# MLF homework1

b07902031 資工二 黃永雯

## Problem 1


## Problem 2

Semi-supervised learning gets the input containing a group of labeled data and a group of unlabeled data, which can be used for large data inputs that have the characteristic of cluster or similarity. One application of semi-supervised learning may be gene prediction since the input data size for such task is always very large. It can be used by giving a group of data with genome that have been annotated before and another group of data with genomes to find new gene. The learning algorithm can be initialize with the labeled data and can be improved by using it to scan the unlabeled data. If it

is improved better, it may be able to use the gene sequence to predict the function of the gene. It can be used by giving labeled data with the gene encoded before and find the similarity between the known genes and new-known genes to get the information of the function.

**Problem 3**

$E_f\{E_{OTS}(\text{A(D), f})\}$ = constant is true. It can be proved as follows: Since for any deterministic algorithm A, it can be calculated by: there is no $\text{f}(x_n) = \text{g}(x_n)$ + there is only one $\text{f}(x_n) = \text{g}(x_n)$ + ... + all the $\text{f}(x_n) = \text{g}(x_n)$. The formula can be written as follows: $\frac{1}{2^L * L} \sum_{k=1}^{L}(C_k^L * k) = \frac{1}{2^L} * \frac{1}{L} * \text{L} * 2^{L-1} = \frac{1}{2} =$ constant. Moreover, it can also be described as "no free lunch theorem" since no algorithm for data outside D will be better than the other.

**Problem 4**

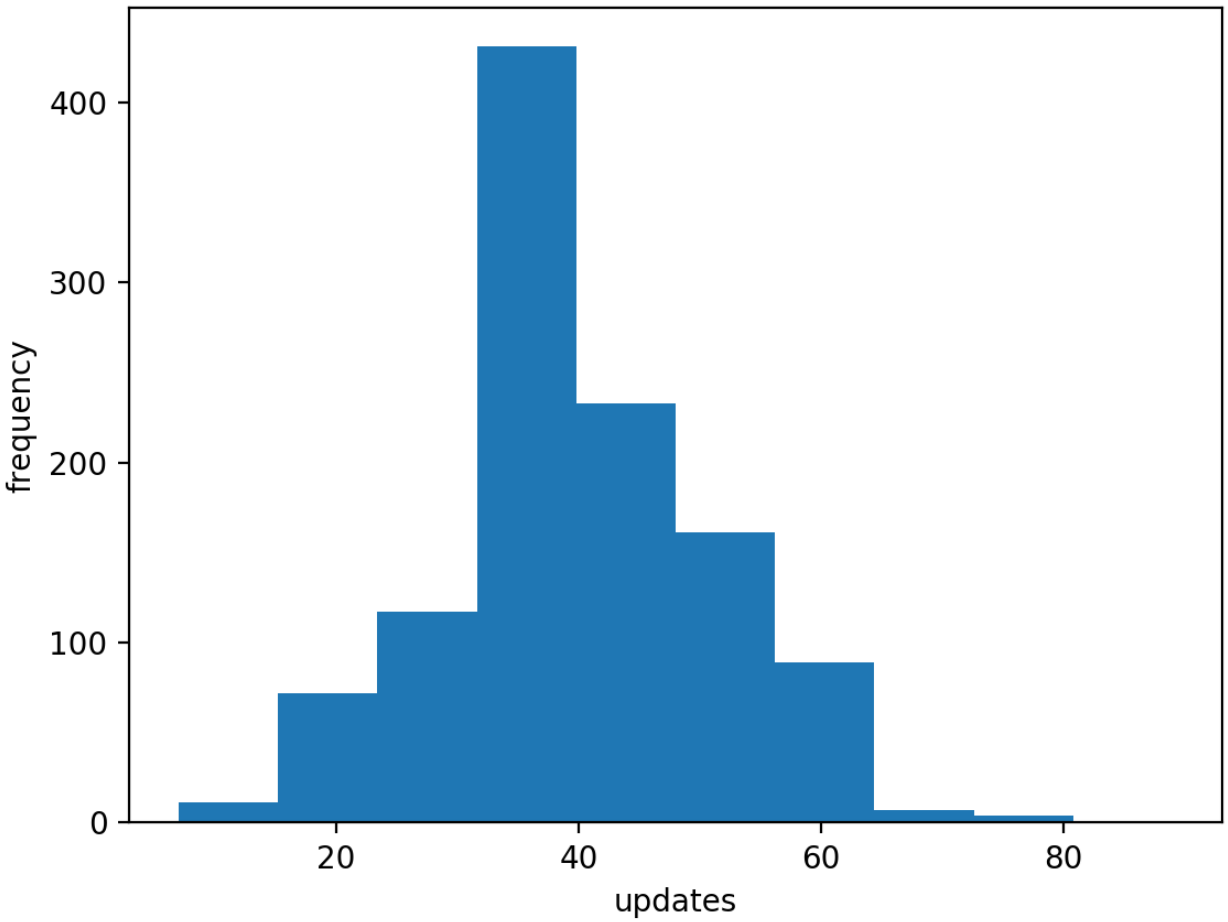
Number 1 is colored green in case A and case D. Since there are four kinds of dice (A, B, C, D) in the bag, the possibility of getting A or D every time is $\frac{1}{2}$. Moreover, since every time the possibility of getting green 1 is $\frac{1}{2}$, picking 5 dices means that the possibility becomes $(\frac{1}{2})^5 = \frac{1}{32}$

**Problem 5**

Number 1 is colored green in case A and case D, number 2 is colored green in case B ans case D, number 3 is colored in green in case A and case D, number 4 is colored in green in case B and case C, number 5 is colored in green in case A and case C, number 6 is colored green in case B and C. Therefore, from the previous question can know

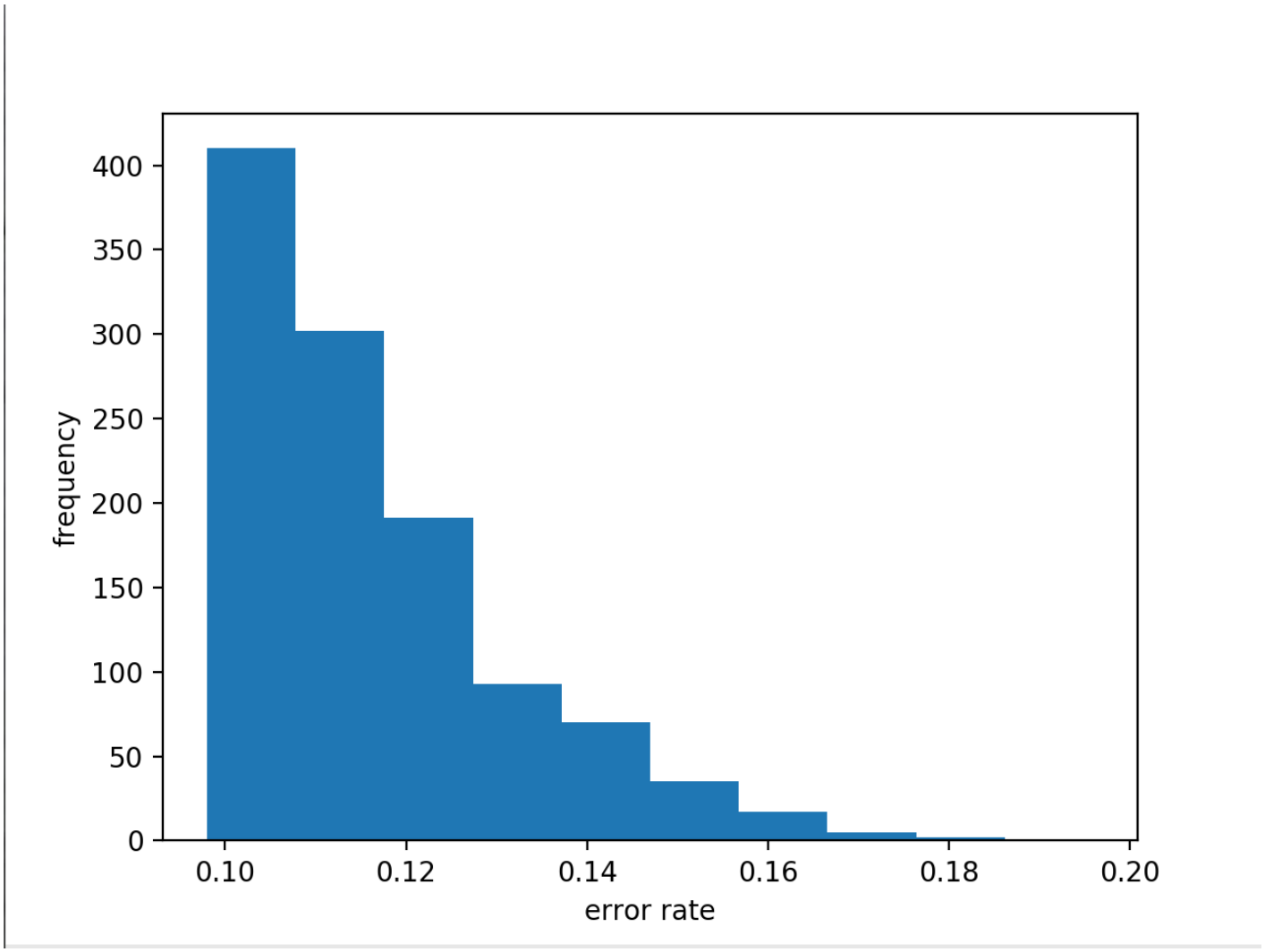
that getting 5 green for a specific number is $(\frac{1}{2})^5$. And now there are 4 possibilities (AD、BD、BC、AC), the possibility becomes 4 * $(\frac{1}{2})^5$. However, case A, case B, case C, case D are both calculated twice, the possibility becomes 4 * $(\frac{1}{2})^5$ - 4 * $(\frac{1}{4})^5 = \frac{31}{256}$. For the previous problem and this problem, the dice can be interpreted as data and the numbers can be interpreted as the hypothesis set. From Hoeffding's Inequality can know that with large data, the correctness of unknown data can be close to the correctness of known data. Therefore, in comparison with the previous problem can find that: this problem contain more hypothesis, which means that the probability is more higher than the previous problem.

**Problem 6**



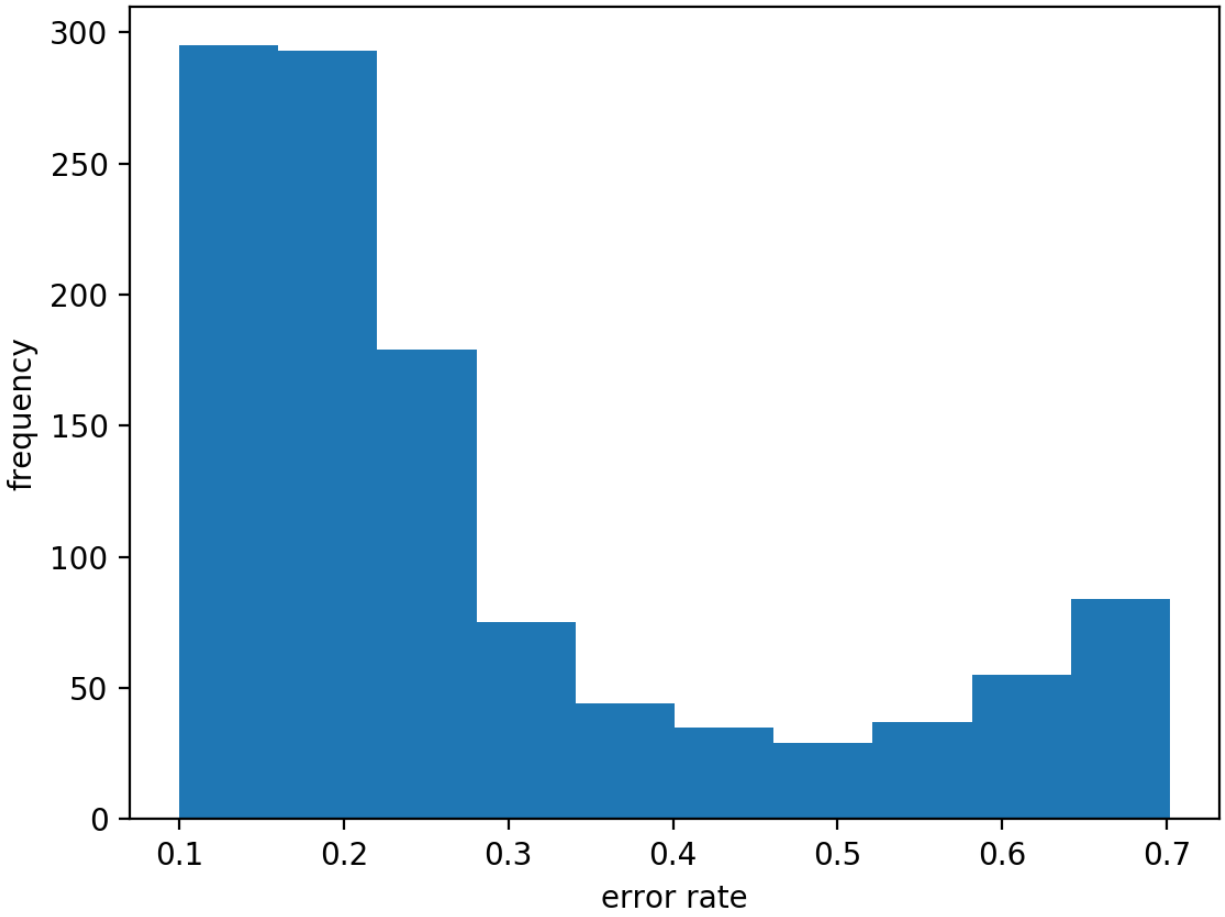

The average updates before the algorithm halts is 40.12344582593251. The above picture is the histogram of the number of updates versus the frequency of the number. (The description of the code of probelm 6, 7 and 8 is in README)

**Problem 7**




The average error rate on the test set is 0.11579218472468879. The above picture is the histogram of the error rate versus frequency.

## Problem 8




The average error rate on the test set is 0.28439431616341027. The above picture is the histogram of the error rate versus frequency. Compared with the previous problem can find that for data set that is not linear separable, the error rate of PLA algorithm is higher than pocket algorithm, which means that the performance of pocket algorithm is better.

## Problem 9

The plan will not work since the learning rate does not affect the run time for Perceptron Learning Algorithm. It only changes the scale of w, for example, in this case

(scaling down all $x_n$ linearly by a factor of 10) will only change the final w be $\frac{1}{10}$ of the final w with learning rate = 1. Using mathematical method can be described as follows: Since the upper-bound of T (the number of updates) $\leq \frac{R^2}{\rho^2}$, where $R^2 = \max_n ||x_n||^2$ and $\rho = \min_n y_n * \frac{W_f^T}{||W_f||} * x_n$. Thus, changing the learning rate will not influence the number of updates.