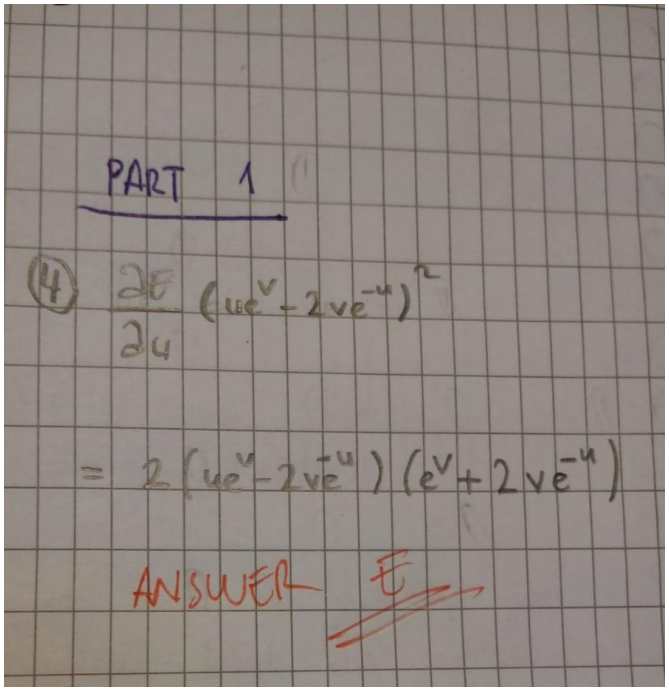
CSE 4088
INTRODUCTION TO MACHINE LEARNING
HOMEWORK #3
REPORT

PART 1:

4.



The following questions will be directly shown via screen shots. Because, the requested parameters have already been provided in the corresponding answers.

```
In [97]: runfile('C:/Users/Emre/Desktop/gradient_descent.py',
wdir='C:/Users/Emre/Desktop')
################################
QUESTION 5 & 6
Total Iterations:  10
u:  0.04473629039778207   v:  0.023958714099141746
################################
QUESTION 7
The error rate after 30 iterations:  0.13981379199615315
################################
QUESTION 8
Stop condition:  0.009458450672127984
Eout is:  0.0539683773999145 Total # of epoch:  8

In [98]:
```

IPython console | History log

of-lines: CRLF | Encoding: UTF-8 | Line: 138 | Column: 41 | Memory: 49 %

As it can be easily seen on the picture, I have printed out so;

5. D

6. E

7. A

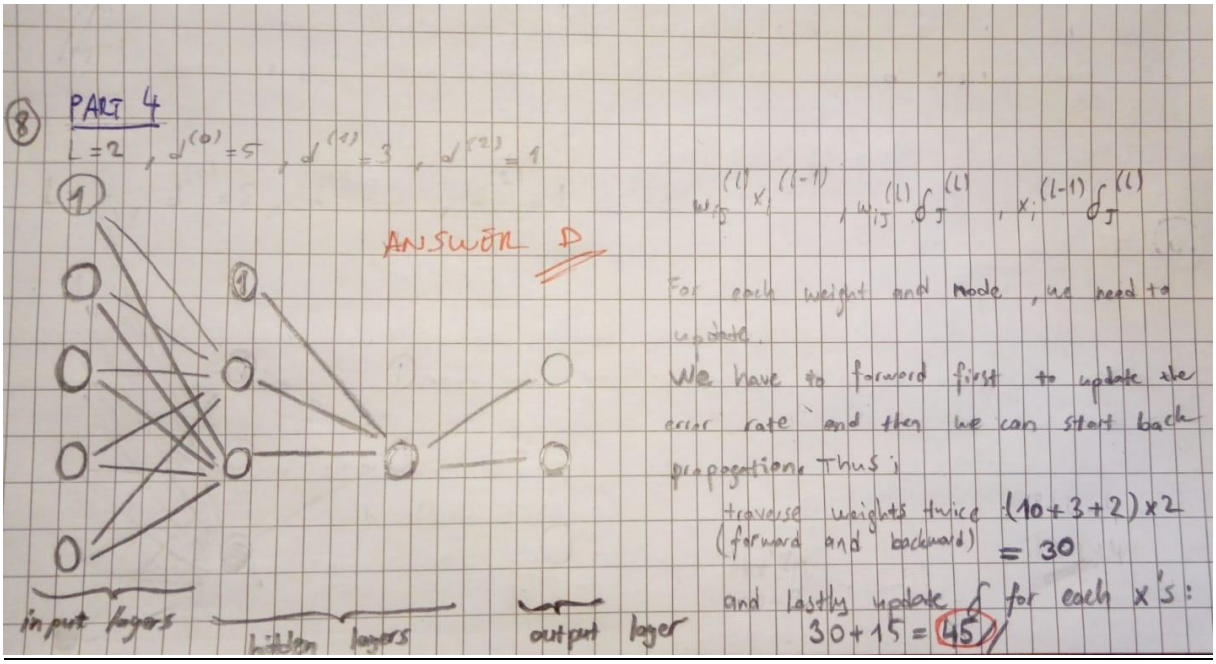
PART 2:

8. I got Eout is equal to approximately .0540

9. A

PART 4:

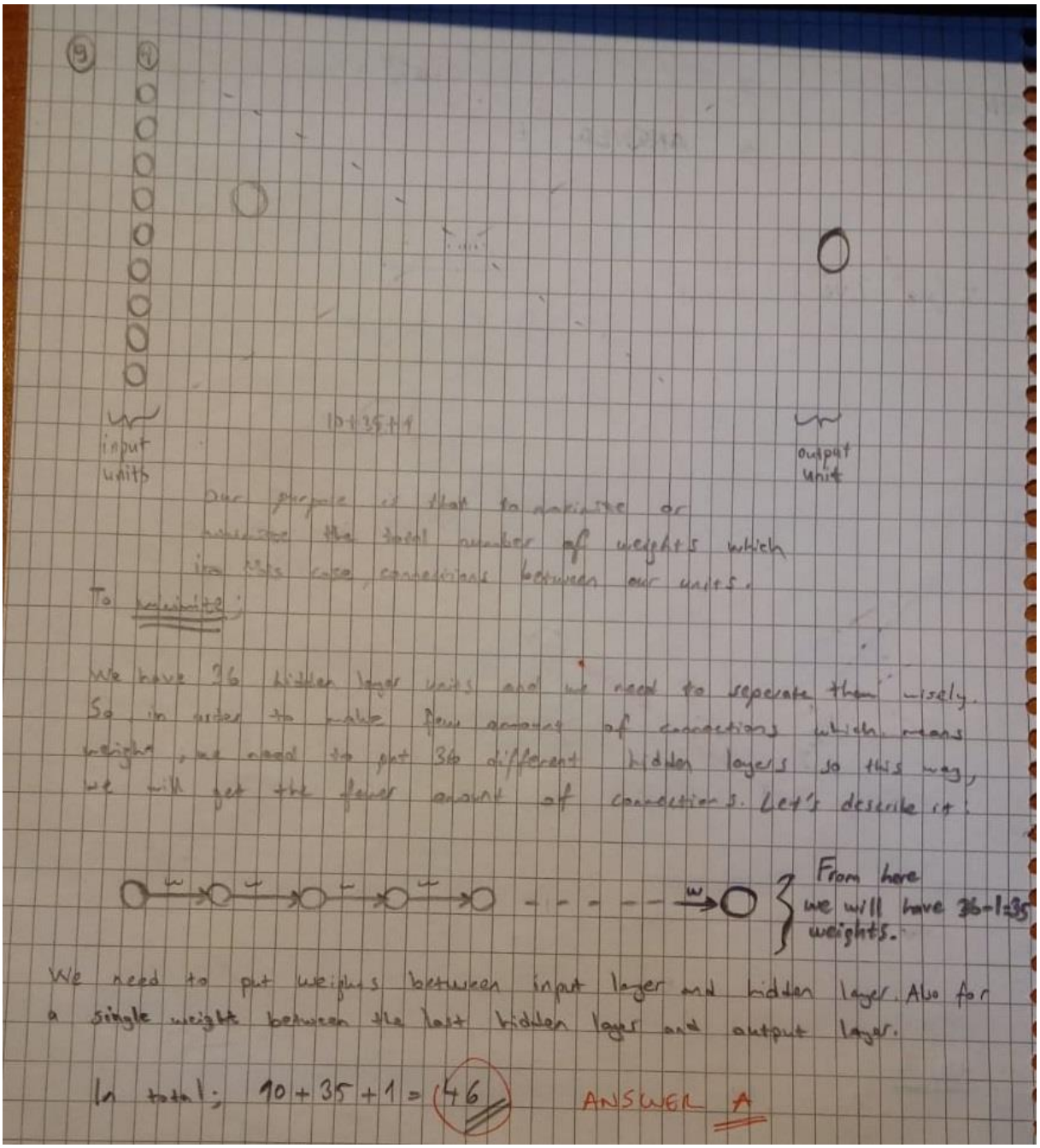
8.



For each weight and node, we need to update our weights. We have to forward first to take the input and then we can backward to update the error rate. Therefore,

Traverse weights twice: (10 + 3 + 2) x 2 = 30 (Forward & Backward),

and lastly update “delta” for each x’s: 30 + 15 = **45**



9. As I have written above, our purpose is that to maximize or minimize the total number of weights which in this case, connections between our units.

<u>To minimize</u>,

We have 36 hidden layer units and we need to seperate them wisely. So, in order to make fewer amount of connections which means weights, we need to put 36 different hidden layers so this way, we will get the fewer amount of connections.

From here we will have 36-1=35 weights in our hidden layers.

We need to put weights between input layer and hidden layer. Also for a single weight between the last hidden layer and output layer.

In total; 10 + 35 + 1 = **46**

10. To maximize,

On the contrary, if we have the minimum amount of layers then we can have maximum weight easily. So, we can have 2 hidden layers with 18 units each in total.

By this way, (10 x 18) + (18 x 18) + (18 x 1) = **522** total number of weights