

P1:

① $\vec{w} = 0\hat{i} + \hat{j} + \hat{k}$ ($\vec{a} = 0\hat{i} + 2\hat{j} + 2\hat{k}$)

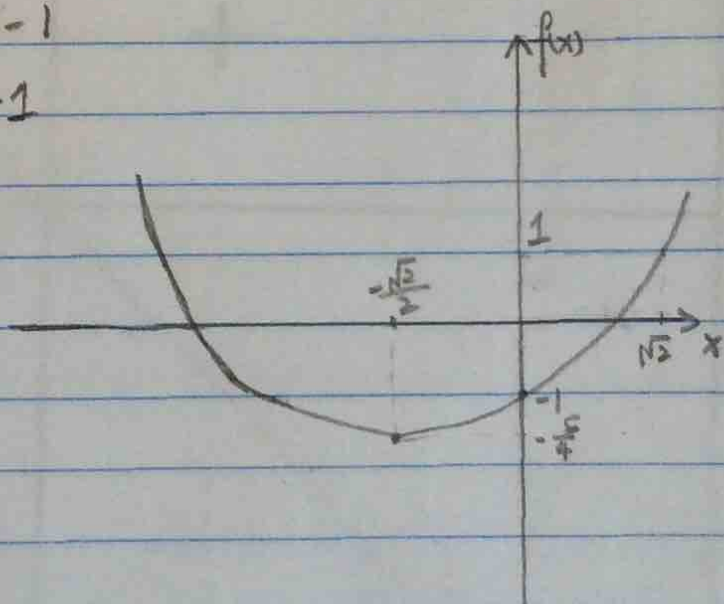
② $\gamma = \frac{1}{2} \cdot \sqrt{0^2 + 2^2 + 2^2} = \sqrt{2}$

③ $\gamma = \frac{1}{\|w\|} \Rightarrow w_1^2 + w_2^2 + w_3^2 = \left(\frac{1}{\gamma}\right)^2 = \left(\frac{1}{\sqrt{2}}\right)^2$

$$w_2 = w_3 = \frac{1}{2} \quad w_1 = 0 \quad \hat{w} = [0 \ \frac{1}{2} \ \frac{1}{2}]^T$$

④ $\begin{cases} \hat{w}_0 \leq -1 - w^T \phi(x_1) = -1 \\ \hat{w}_0 \geq 1 - w^T \phi(x_2) = -1 \end{cases}$
 $\therefore \hat{w}_0 = -1$

$$\begin{aligned} \text{5. } f(x) &= -1 + \frac{\sqrt{2}}{2}x + \frac{1}{2}x^2 \\ &= \frac{1}{2}\left(x + \frac{\sqrt{2}}{2}\right)^2 - \frac{5}{4} \end{aligned}$$



P2: ① ② on next page

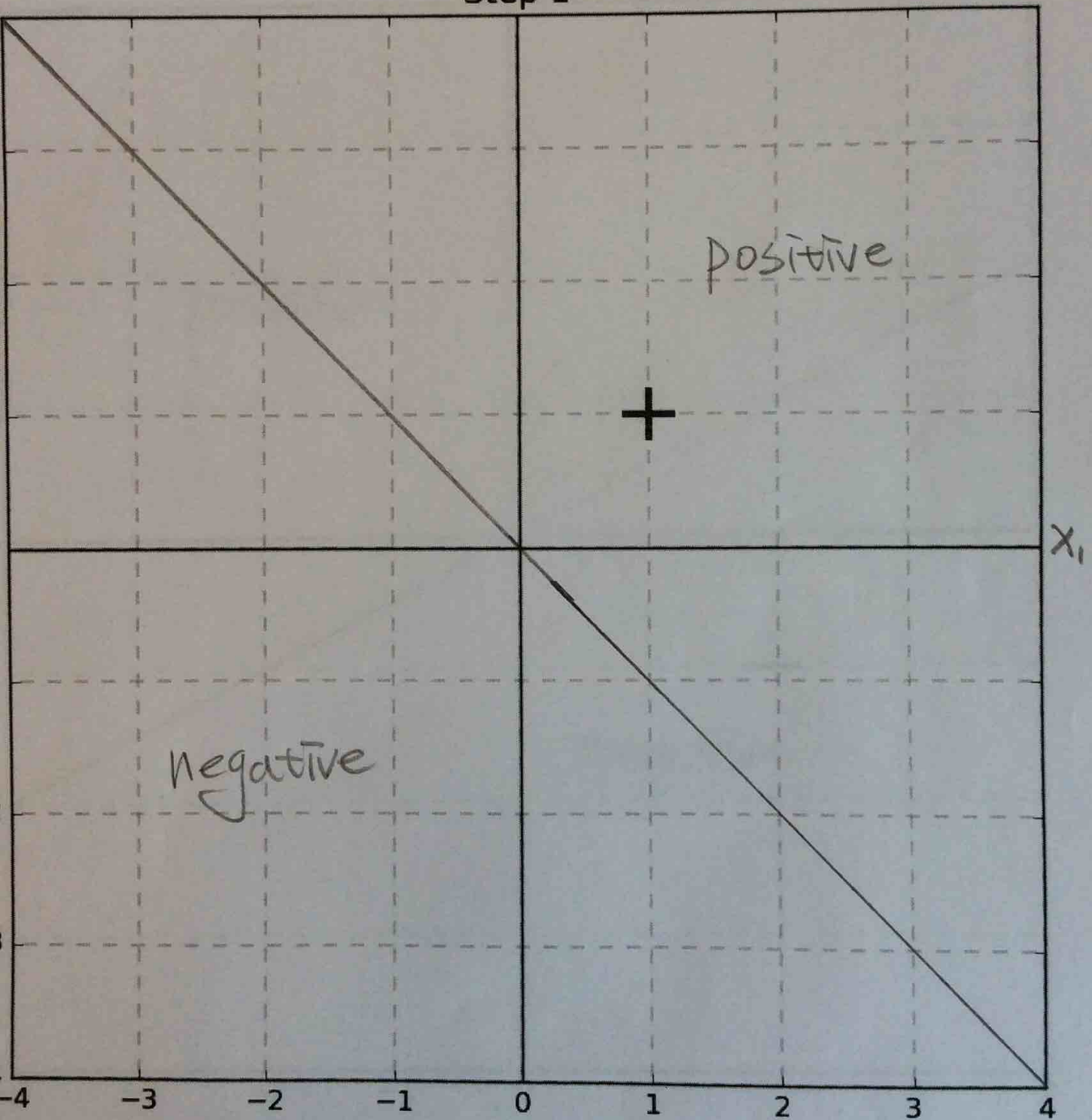
③ mistakes $\leq \frac{R^2}{\gamma^2} = \frac{25}{0.5^2} = 100$

$$100 - 4 = 96$$

96 mistakes allowed for future data

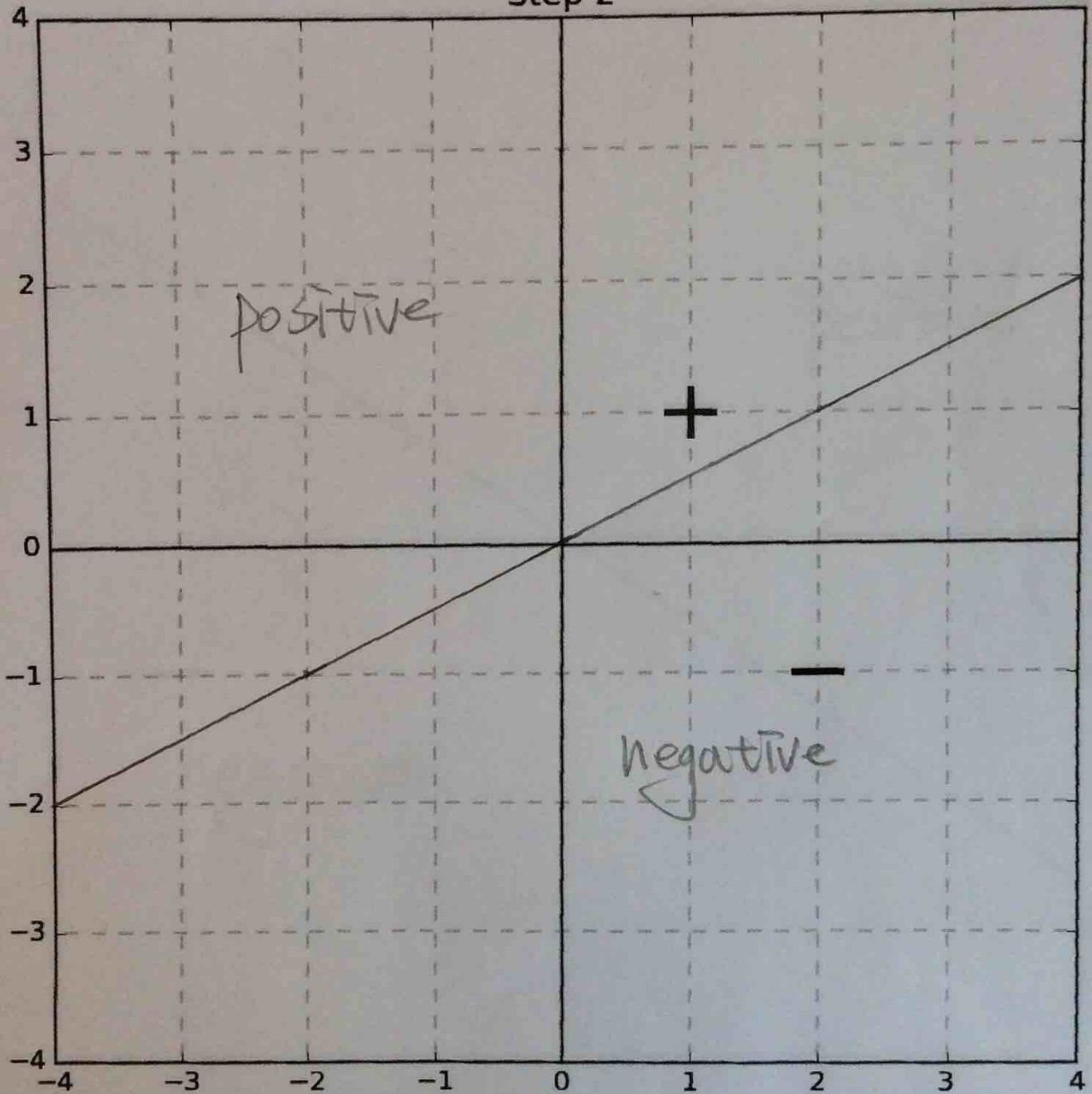
Equation: $x_1 + x_2 = 0$

Step 1

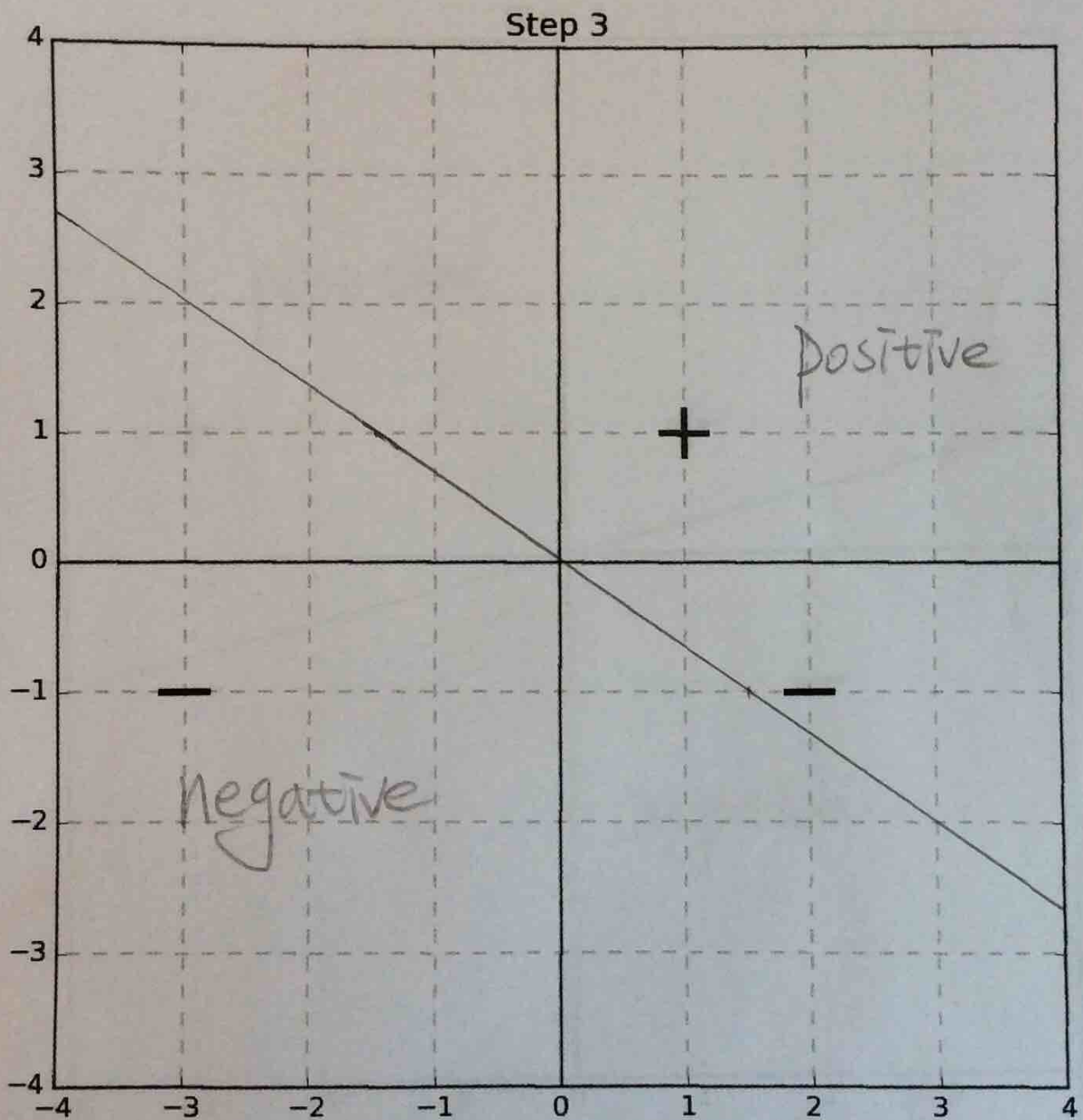


Equation: $X_1 - 2X_2 = 0$

Step 2

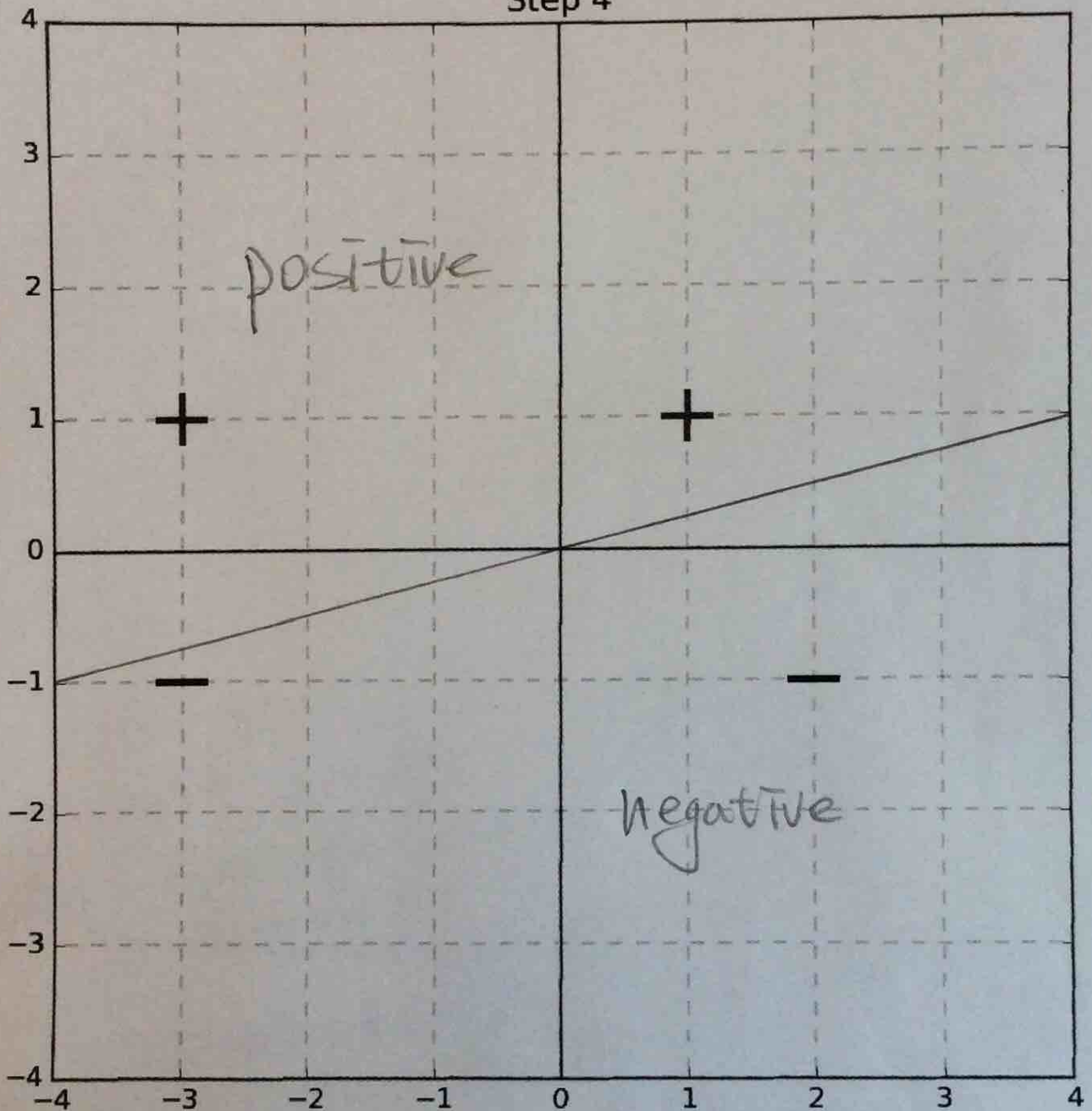


Equation: $2x_1 + 3x_2 = 0$



Equation: $X_1 - 4X_2 = 0$

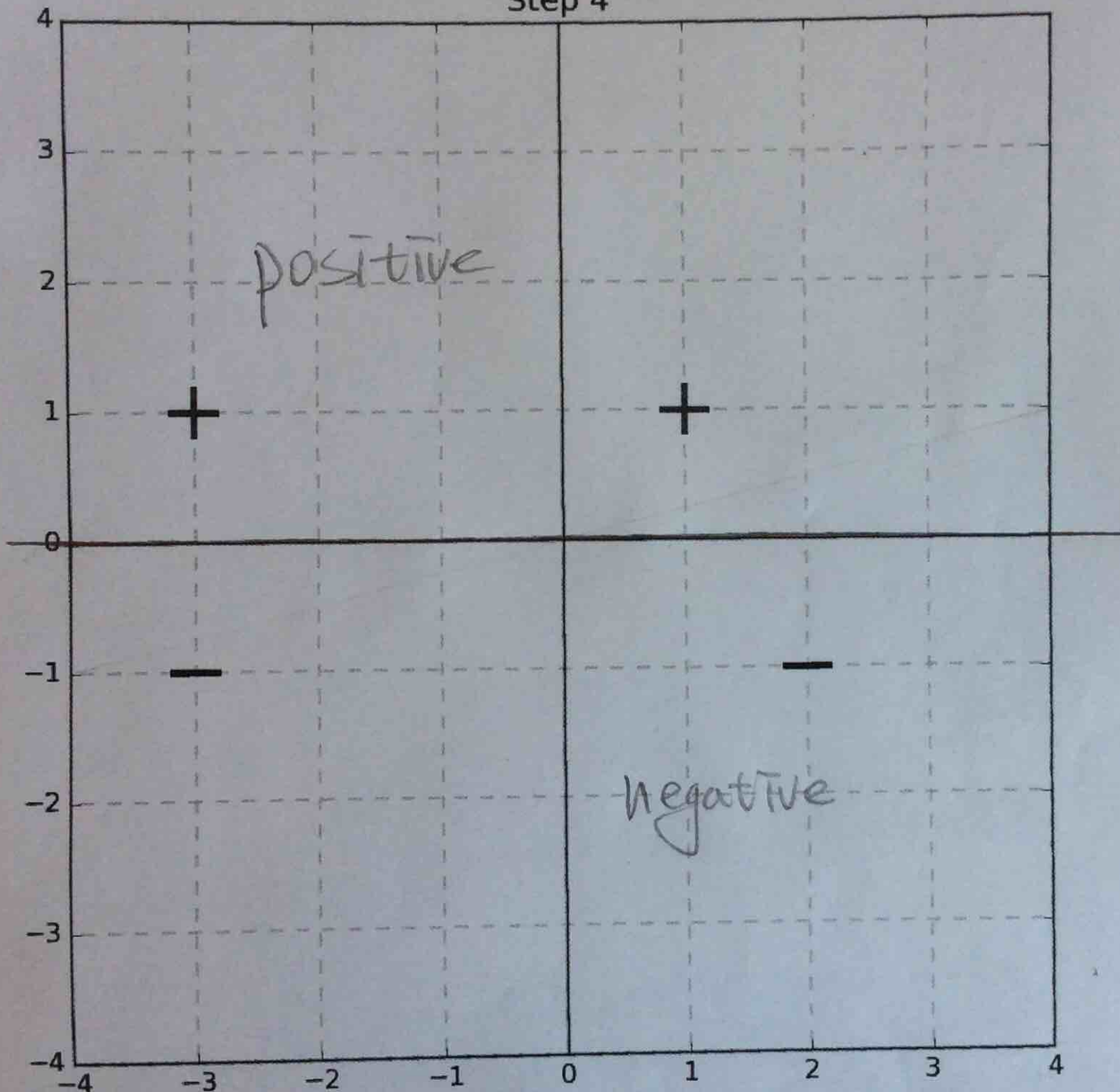
Step 4



part 1 does not maximize the margin

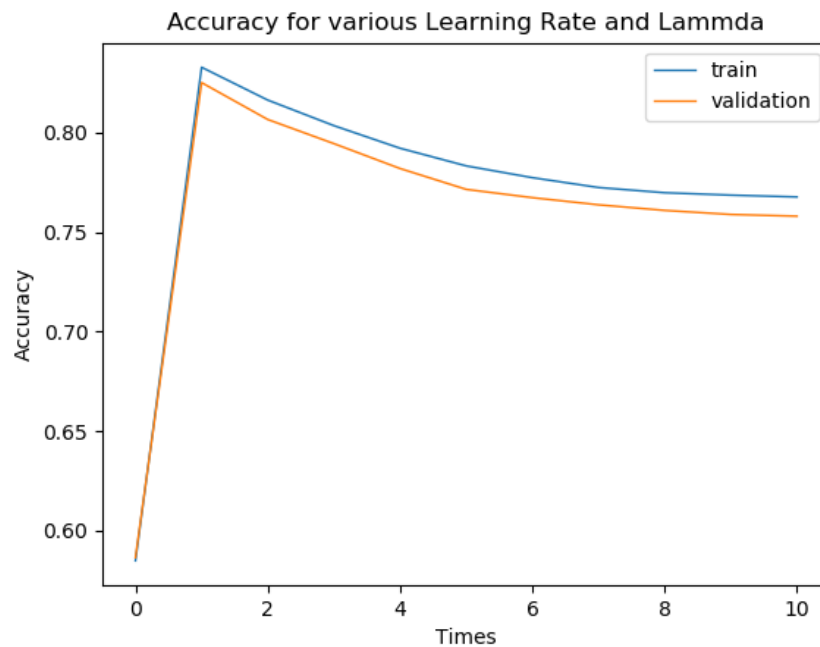
Equation: $X_2 = 0$

Step 4



Prat II. Programming

According to Learning rate vs Accuracy graph, we could find that when learning rate = 0.1, lammda = 0.001, the accuracy is best. The best accuracy is 0.8393169.



```
Python 3.6.4rc1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4rc1 (v3.6.4rc1:3398dcb, Dec 5 2017, 20:41:32) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\zhaoq\Desktop\EE511\HW#2\QIHAN_ZHAO_HW#2\test.py =====
....Training Accuracy: 0.8393169742944013
Validation Accuracy: 0.8393169742944013
.
-----
Ran 5 tests in 18.733s
OK
>>> |
```