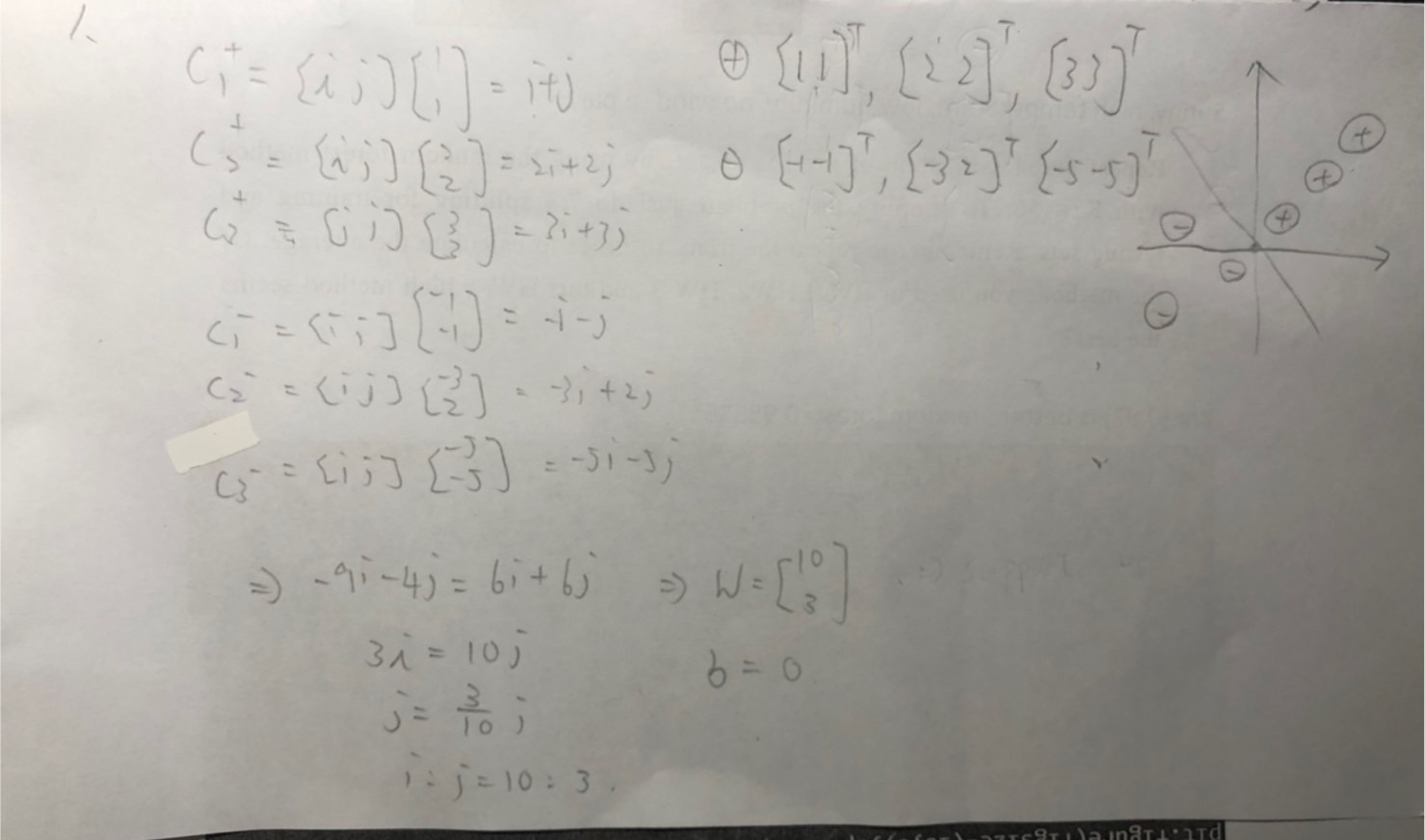
HW #6 Due: 5/24/2022

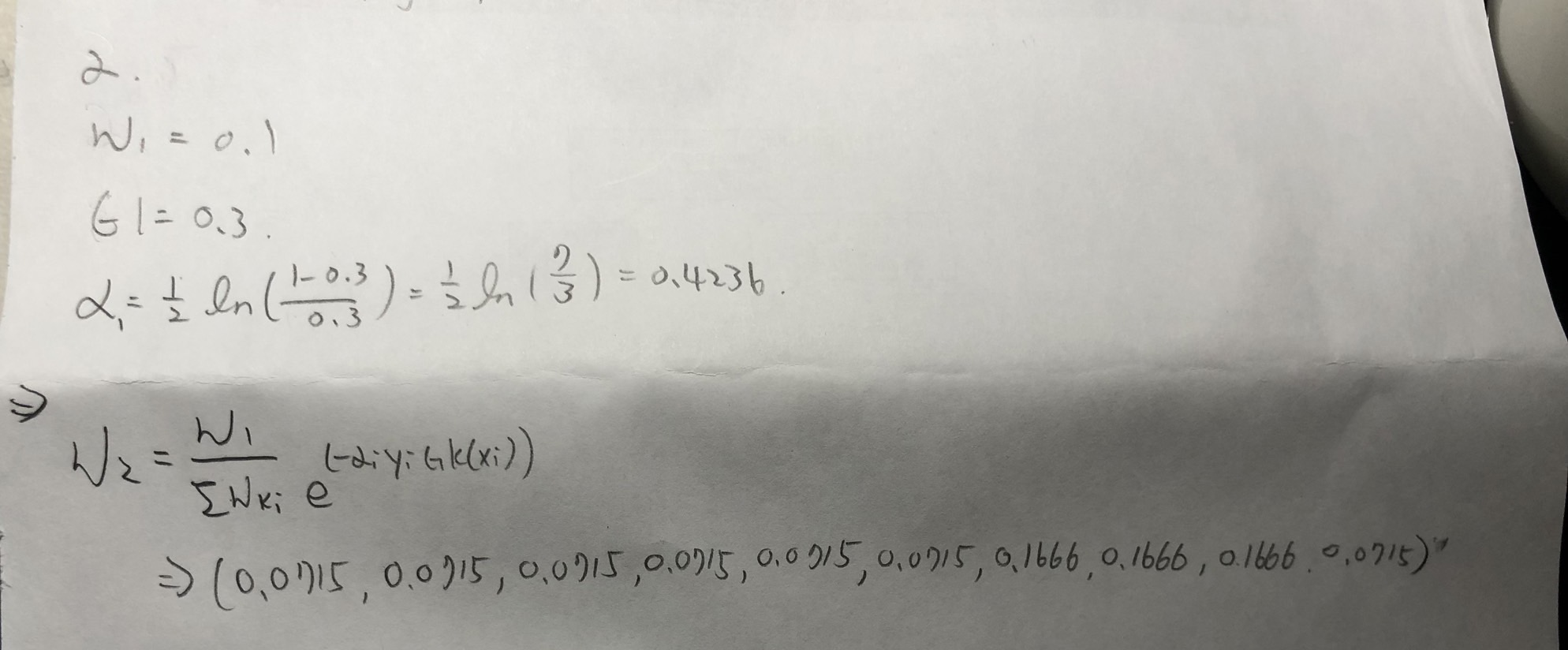
110598056 呂昀星

1. Please identify the support vectors and the corresponding 𝐰 and 𝑏 for the following training samples:  and

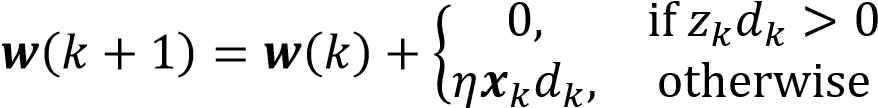
.



1. In the Adaboost lecture notes, we have an example of 5 2-dimensional samples. Continue the calculation for the second iteration.



1. We mentioned in the lecture (back propagation) that the perceptron with a loss function of 𝐽(𝒘) = |𝑧𝑘𝑑𝑘| − 𝑧𝑘𝑑𝑘 can be trained by using the following updating rule:



Show that this algorithm is directly derived from the algorithm.

A: Just randomly pick a data point to do gradient descent

1. Use the kernel SVM to classify the Iris data set. As usual, take 70% of the samples as training set and the rest 30% for testing. To simplify the problem, use the default values for *C* and 𝛾. Report the average accuracy after 10 trials.

Accuracy : 0.9777777777777776

1. from sklearn import datasets
2. from sklearn.model\_selection import train\_test\_split
3. from sklearn.metrics import accuracy\_score
4. iris = datasets.load\_iris()
5. #print(iris.feature\_names)
6. #print(iris.target\_names)
7. data\_x=iris.data
8. data\_y=iris.target
9. x\_train,x\_test,y\_train,y\_test=train\_test\_split(data\_x,data\_y,test\_size=0.3)
10. from sklearn.preprocessing import StandardScaler
11. sc = StandardScaler()
12. x\_train = sc.fit\_transform(x\_train)
13. x\_test = sc.transform(x\_test)
14. from sklearn import svm
15. svc = svm.SVC().fit(x\_train,y\_train)
17. #avg accuracy for 10 times
18. count=0
19. for i in range(10):
20. y\_pred = svc.predict(x\_test)
21. count+=accuracy\_score(y\_test, y\_pred)
23. count=count/10
24. print ("Accuracy : ", count)
26. Repeat problem 4 with the Adaboost classifier. When compare the accuracy obtained from problem 4 and this problem, which one is better?

Accuracy: 0.9111111111111112 , SVM is better

from sklearn import datasets

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

iris = datasets.load\_iris()

#print(iris.feature\_names)

#print(iris.target\_names)

data\_x=iris.data

data\_y=iris.target

x\_train,x\_test,y\_train,y\_test=train\_test\_split(data\_x,data\_y,test\_size=0.3)

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

x\_train = sc.fit\_transform(x\_train)

x\_test = sc.transform(x\_test)

from sklearn.ensemble import AdaBoostClassifier

ada = AdaBoostClassifier().fit(x\_train,y\_train)

#avg accuracy for 10 times

count=0

for i in range(10):

y\_pred = ada.predict(x\_test)

count+=accuracy\_score(y\_test, y\_pred)

count=count/10

print ("Accuracy : ", count)