{x_n} \in \mathbb{R}$ [redacted] $^{m\times1}$, $m=2$.

$$\therefore \frac{\partial L}{\partial w_1} = \frac{1}{N}\cdot\sum_{n=1}^{N} \frac{e^{(-y_n)(w_1x_{n1}+w_2x_{n2})}}{1+e^{(-y_n)(w_1x_{n1}+w_2x_{n2})}}\cdot(-y_n)\cdot(x_{n1})$$

$$\frac{\partial L}{\partial w_2} = \frac{1}{N}\cdot\sum_{n=1}^{N} \frac{e^{(-y_n)(w_1x_{n1}+w_2x_{n2})}}{1+e^{(-y_n)(w_1x_{n1}+w_2x_{n2})}}\cdot(-y_n)\cdot(x_{n2})$$

Consider $\theta(x) = \frac{e^x}{1+e^x}$, then we can have simpler forms below

$$\frac{\partial L}{\partial w_1} = \frac{1}{N}\sum_{n=1}^{N} (-y_n)\cdot(x_{n1})\cdot\theta[(-y_n)(w_1x_{n1}+w_2x_{n2})]$$

$$\frac{\partial L}{\partial w_2} = \frac{1}{N}\sum_{n=1}^{N} (-y_n)(x_{n2})\cdot\theta[(-y_n)(w_1x_{n1}+w_2x_{n2})]$$

(Please continue to the next page for 2(a))

Problem 2 (a): $\underline{w} = [w_1\ w_2 \cdots w_m]^T$, $\underline{x_n} = [x_{n1}\ x_{n2} \cdots x_{nm}]^T$.

Clearly, if we extend the previous calculation to the m-th feature,

$$\frac{\partial L}{\partial w_m} = \frac{1}{N}\sum_{n=1}^{N} (-y_n)(x_{nm}) \cdot \theta[(-y_n)\cdot(w_1 x_{n1} + w_2 x_{n2} + \cdots + w_m \cdot x_{nm})]$$

and then we write all the scalar equations in a vectorized form,

$$\frac{\partial L}{\partial \underline{w}} = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \vdots \\ \frac{\partial L}{\partial w_m} \end{bmatrix}_{m\times 1} = \begin{bmatrix} \frac{1}{N}\sum_{n=1}^{N}(-y_n)(x_{n1})\cdot\theta(-y_n\,\underline{w}^T\underline{x_n}) \\ \frac{1}{N}\sum_{n=1}^{N}(-y_n)(x_{n2})\cdot\theta(-y_n\,\underline{w}^T\underline{x_n}) \\ \text{[redacted]}\quad \vdots \\ \frac{1}{N}\sum_{n=1}^{N}(-y_n)(x_{nm})\cdot\theta(-y_n\cdot\underline{w}^T\underline{x_n}) \end{bmatrix}_{m\times 1} = \nabla_{\underline{w}} L_{in}(\underline{w}).$$

Finally we can have

$$\frac{\partial L}{\partial \underline{w}} = \nabla_{\underline{w}} L_{in}(\underline{w}) = \frac{1}{N}\sum_{n=1}^{N} -y_n \cdot \underline{x_n} \cdot \theta(-y_n \cdot \underline{w}^T \cdot \underline{x_n}),$$ where

$\theta(x) = \dfrac{e^x}{e^x+1}$, $\underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}_{m\times 1}$, and $\underline{x_n} = [x_{n1}\ x_{n2} \cdots x_{nm}]^T$ [redacted]

$\underline{x_n} = \begin{bmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{nm} \end{bmatrix}_{m\times 1}$ $\quad \underline{w} \in \mathbb{R}^{m\times 1}$, $\underline{x_n} \in \mathbb{R}^{m\times 1}$

Therefore proved.

Problem 2 (b):

Without loss of generality (WOLOG), we assume $m=2$, and an input data point has 2 features $\underline{x}_n = [x_{n1} \; x_{n2}]^T = [+1 \; +1]^T$.

The current weights are $\underline{w} = [w_1 \; w_2]^T = [\frac{1}{2} \; \frac{1}{2}]^T$, but the true label $y_n = -1$ (Assume we are doing Binary Classification between -1 and $+1$).

Now $\text{sign}(\underline{w}^T \underline{x}_n) = \underline{w}^T \underline{x}_n = \frac{1}{2} + \frac{1}{2} = +1$, so this is a misclassified input.

To facilitate calculation, WOLOG, we assume having only 1 data point, so $N = n = 1$. Then we can have

$$\left(\frac{\partial L}{\partial w_1}\right)_{wrong} = \frac{1}{1} \cdot \sum_{n=1}^{N=1} \frac{e^{(-(-1))(\frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 1)}}{1 + e^{(-(-1))(1\cdot\frac{1}{2} + 1\cdot\frac{1}{2})}} \cdot (-(-1)) \cdot x_{n1}$$

$$= \frac{e^1}{1+e^1} \cdot 1 \cdot 1 = \frac{e}{1+e}$$

$$\left(\frac{\partial L}{\partial w_2}\right)_{wrong} = \frac{1}{N} \cdot \sum_{n=1}^{N} \frac{e^{(-y_n)(w_1 x_{n1} + w_2 x_{n2})}}{1 + e^{(-y_n)(w_1 x_{n1} + w_2 x_{n2})}} \cdot (-y_n) \cdot x_{n2}$$

$$= \frac{1}{1} \cdot \sum_{n=1}^{N=1} \frac{e^{(-(-1))(\frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 1)}}{1 + e^{(-(-1))(\frac{1}{2}\cdot 1 + \frac{1}{2}\cdot 1)}} \cdot (-(-1)) \cdot (+1)$$

$$= \frac{e^1 \times 1 \times 1}{1+e^1} = \frac{e}{1+e}, \quad \text{So} \left(\frac{\partial L}{\partial \underline{w}}\right)_{wrong} = \begin{bmatrix} \frac{e}{1+e} \\ \frac{e}{1+e} \end{bmatrix}$$

Please continue to the next Page for 2 (b).

Problem 2(b): Now we have the gradient of a misclassified input, what about the gradient of a correctly classified input? We keep the weights $\underline{w}$ and features $\underline{x}_n$ the same but change the true label $y_n = -1$ to $y_n^* = +1 = \mathrm{sign}(\underline{w}^T \underline{x}_n) = \underline{w}^T \underline{x}_n$ (Assume we are doing Binary Classification between -1 and $+1$), so now the classification is correct, then we calculate the gradient.

$$\left(\frac{\partial L}{\partial w_1}\right)_{correct} = \frac{1}{1}\sum_{n=1}^{N=1} \frac{e^{(-(+1))(+1)}}{1+e^{(-(+1))(+1)}} \cdot (-(+1)) \cdot \overset{x_{n1}}{\uparrow}(+1) = \frac{e^{-1}}{1+e^{-1}} = \frac{1}{1+e}$$

$$\left(\frac{\partial L}{\partial w_2}\right)_{correct} = \frac{1}{1}\sum_{n=1}^{N=1} \frac{e^{(-(+1))(+1)}}{1+e^{(-(+1))(+1)}} \cdot (-(+1)) \cdot \overset{x_{n2}}{\uparrow}(+1) = \frac{e^{-1}}{1+e^{-1}} = \frac{1}{1+e}$$

$$\left(\frac{\partial L}{\partial w_1}\right)_{wrong} = \frac{e}{1+e} > \frac{1}{1+e} = \left(\frac{\partial L}{\partial w_1}\right)_{correct}$$

$$\left(\frac{\partial L}{\partial w_2}\right)_{wrong} = \frac{e}{1+e} > \frac{1}{1+e} = \left(\frac{\partial L}{\partial w_2}\right)_{correct}$$

$$e \approx 2.718.$$

This still holds true if we extend to multiple features ($m > 2$) and more data points ($N > 1$) because of WOLOG.

Therefore, we have proved that a misclassified input contributes more to the gradient than a correctly classified input.

Problem 2 (C) (bonus problem):

According to Lecture 7, we know that the optimal step size ε should be

(Assume we have m features here) $\varepsilon^* = \dfrac{\underline{g}^T \underline{g}}{\underline{g}^T \underline{H}\, \underline{g}}$, $\varepsilon^* \in \mathbb{R}$. $\begin{cases} \underline{g} \in \mathbb{R}^{m\times 1}, \underline{g}^T \in \mathbb{R}^{1\times m} \\ \underline{H} \in \mathbb{R}^{m\times m} \end{cases}$

ε^* is a scalar and $\underline{g} = \frac{\partial L}{\partial \underline{w}} = \nabla_{\underline{w}} L_{in}(\underline{w})$, $\underline{g} \in \mathbb{R}^{m\times 1}$ (assume m features)

And the Hessian of $L_{in}(\underline{w})$ with respect to $\underline{w}$ should be

$$\underline{H} = \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1m} \\ H_{21} & H_{22} & \cdots & H_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ H_{m1} & H_{m2} & \cdots & H_{mm} \end{bmatrix}_{m\times m} = \begin{bmatrix} \frac{\partial^2 L}{\partial w_1 \partial w_1} & \frac{\partial^2 L}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 L}{\partial w_1 \partial w_m} \\ \frac{\partial^2 L}{\partial w_2 \partial w_1} & \frac{\partial^2 L}{\partial w_2 \partial w_2} & \cdots & \frac{\partial^2 L}{\partial w_2 \partial w_m} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 L}{\partial w_m \partial w_1} & \frac{\partial^2 L}{\partial w_m \partial w_2} & \cdots & \frac{\partial^2 L}{\partial w_m \partial w_m} \end{bmatrix}_{m\times m}$$

where $H_{ij} = \frac{\partial^2 L}{\partial w_i \partial w_j}$, $1 \le i, j \le m$, the Hessian is also a symmetric matrix.

Since I have derived in 2 (a) that

$$\underline{g} = \frac{\partial L}{\partial \underline{w}} = \nabla_{\underline{w}} L_{in}(\underline{w}), \quad \underline{g}^T = \left[\frac{\partial L}{\partial \underline{w}}\right]^T, \quad \begin{cases} \underline{g} \in \mathbb{R}^{m\times 1} \\ \underline{g}^T \in \mathbb{R}^{1\times m} \end{cases}$$

As long as I can derive an equation for $H_{ij} \in \mathbb{R}$, then we can solve the optimal step size ε^*.

please continue to the next page for 2 (C).

Problem 2 (c): Since I need to use the Chain Rule to derive H_{ij}, I want to first work on $\frac{d[\theta(x)]}{dx}$, where $\theta(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$

$$\frac{d[\theta(x)]}{dx} = (-1)\cdot(1+e^{-x})^{-2}\cdot e^{-x}\cdot(-1) = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1+e^{-x}-1}{(1+e^{-x})^2}$$

$$= \frac{1+e^{-x}}{(1+e^{-x})^2} - \frac{1}{(1+e^{-x})^2} = \frac{1}{1+e^{-x}} - \frac{1}{(1+e^{-x})^2} = \theta(x)\cdot[1-\theta(x)]$$

$$\frac{\partial L}{\partial w_i} = \frac{1}{N}\sum_{n=1}^{N}(-y_n)\cdot x_{ni}\cdot\theta(-y_n\cdot \underline{w}^T\underline{x}_n),$$ So that

$$\frac{\partial^2 L}{\partial w_i\,\partial w_j} = \frac{1}{N}\sum_{n=1}^{N}(-y_n)\cdot x_{ni}\cdot\frac{\partial[\theta(-y_n\underline{w}^T\underline{x}_n)]}{\partial w_j} \quad ☆$$

$$☆\ \frac{\partial[\theta(-y_n\underline{w}^T\underline{x}_n)]}{\partial w_j} = \frac{\partial\{\theta[(-y_n)(w_1x_{n1}+w_2x_{n2}+w_j\cdot x_{nj}+\cdots+w_m\cdot x_{nm})]\}}{\partial w_j}$$

$$= \theta(-y_n\underline{w}^T\underline{x}_n)\cdot\left[1-\theta(-y_n\underline{w}^T\underline{x}_n)\right]\cdot\left[(-y_n)\cdot x_{nj}\right]$$

$$\frac{\partial^2 L}{\partial w_i\,\partial w_j} = \frac{1}{N}\sum_{n=1}^{N}\left(x_{ni}\cdot x_{nj}\cdot(y_n)^2\cdot\theta(-y_n\underline{w}^T\underline{x}_n)\left[1-\theta(-y_n\underline{w}^T\underline{x}_n)\right]\right)$$

$= H_{ij} \in \mathbb{R}$ (scalar value), $1 \le i, j \le m$.

Therefore, I have derived Hessian of $L_{in}(\underline{w})$ with respect to $\underline{w}$, next page I will summarize all the work to find the optimal step size ε^*.

Problem 2 (c): To summarize, the optimal Step Size ε^* is

$$\varepsilon^* = \frac{\underline{g}^T \underline{g}}{\underline{g}^T \underline{H}\ \underline{g}}, \quad \varepsilon^* \in \mathbb{R}.$$

where $\underline{g} = \frac{\partial L}{\partial \underline{w}} = \nabla_{\underline{w}} L_{in}(\underline{w}) = \frac{1}{N} \sum_{n=1}^{N} -y_n \cdot \underline{x}_n \cdot \theta(-y_n \underline{w}^T \underline{x}_n)$, $\underline{g} \in \mathbb{R}^{m\times 1}$

$$\underline{g}^T = \left(\frac{\partial L}{\partial \underline{w}}\right)^T = \left[\nabla_{\underline{w}} L_{in}(\underline{w})\right]^T, \quad \underline{g}^T \in \mathbb{R}^{1\times m}.$$

$$\underline{H} = \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1m} \\ H_{21} & H_{22} & \cdots & H_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ H_{m1} & H_{m2} & \cdots & H_{mm} \end{bmatrix}_{m\times m}, \quad H_{ij} = \frac{\partial^2 L}{\partial w_i \partial w_j}, \ 1 \le i, j \le m.$$

$\underline{H} \in \mathbb{R}^{m\times m}$, $H_{ij} \in \mathbb{R}$, and.

$$H_{ij} = \frac{1}{N} \sum_{n=1}^{N} \left\{ x_{ni} \cdot x_{nj} \cdot (y_n)^2 \cdot \theta(-y_n \underline{w}^T \underline{x}_n)\left[1 - \theta(-y_n \underline{w}^T \underline{x}_n)\right] \right\}$$

$$\underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_i \\ \vdots \\ w_j \\ \vdots \\ w_m \end{bmatrix}_{m\times 1} \qquad \underline{x}_n = [x_{n1}\ x_{n2} \cdots x_{ni} \cdots x_{nj} \cdots x_{nm}]^T$$

$$\theta(x) = \frac{e^x}{1+e^x}. \qquad \underline{X} = \begin{bmatrix} \underline{x}_1^T \\ \underline{x}_2^T \\ \vdots \\ \underline{x}_n^T \\ \vdots \\ \underline{x}_N^T \end{bmatrix}_{N\times m}$$

(This $\underline{X}$ vector/matrix is not used in all subproblems, but I still write it here just in case of any misunderstanding)

Expression for ε^* solved/found.

$$\left(\underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_i \\ \vdots \\ w_j \\ \vdots \\ w_m \end{bmatrix}_{m\times 1}, \ \underline{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{ni} \\ \vdots \\ x_{nj} \\ \vdots \\ x_{nm} \end{bmatrix}_{m\times 1}, \quad \underline{w} \in \mathbb{R}^{m\times 1}, \ \underline{x}_n \in \mathbb{R}^{m\times 1} \right)$$

Problem 3 (a):

Similar to problem 2, we can still assume, without loss of generality (WOLOG), that there are m features in total, so that we can have

$$\underline{w}(t) = [w_1(t)\ w_2(t)\ \cdots\ w_m(t)]^T,\quad \underline{w}(t) \in \mathbb{R}^{m\times1}$$

$$\underline{x}(t) = [x_1(t)\ x_2(t)\ \cdots\ x_m(t)]^T,\quad \underline{x}(t) \in \mathbb{R}^{m\times1}.$$

If one example is misclassified, we have

$$y(t) \neq \text{sign}[\underline{w}^T(t)\cdot\underline{x}(t)] = \text{sign}\Big[\sum_{i=1}^{m} w_i(t)\cdot x_i(t)\Big]$$

According to the problem, we only consider Binary Classification here,

$\therefore$ If $y(t) = +1$ ① $\Rightarrow y(t) = +1 \neq \text{sign}[\underline{w}^T(t)\cdot\underline{x}(t)] \Rightarrow$

$\underline{w}^T(t)\cdot\underline{x}(t) < 0 \Rightarrow y(t)\,\underline{w}^T(t)\cdot\underline{x}(t) < 0$ ✓

$\therefore$ If $y(t) = -1$ ② $\Rightarrow y(t) = -1 \neq \text{sign}[\underline{w}^T(t)\cdot\underline{x}(t)] \Rightarrow$

$\underline{w}^T(t)\cdot\underline{x}(t) > 0 \Rightarrow y(t)\cdot\underline{w}^T(t)\,\underline{x}(t) < 0$ ✓

Both labels $\begin{cases} y(t) = +1 \\ y(t) = -1 \end{cases}$ of Binary Classification have been shown,

So we have proved that $y(t)\,\underline{w}^T(t)\,\underline{x}(t) < 0$.

Problem 3(b): I will solve 3(b) using 2 proof methods.

Proof Method ① (vector-form).

Given the update rule $\underline{w}(t+1) = \underline{w}(t) + y(t)\cdot\underline{x}(t)$, we have

$$y(t)\cdot\underline{w}^T(t+1)\cdot\underline{x}(t) = y(t)\cdot\left[\underline{w}(t) + y(t)\,\underline{x}(t)\right]^T\cdot\underline{x}(t)$$

$$= y(t)\cdot\underline{w}^T(t)\cdot\underline{x}(t) + [y(t)]^2\cdot\underline{x}^T(t)\cdot\underline{x}(t), \quad \underline{x}(t)\in\mathbb{R}^{m\times 1},\ \underline{x}^T(t)\in\mathbb{R}^{1\times m}$$

$$= y(t)\cdot\underline{w}^T(t)\cdot\underline{x}(t) + [y(t)]^2\cdot\|\underline{x}(t)\|_2^2$$

where $\|\underline{x}(t)\|_2^2 = \sum_{i=1}^{m}[x_i(t)]^2$.

Since $y(t) = +1$ or (-1), and it will be meaningless if all features in $\underline{x}(t)$ are 0, so we have $y(t)\neq 0$ and $\underline{x}(t)\neq\underline{0}$,

$$\therefore\ [y(t)]^2\cdot\|\underline{x}(t)\|_2^2 > 0$$

$$\therefore\ y(t)\cdot\underline{w}^T(t+1)\cdot\underline{x}(t) - y(t)\cdot\underline{w}^T(t)\cdot\underline{x}(t) = [y(t)]^2\cdot\|\underline{x}(t)\|_2^2 > 0.$$

$$\therefore\ y(t)\cdot\underline{w}^T(t+1)\cdot\underline{x}(t) > y(t)\cdot\underline{w}^T(t)\cdot\underline{x}(t),$$ Proved.

Proof Method ② (scalar-form) is on the next page.

Problem 3 (b):

Proof Method ② (Scalar-form).

Given the update rule $\underline{w}(t+1) = \underline{w}(t) + y(t)\cdot\underline{x}(t)$, we have

$$\begin{bmatrix} w_1(t+1) \\ w_2(t+1) \\ \vdots \\ w_m(t+1) \end{bmatrix} = \begin{bmatrix} w_1(t) \\ w_2(t) \\ \vdots \\ w_m(t) \end{bmatrix} + \begin{bmatrix} y(t)\cdot x_1(t) \\ y(t)\cdot x_2(t) \\ \vdots \\ y(t)\cdot x_m(t) \end{bmatrix} = \begin{bmatrix} w_1(t) + y(t)\cdot x_1(t) \\ w_2(t) + y(t)\cdot x_2(t) \\ \vdots \\ w_m(t) + y(t)\cdot x_m(t) \end{bmatrix}_{m\times 1}$$

$$y(t)\cdot\underline{w}^T(t+1)\cdot\underline{x}(t) = y(t)\cdot\left[\sum_{i=1}^{m}\left(w_i(t) + y(t)\cdot x_i(t)\right)\cdot x_i(t)\right]$$

$$= y(t)\left[\sum_{i=1}^{m}\left(w_i(t)\cdot x_i(t)\right) + \sum_{i=1}^{m}\left(y(t)\cdot x_i(t)\cdot x_i(t)\right)\right]$$

$$= y(t)\left[\sum_{i=1}^{m} w_i(t)\cdot x_i(t)\right] + [y(t)]^2\cdot\sum_{i=1}^{m}[x_i(t)]^2$$

$$= y(t)\cdot\underline{w}^T(t)\cdot\underline{x}(t) + \underbrace{[y(t)]^2\cdot\|\underline{x}(t)\|_2^2}$$

>0 as discussed in Proof Method ①

$$\therefore\ y(t)\cdot\underline{w}^T(t+1)\cdot\underline{x}(t) - y(t)\cdot\underline{w}^T(t)\cdot\underline{x}(t) = [y(t)]^2\cdot\|\underline{x}(t)\|_2^2 > 0$$

$\therefore\ y(t)\cdot\underline{w}^T(t+1)\cdot\underline{x}(t) > y(t)\cdot\underline{w}^T(t)\cdot\underline{x}(t)$, Proved.

Problem 3 (C): Recall the update rule, we have

$\underline{w}^T(t+1) \cdot \underline{x}(t) = [\underline{w}(t) + y(t) \cdot \underline{x}(t)]^T \cdot \underline{x}(t)$

$= \underline{w}^T(t) \cdot \underline{x}(t) + y(t) \cdot \underline{x}^T(t) \cdot \underline{x}(t)$, $\underline{x}^T(t) \in \mathbb{R}^{1\times m}$, $\underline{x}(t) \in \mathbb{R}^{m\times 1}$

$= \underline{w}^T(t) \cdot \underline{x}(t) + y(t) \cdot \sum_{i=1}^{m} [x_i(t)]^2$

$= \underline{w}^T(t) \cdot \underline{x}(t) + y(t) \cdot \|\underline{x}(t)\|_2^2$

Since $(\underline{x}(t), y(t))$ is one "Currently missclassified" training data point, $\therefore$ $y(t) \neq \text{sign}[\underline{w}^T(t) \cdot \underline{x}(t)]$, then we can have

Case ①. If $y(t) = +1$, $\underline{w}^T(t)\underline{x}(t) < 0$, $y(t) \cdot \|\underline{x}(t)\|_2^2 > 0$.

Case ②. If $y(t) = -1$, $\underline{w}^T(t)\underline{x}(t) > 0$, $y(t) \cdot \|\underline{x}(t)\|_2^2 < 0$.

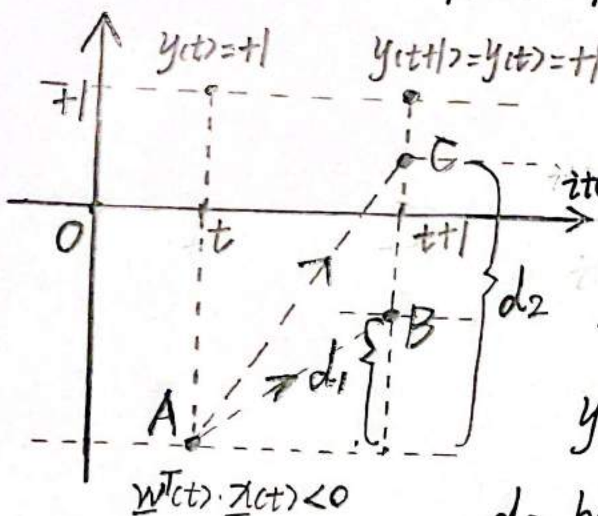
Since $\underline{w}^T(t+1) \cdot \underline{x}(t) = \underline{w}^T(t) \cdot \underline{x}(t) + y(t) \cdot \|\underline{x}(t)\|_2^2$, and since we know that the features $\underline{x}(t)$ and the label of 1 data point do not change while the number of iteration (t) increases,

$\therefore$ $y(t+1) = y(t)$, $\underline{x}(t+1) = \underline{x}(t)$ for this misclassified data point. Then in case ①, the updated $\underline{w}(t+1)$ is trying to move upwards with $y(t) \cdot \|\underline{x}(t)\|_2^2 > 0$. And in case ②, the updated $\underline{w}(t+1)$ is trying to move downwards with $y(t) \cdot \|\underline{x}(t)\|_2^2 < 0$. An example 2D picture is shown on the next page.

Problem 3 (c):

Case ①: $y(t) = y(t+1) = +1$, $\underline{w}^T(t) \cdot \underline{x}(t) < 0$, $y(t) \cdot \|\underline{x}(t)\|_2^2 > 0$,

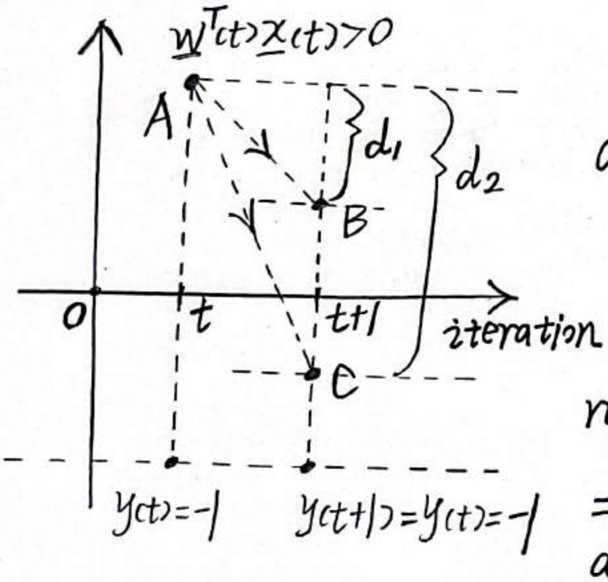
$$\underline{w}^T(t+1) \cdot \underline{x}(t) = \underline{w}^T(t+1) \cdot \underline{x}(t+1) = \underline{w}^T(t) \cdot \underline{x}(t) + y(t) \cdot \|\underline{x}(t)\|_2^2 .$$



Point A is where $\underline{w}^T(t)\underline{x}(t) < 0$ located, and $\underline{w}^T(t+1)\underline{x}(t)$ could be in Point B or C, which depends on the "moving upwards" strength of $y(t) \cdot \|\underline{x}(t)\|_2^2 = d_1$ or d_2, but both d_1 and d_2 belongs to a move in the right direction.

Case ②: $y(t) = y(t+1) = -1$, $\underline{w}^T(t)\underline{x}(t) > 0$, $y(t) \cdot \|\underline{x}(t)\|_2^2 < 0$,

$$\underline{w}^T(t+1) \cdot \underline{x}(t+1) = \underline{w}^T(t+1)\underline{x}(t) = \underline{w}^T(t)\underline{x}(t) + y(t) \cdot \|\underline{x}(t)\|_2^2 .$$



Point A is where $\underline{w}^T(t)\underline{x}(t) > 0$ located, and $\underline{w}^T(t+1)\underline{x}(t)$ could be in Point B or Point C, depending on the "moving downwards" strength of np.abs$(y(t) \cdot \|\underline{x}(t)\|_2^2) = \| y(t) \cdot \|\underline{x}(t)\|_2^2 \| = d_1$ or d_2, but both d_1 and d_2 belongs to a move in the right direction.

Therefore, we have proved that as far as classifying $\underline{x}(t)$ is concerned, moving from $\underline{w}(t)$ to $\underline{w}(t+1)$ is a move in the right direction.

Problem 4:

① Conv Layer 1. Stride = 2, Padding = 2, $\underline{x}_n \in \mathbb{R}^{1\times 4}$, $\underline{x}_n^T \in \mathbb{R}^{4\times 1}$.

$$\begin{bmatrix} c\langle1\rangle_1^1 & c\langle2\rangle_1^1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c\langle1\rangle_1^1 & c\langle2\rangle_1^1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c\langle1\rangle_1^1 & c\langle2\rangle_1^1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c\langle1\rangle_1^1 & c\langle2\rangle_1^1 \\ c\langle1\rangle_2^1 & c\langle2\rangle_2^1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c\langle1\rangle_2^1 & c\langle2\rangle_2^1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c\langle1\rangle_2^1 & c\langle2\rangle_2^1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c\langle1\rangle_2^1 & c\langle2\rangle_2^1 \end{bmatrix}_{8\times 8} \begin{bmatrix} 0 \\ 0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ 0 \\ 0 \end{bmatrix}_{8\times 1}$$

$PAD(\underline{x}_n^T) \in \mathbb{R}^{8\times 1}$

$\underline{W_1} \in \mathbb{R}^{8\times 8}$; $\underline{W_1}$ is stacked by two 2×1 conv filters.

$= \underline{W_1} \cdot PAD(\underline{x}_n^T)$

$\downarrow$ ReLU

$\underline{A} = [A_1\ A_2 \cdots A_8]^T$, $\underline{A} \in \mathbb{R}^{8\times 1}$

simplified-version results to save space for next layer.

② Sum-Pooling Layer 1. Stride = 2.

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}_{4\times 8} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ \hdashline A_5 \\ A_6 \\ A_7 \\ A_8 \end{bmatrix}_{8\times 1}$$

channel 1, conv 1.

channel 2, conv 1.

$= \underline{W_2} \cdot \underline{A} = \underline{B}$

$\underline{W_2} \in \mathbb{R}^{4\times 8}$, $\underline{B} \in \mathbb{R}^{4\times 1}$

$\underline{B}$ is the input to the second convolutional layer, it is stacked by 2 channels from conv 1.

$\underline{W_1}$ is the convolution matrix 1,

$\underline{W_2}$ is the sum-pooling matrix 1.

$$\underline{B} = \begin{bmatrix} B_1 \\ B_2 \\ \hdashline B_3 \\ B_4 \end{bmatrix}_{4\times 1} \begin{array}{l} \text{channel 1} \\ \\ \text{channel 2} \\ \end{array}$$

go to the next page for the following layers.

Problem 4:

③. Conv Layer 2. No Padding this conv layer, Stride = 2.

$$\begin{bmatrix} c\langle 1\rangle_1^2 & c\langle 2\rangle_1^2 & 0 & 0 \\ 0 & 0 & c\langle 1\rangle_1^2 & c\langle 2\rangle_1^2 \\ c\langle 1\rangle_2^2 & c\langle 2\rangle_2^2 & 0 & 0 \\ 0 & 0 & c\langle 1\rangle_2^2 & c\langle 2\rangle_2^2 \\ c\langle 1\rangle_3^2 & c\langle 2\rangle_3^2 & 0 & 0 \\ 0 & 0 & c\langle 1\rangle_3^2 & c\langle 2\rangle_3^2 \\ c\langle 1\rangle_4^2 & c\langle 2\rangle_4^2 & 0 & 0 \\ 0 & 0 & c\langle 1\rangle_4^2 & c\langle 2\rangle_4^2 \end{bmatrix}_{8\times 4} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix}_{4\times 1} = \underline{W_3}\,\underline{B} \xrightarrow{\text{ReLU}} [redacted]\ \underline{D}$$

$\underline{W_3}$ is the convolution matrix 2, and $\underline{W_3} \in \mathbb{R}^{8\times 4}$.

$\underline{W_3}$ is stacked by four 2×1 convolutional filters, and $\underline{D} \in \mathbb{R}^{8\times 1} = \mathbb{R}^{4\times(2\times 1)}$, and $\underline{D}$ is also stacked by 4 channels from conv 2.

④. Sum-pooling Layer 2, Stride = 2.

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}_{4\times 8} \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \\ D_7 \\ D_8 \end{bmatrix}_{8\times 1} = \underline{W_4}\,\underline{D} = \underline{E} \in \mathbb{R}^{4\times 1}$$

$$\underline{E} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{bmatrix} \begin{array}{l} \text{Channel 1} \\ \text{Channel 2} \\ \text{Channel 3} \\ \text{Channel 4} \end{array}$$

squeezed form of four channels from Conv Layer 2.

$\underline{W_4}$ is the Sum-pooling matrix 2.

$\underline{W_4} \in \mathbb{R}^{4\times 8}$

Go the next page for FC Layer and expression Summary.

Problem 4:

⑤ Fully-Connected Layer. 3 possible categories.

$$\begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \end{bmatrix}_{3\times4} \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ E_4 \end{bmatrix}_{4\times1} = \underline{W_5}\,\underline{E} \xrightarrow{ReLU} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}_{3\times1} = (\underline{y_n})^T$$

$\underline{W_5}$ is the fully-connected matrix.

Finally, if we take the transpose again $\left[(\underline{y_n})^T\right]^T = \left(\underline{y_n}^T\right)^T = \underline{y_n} \in \mathbb{R}^{1\times3}$,

So we have mapped from $\underline{x_n} = [x_1, x_2, x_3, x_4] \in \mathbb{R}^{1\times4}$ to

$\underline{y_n} = [y_1, y_2, y_3] \in \mathbb{R}^{1\times3}$.

To summarize, the final expression should be

$$\underline{y_n} = \left\{ ReLU\left(\underline{W_5}\,\underline{W_4}\,ReLU\,\underline{W_3}\,\underline{W_2}\,ReLU\,\underline{W_1}\left[PAD(\underline{x_n}^T)\right]\right)\right\}^T$$

With this matrix operation expression, the mapping between $\underline{x_n} \in \mathbb{R}^{1\times4}$ and $\underline{y_n} \in \mathbb{R}^{1\times3}$ can be successfully established.