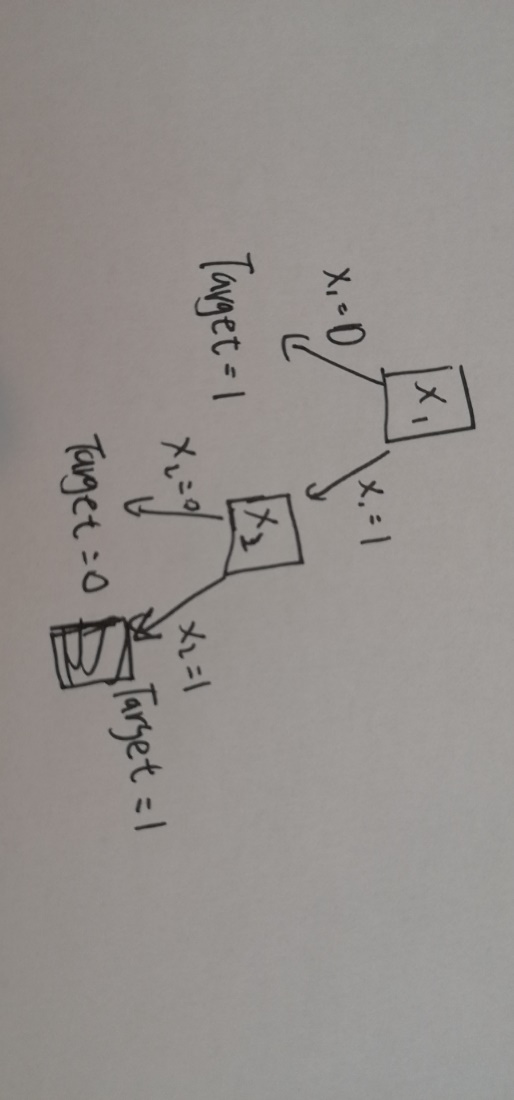
Z5305320

Dong AO

(a): Let’s say D represent this question data set. And X1, X2 and X3 represent the features.

(i):

The tree computed:



**The training error is** since I put dataset back to the tree, I find there is one piece of data is wrong and the total number of data is 4. Thus, it is **.**

I use and etc. to denote the second term in the formular of and etc.

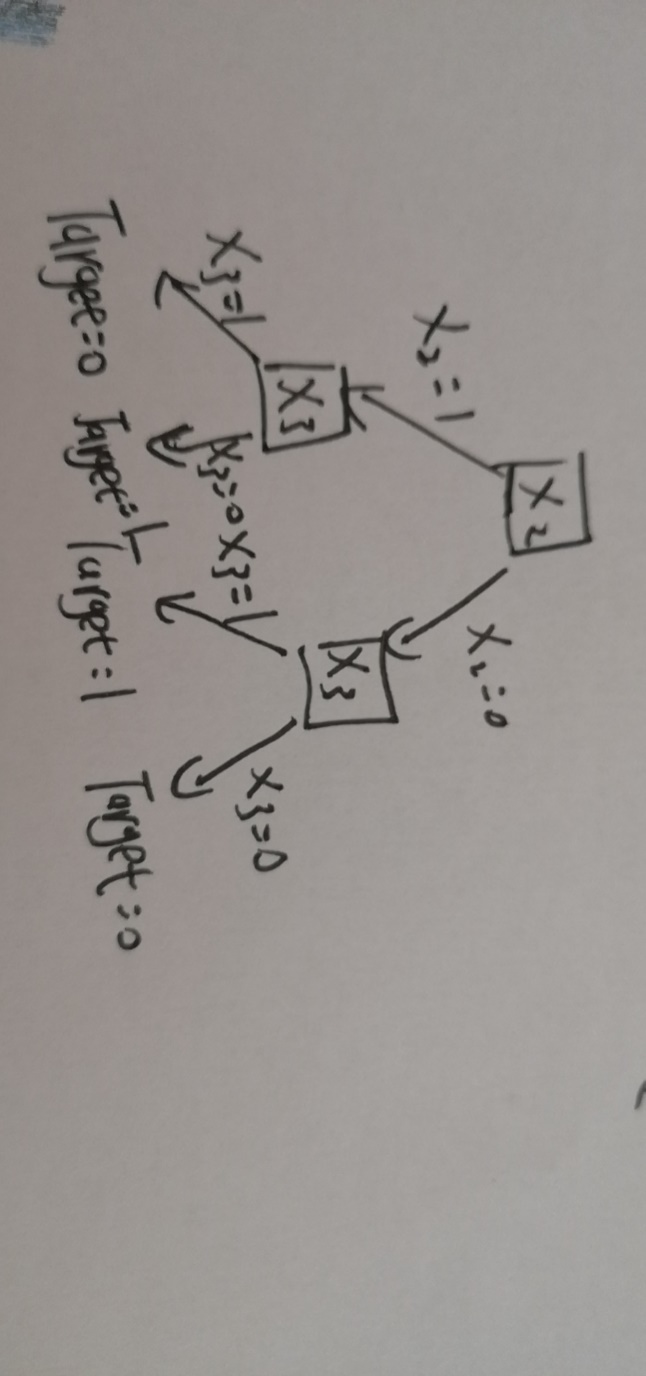
Since has the highest gain, I choose as tree root. Then, once again:

Then, since they have the same gain, I can randomly choose one between and .

(ii):

I try to put in the root and get the tree below, which has a 0 training error.

Decription: ID3 has the trend to avoid overfitting which appears in the tree below: it is doing better in training error but it is also overfitting over the dataset.



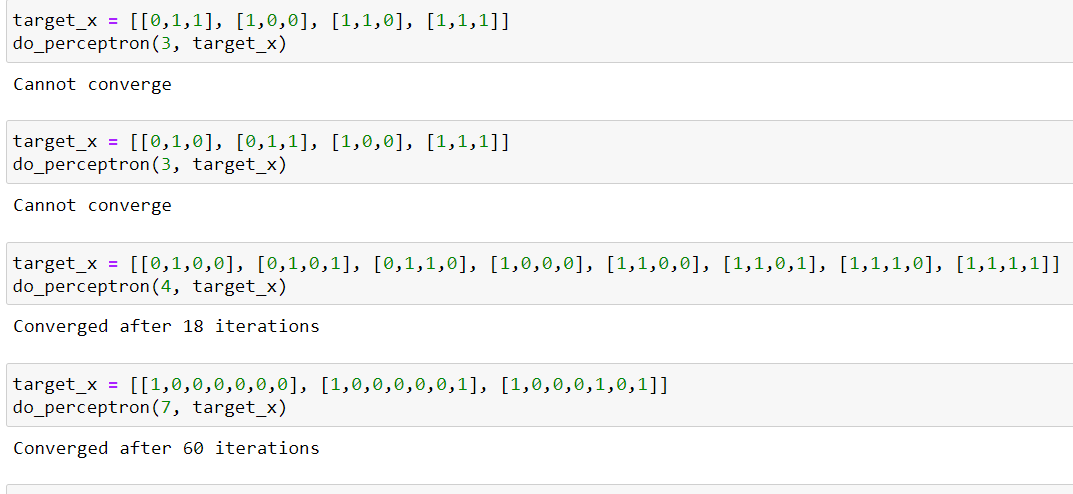
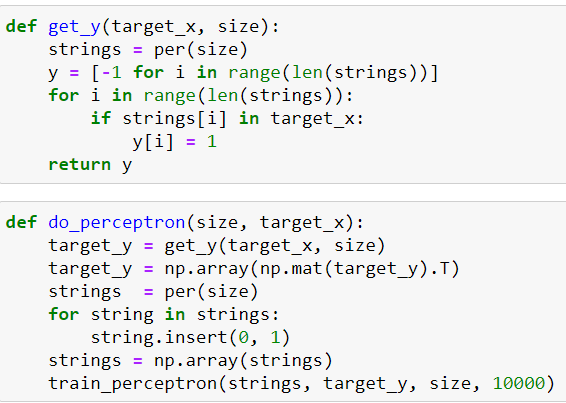
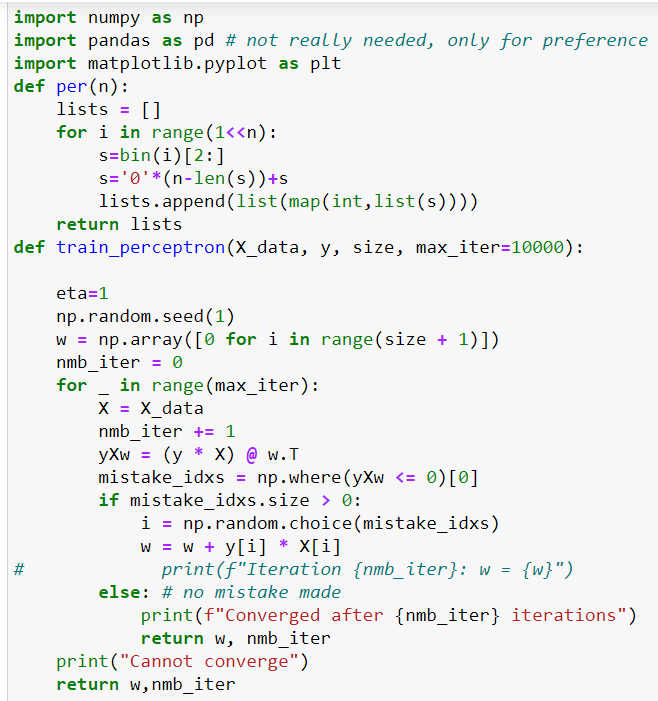
(b):

|  |  |
| --- | --- |
| Dataset | Linearly Separable (Yes/No) |
| (i) | No |
| (ii) | No |
| (iii) | Yes |
| (iv) | Yes |

Description:

1. I generate the correct data of X and the data of Y. Then I need to add a column of 1 to X because we need a bias term.

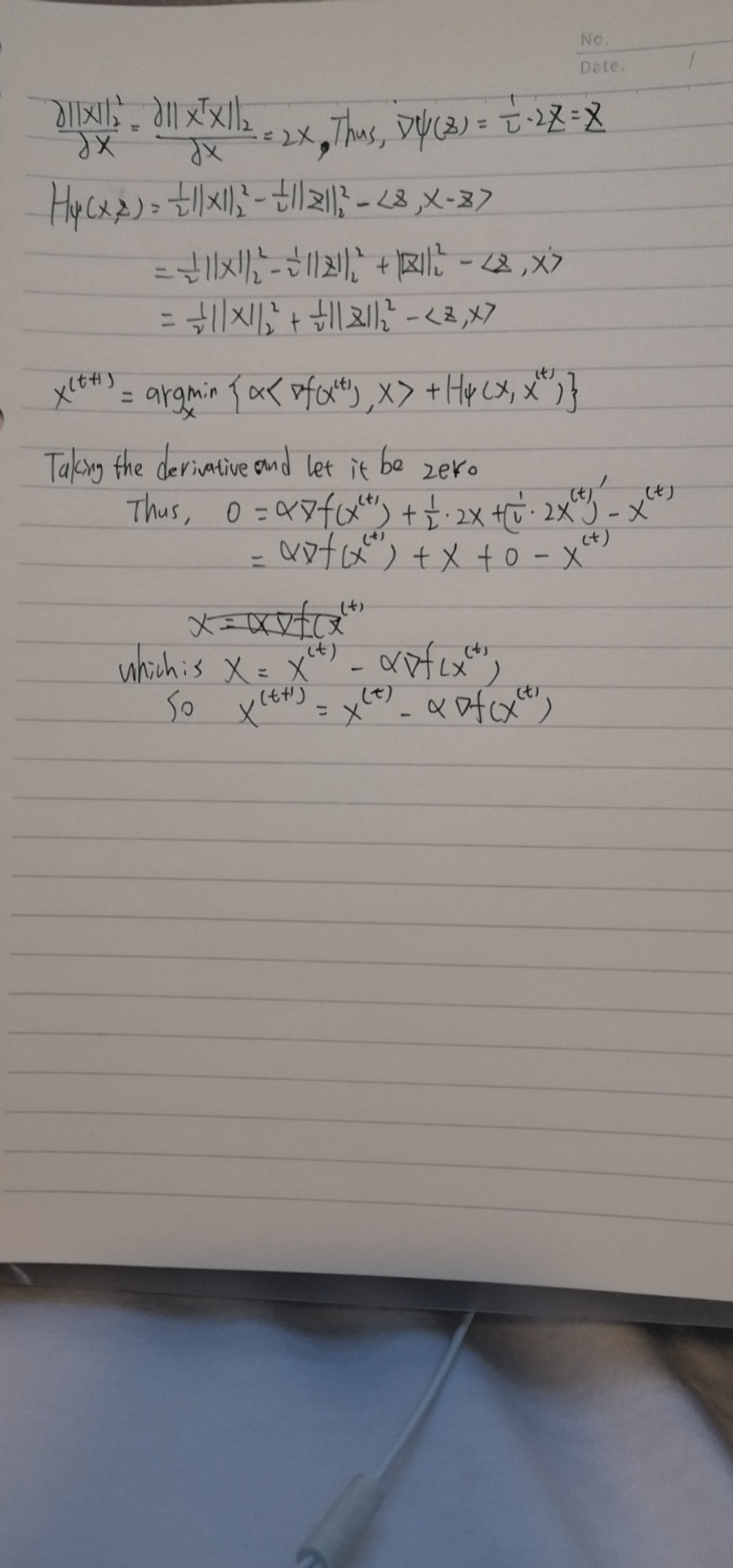
2. We use the standard perceptron: in each iteration, we calculate if there is a mistake, if yes, we try to update the weight with our produce of and . If there is nothing to update, it means we have achieved. Otherwise, it means the dataset cannot converge.



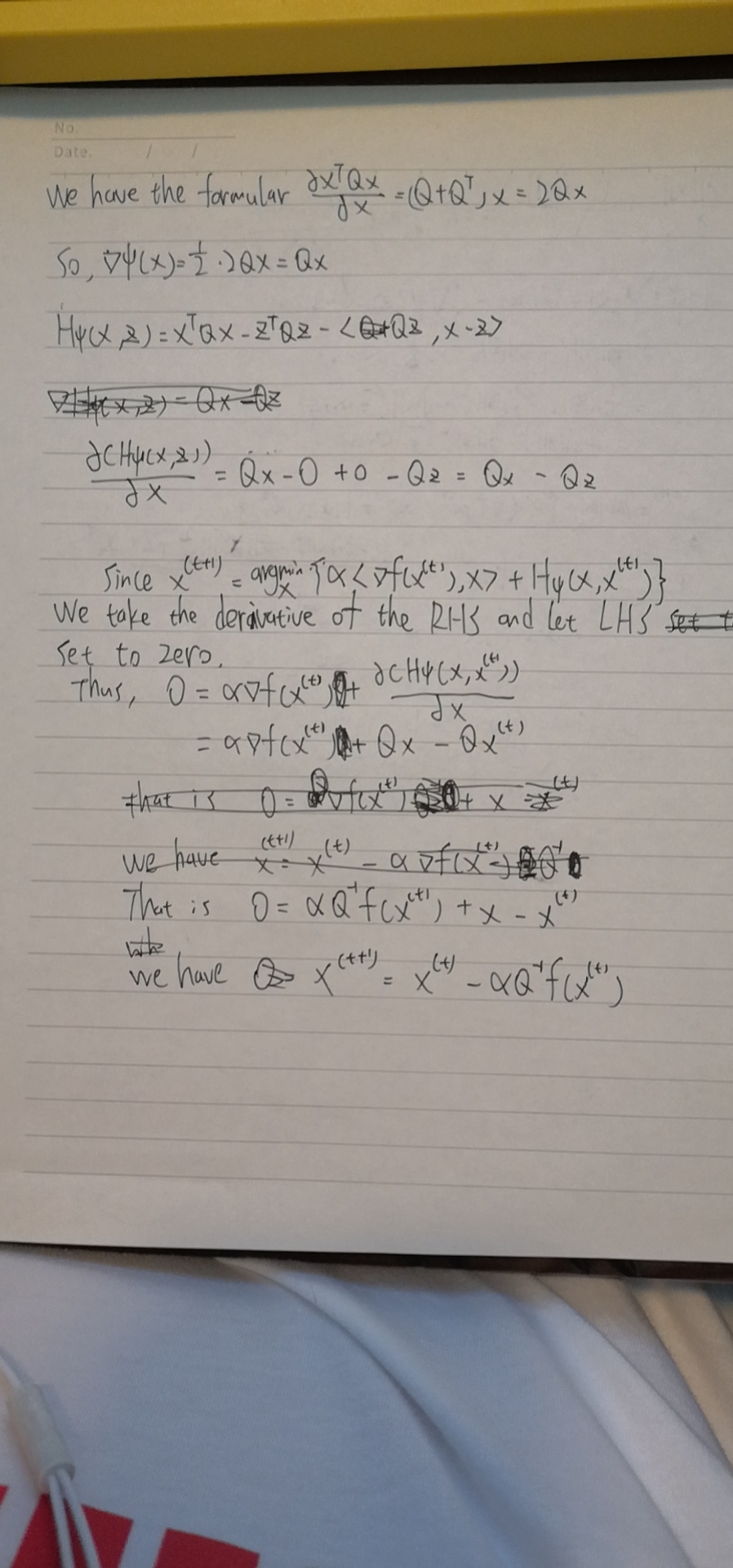
(c):

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| (i) |  |  |  |
| (ii) |  |  |  |
| (iii) |  |  | () |

**(i):**



**(ii):**



**(iii):**

