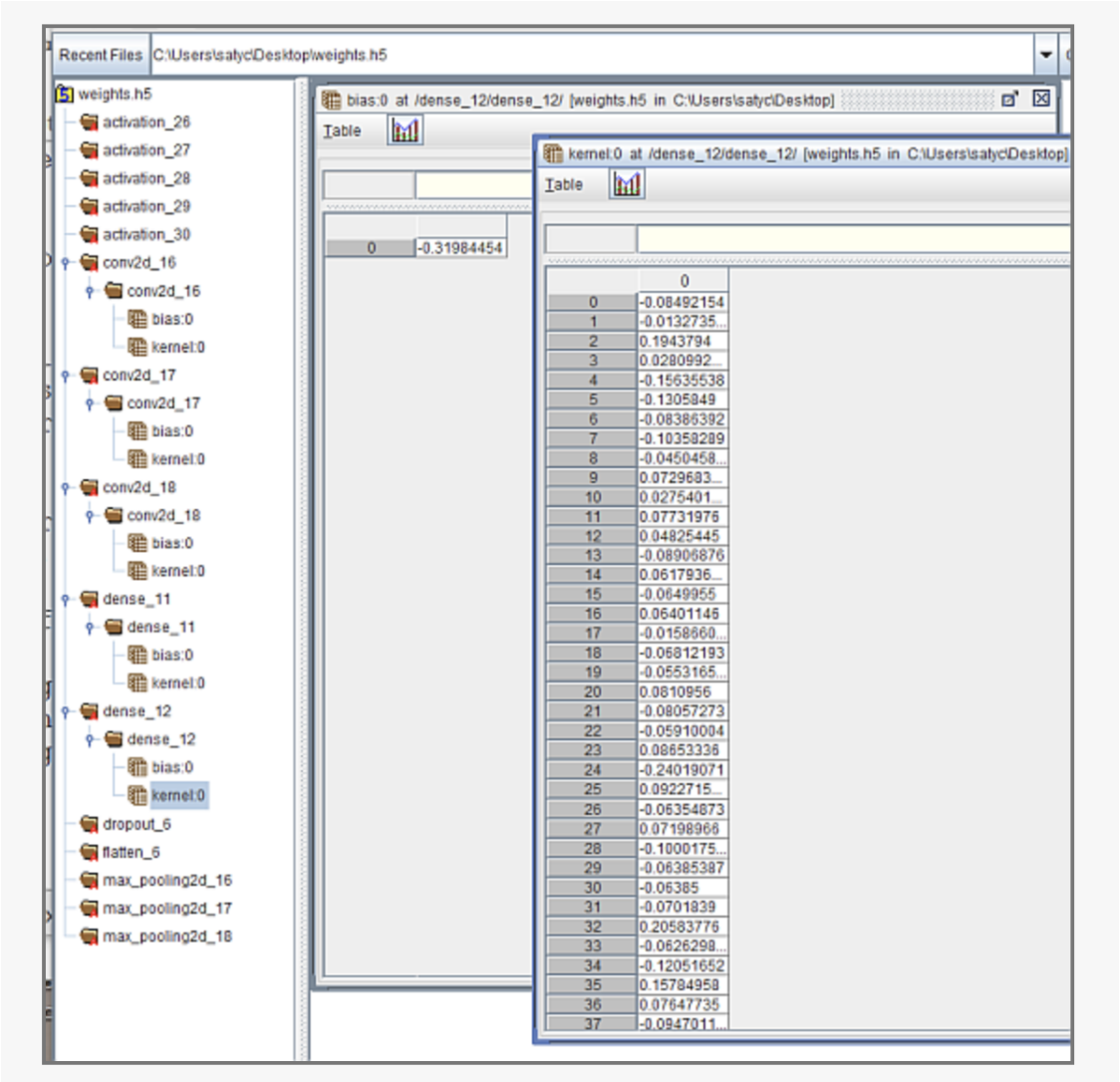
Machine Learning Rubrics

Q1 [0.5+0.5=1 point]. Submit your weights.h5 file. Also, create a submittable screengrab similar to the above [showing values for the second dense layer (eg. dense_12)]. For fun, click around, examine the arrays in the other layers as well. Again, it's all these values that are the end result of training, on account of iterating and minimizing classification errors through those epochs.



+ Point 0.5 => Submission of weights.h5 file.

+ Point 0.5 => Submission of screenshot above.

==========

Q2: [1 point]. Create a screenshot that shows the [correct] classification (you'll also be submitting your what{1,2}.jpg images with this).

What1.jpg (Cat Image) => +0.25 points
What2.jpg (Dog Image) => +0.25 points

Screenshot that shows correct classification => +0.5 point

Correct Classification

- For what1.jpg the output should be 0 (Cat Image => o/p is 0)
- For what2.jpg the output should be 1 (Dog Image => o/p is 1)

If what1.jpg is not submitted => -0.25 point
If what2.jpg is not submitted => -0.25 point

For Incorrect classification

- If only one of the classification is in correct => -0.25 points
- If both the classification are incorrect => -0.5 point

If no screenshot showing classification => -0.5 point

==========

Q3: [2 points]. Get a 'Corgi' image [the world's smartest dogs!], and a 'dog-like' cat image [hint, it's all about the ears!], upload to live/, attempt to (mis)classify, ie. create incorrect results (where the cat pic outputs a 1, and the dog's, 0), make a screenshot. Note that you need to edit the code to point myPic and myPic2 to these image filenames. If you can't get a Corgi to be misclassified, you can use pics from other dog breeds with 'pointy' ears, eg. Huskies :)

Corgi Image => + 0.5 points
Cat Image => + 0.5 points
Misclassification output screenshot => + 1 point

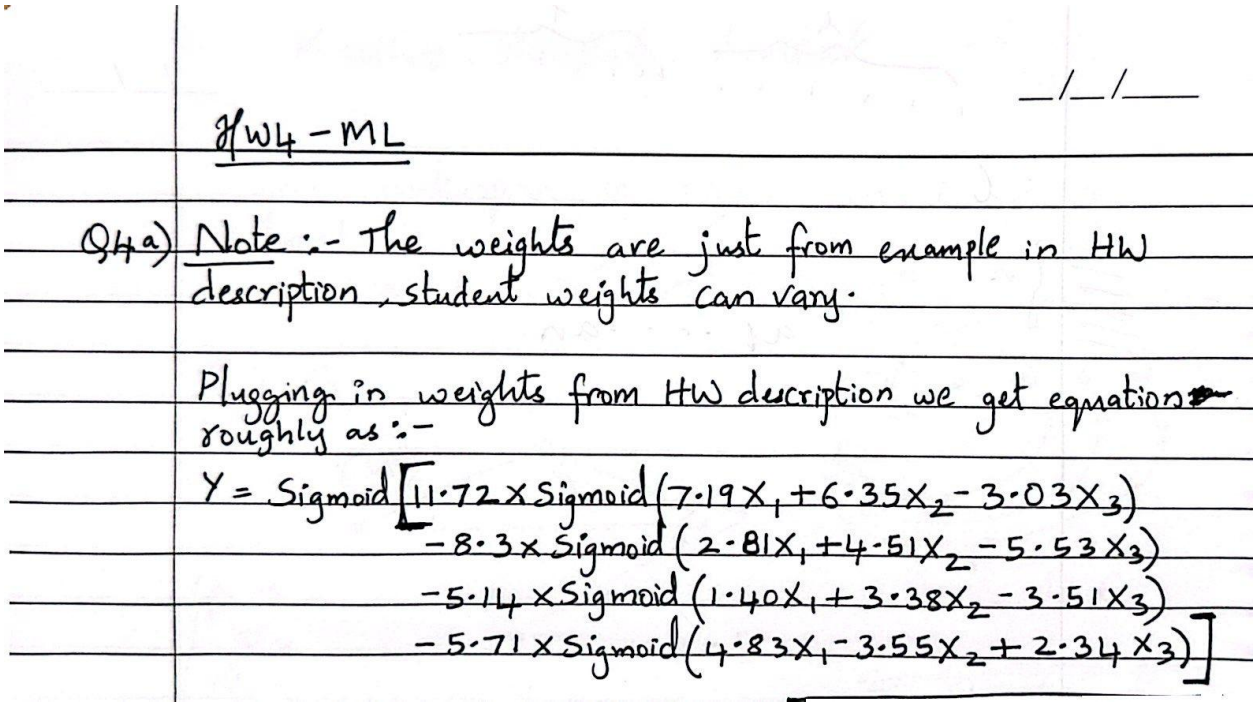
==========

Q4. (1+1 = 2 Point)

Solutions might be different but the method will be the same, compare the answers with the sample solution provided for grading.

Q4a. (1 Point)

Sample Solution:



HW4 - ML

__/__/____

Q4a) Note :- The weights are just from example in HW description, student weights can vary.

Plugging in weights from HW description we get equation roughly as :-

$$Y = \text{Sigmoid}\left[11.72 \times \text{Sigmoid}(7.19X_1 + 6.35X_2 - 3.03X_3) - 8.3 \times \text{Sigmoid}(2.81X_1 + 4.51X_2 - 5.53X_3) - 5.14 \times \text{Sigmoid}(1.40X_1 + 3.38X_2 - 3.51X_3) - 5.71 \times \text{Sigmoid}(4.83X_1 - 3.55X_2 + 2.34X_3)\right]$$

The equation weights are determined by the output of the python code written by the student, so the values like 11.72259331 from screenshot in description or 11.72 as written in sample solution above will be different for each student.

Handwritten/Digital/Word doc/Etc - Accepted

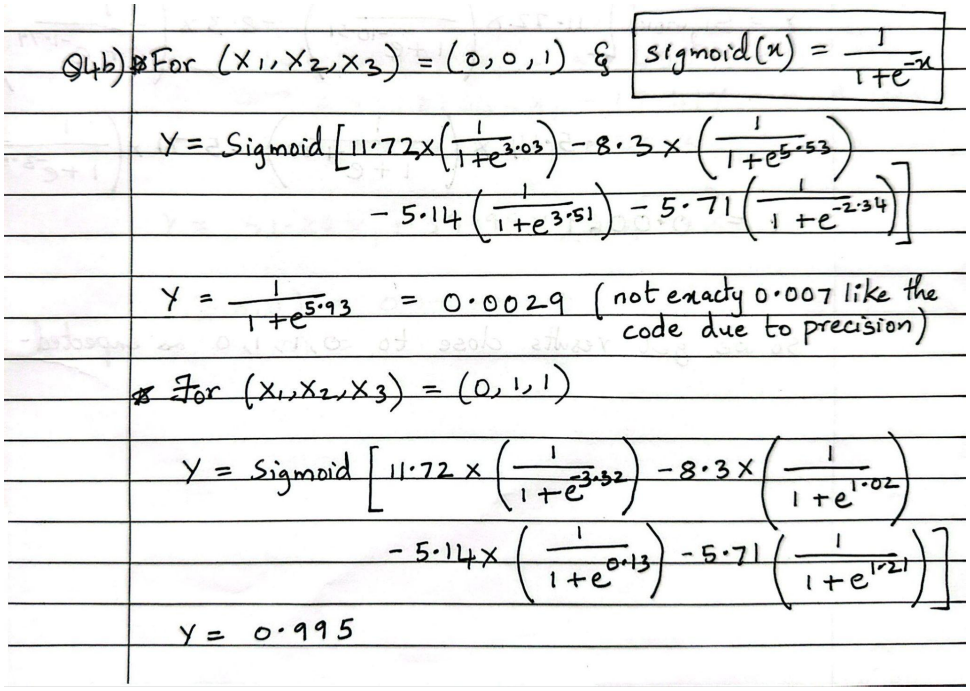
+1 Point => Correct equation - Also allow if students write Sigmoid instead of the entire (1 / 1 + e^(-x)), but ensure that they mention what sigmoid equals somewhere in 4a) or 4b).

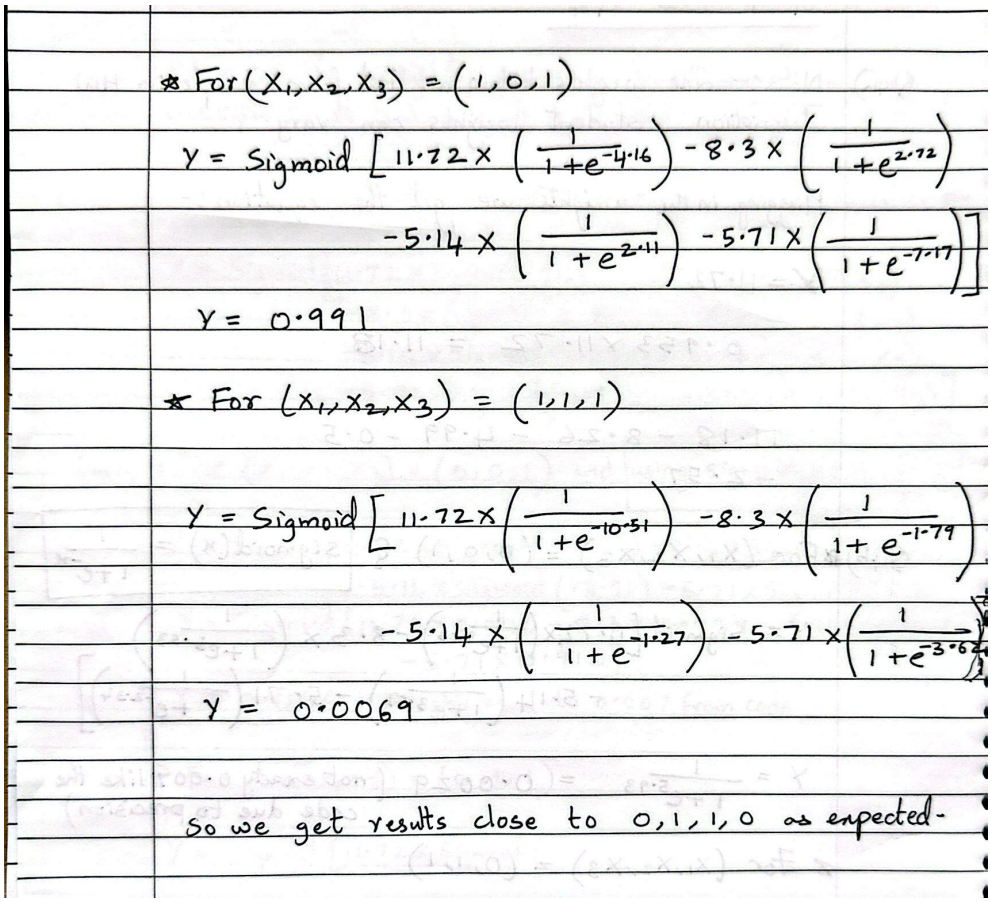
-0.5 Point => If a digital copy does not include any value of the equation properly

-0.5 Point => Wrong equation

Q4b. (1 Point)

Sample Solution:





+1 Point => Correct equation substitution for 4 values of X1,X2,X3 (001,011,101,111) and the four results of Y should be close to 0, 1, 1, 0 respectively. (They need not match exactly as in sample solution because students have different weights)

-0.5 Point => If a digital copy does not include any value of the equation properly
-0.5 Point => Wrong calculation / substitution i.e., answers not close to 0, 1, 1, 0 respectively.

****