ECS171 Fall 2017

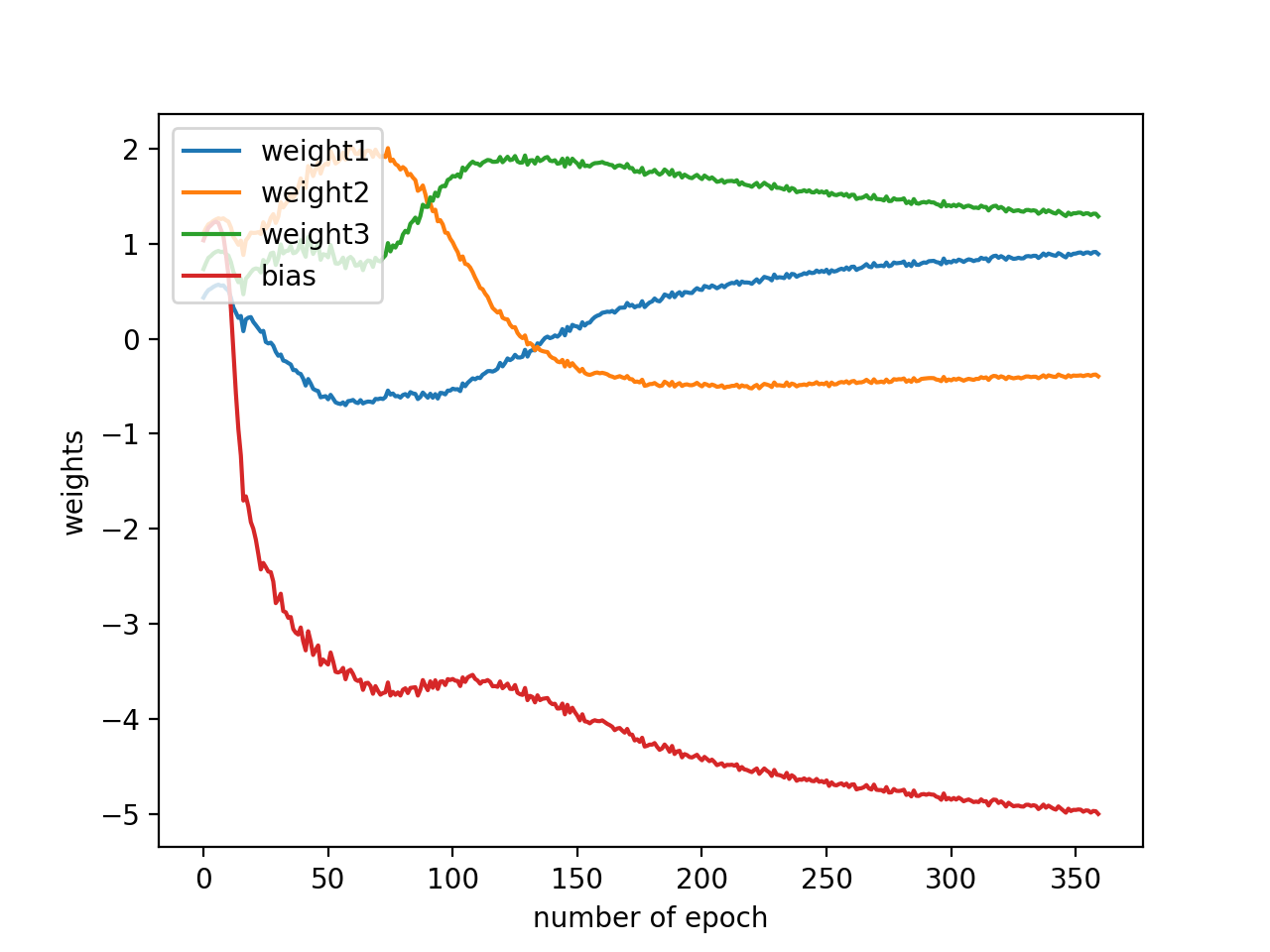
HW2 Report

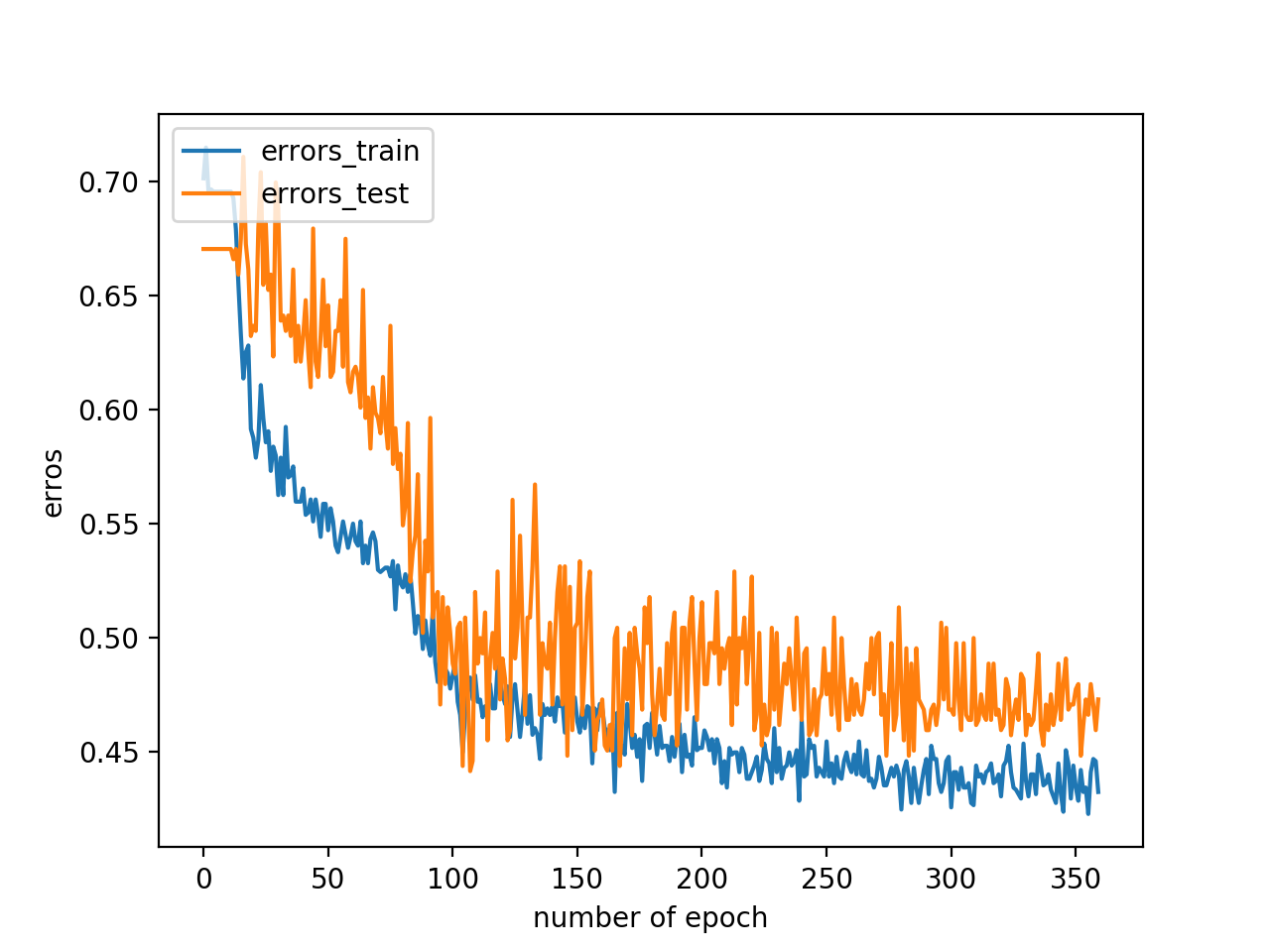
Gabriel Yin

999885129

Problem 1

Note: CYT is represented as 0. Epoch is 360 and the batch size is 4.





Problem 2

Training error: 0.43733 with epoch 360 and batch size 4.

**Note: For second part of problem 2 and problem 3, I used epoch 50 and batch size 106 to get weights.**

Output Layer:

a13 = 1.25001895 + 0.42561731\*a12 + 0.33864841\*a22 + 1.34060514\*a32

a23 = -1.04693103 + -0.41007811\*a12 + 0.45416743\*a22 + -1.00167429\*a32

a33 = -0.46573356 + -0.41449809\*a12 + 0.33651364\*a22 + 0.10922623\*a32

a43 = -0.89571196 + -0.83196318\*a12 + 0.48393652\*a22 + -1.06958175\*a32

a53 = 0.08087774 + 0.48059216\*a12 + 0.211287848\*a22 + -0.37825881\*a32

a63 = -1.75563765 + -0.61235946\*a12 + -0.20563982\*a22 + -2.1646502\*a32

a73 = -1.40927792 + -0.71622372\*a12 + 0.27104738\*a22 + -1.21637273\*a32

a83 = 1.00064397 + 0.80159432\*a12 + -0.29478848\*a22 + 0.77472192\*a32

a93 = -0.58902067 + -0.84687299\*a12 + -0.41719934\*a22 + -0.818939937\*a32

a103 = -1.44403565 + -0.46237993\*a12 + 0.40011171\*a22 + -0.95560074\*a32

Hidden Layer:

a12 = 0.283537 + -0.19136925x1 + 0.28130335x2 + 0.73879099x3 + 0.57831055x4 +

-0.29467672x5 + 0.13249843x6 + 0.86561346x7 + -0.04757586x8

a22 = -0.39708963 + 0.08865377x1 + 0.07860584x2 + -0.94174445x3 + -0.45270586x4 +

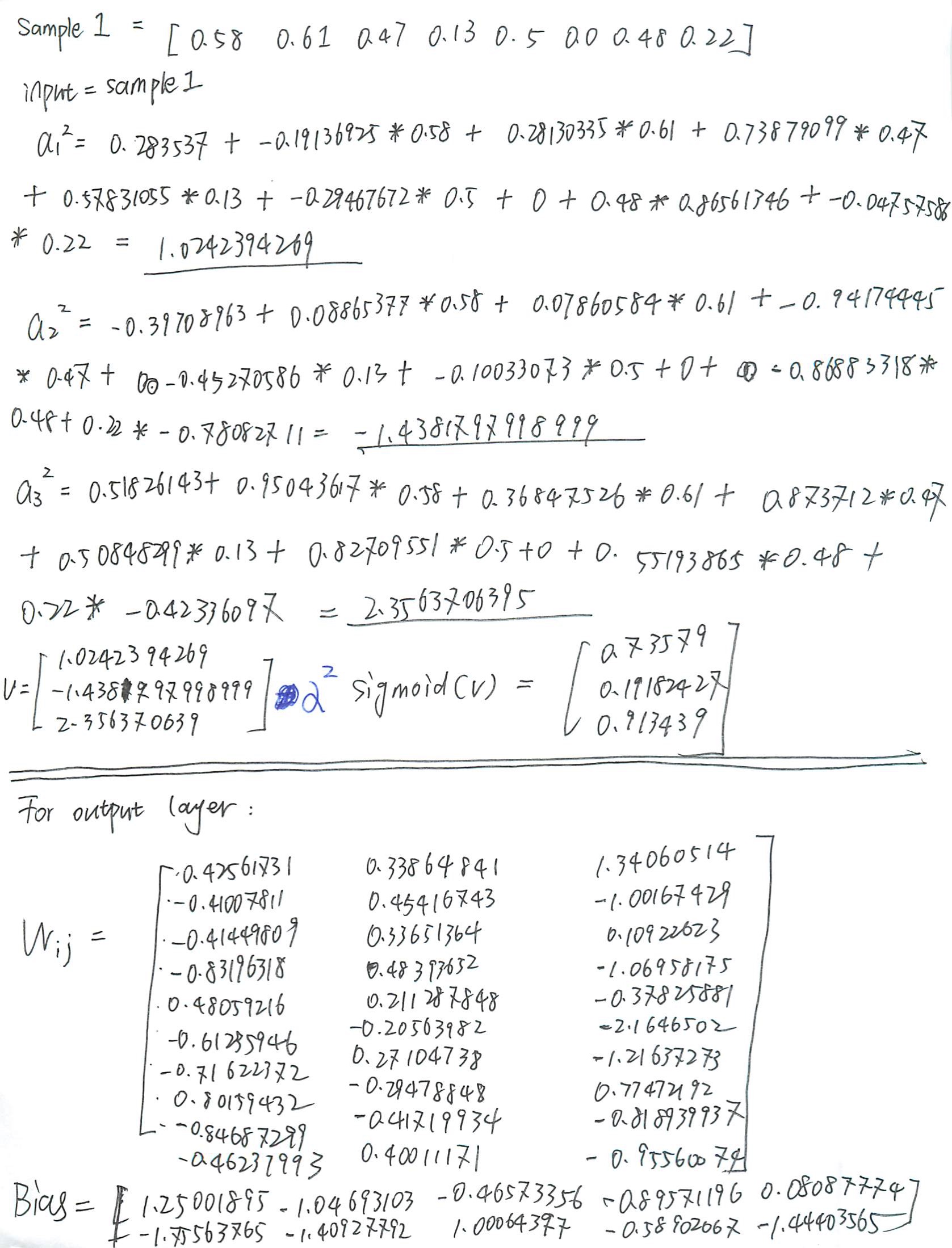
-0.10033073x5 + 0.09803814x6 + -0.86883318x7 + -0.78082711x8

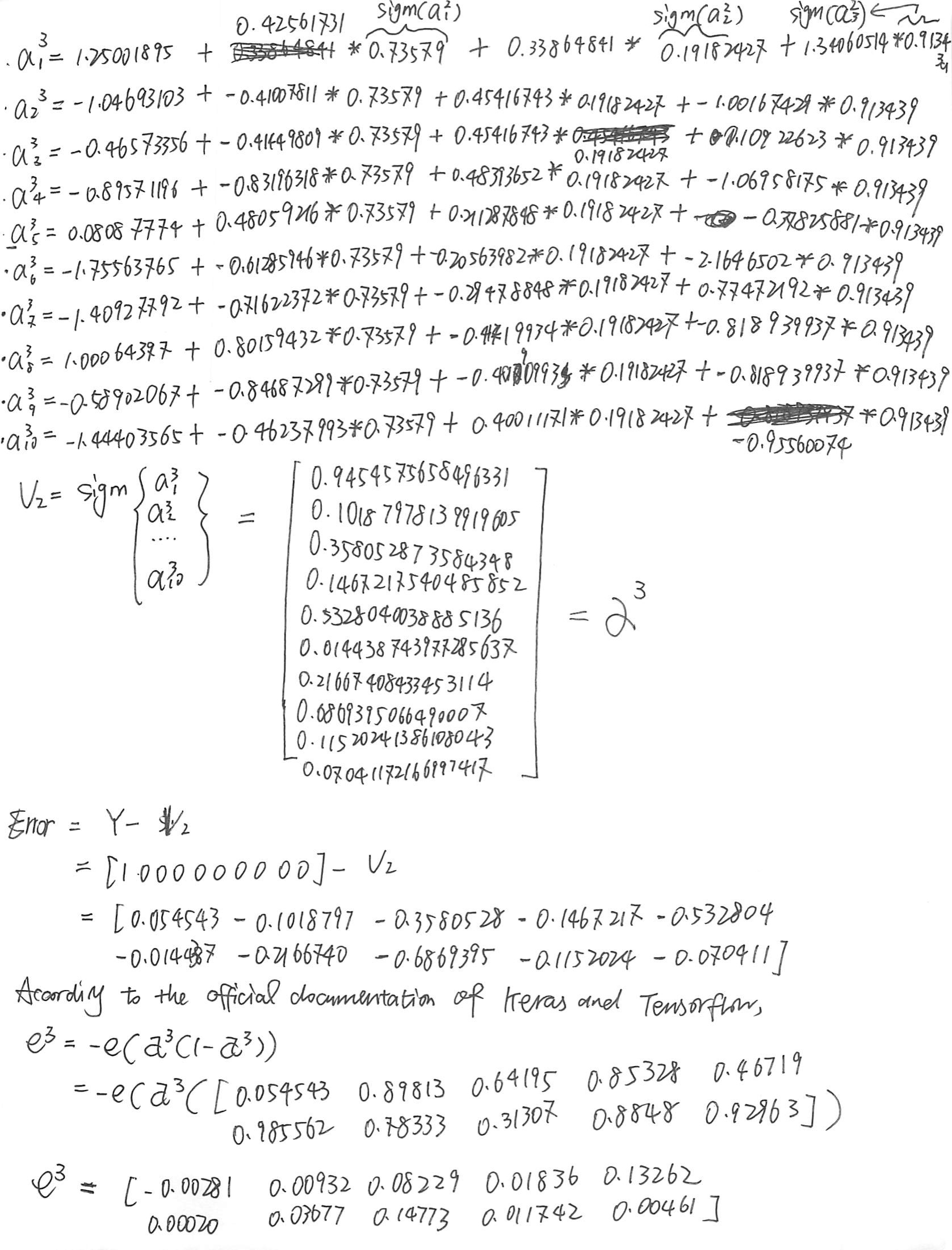
a32 = 0.51826143 + 0.95043617x1 + 0.36847526x2 + 0.873712x3 + 0.50848299x4 +

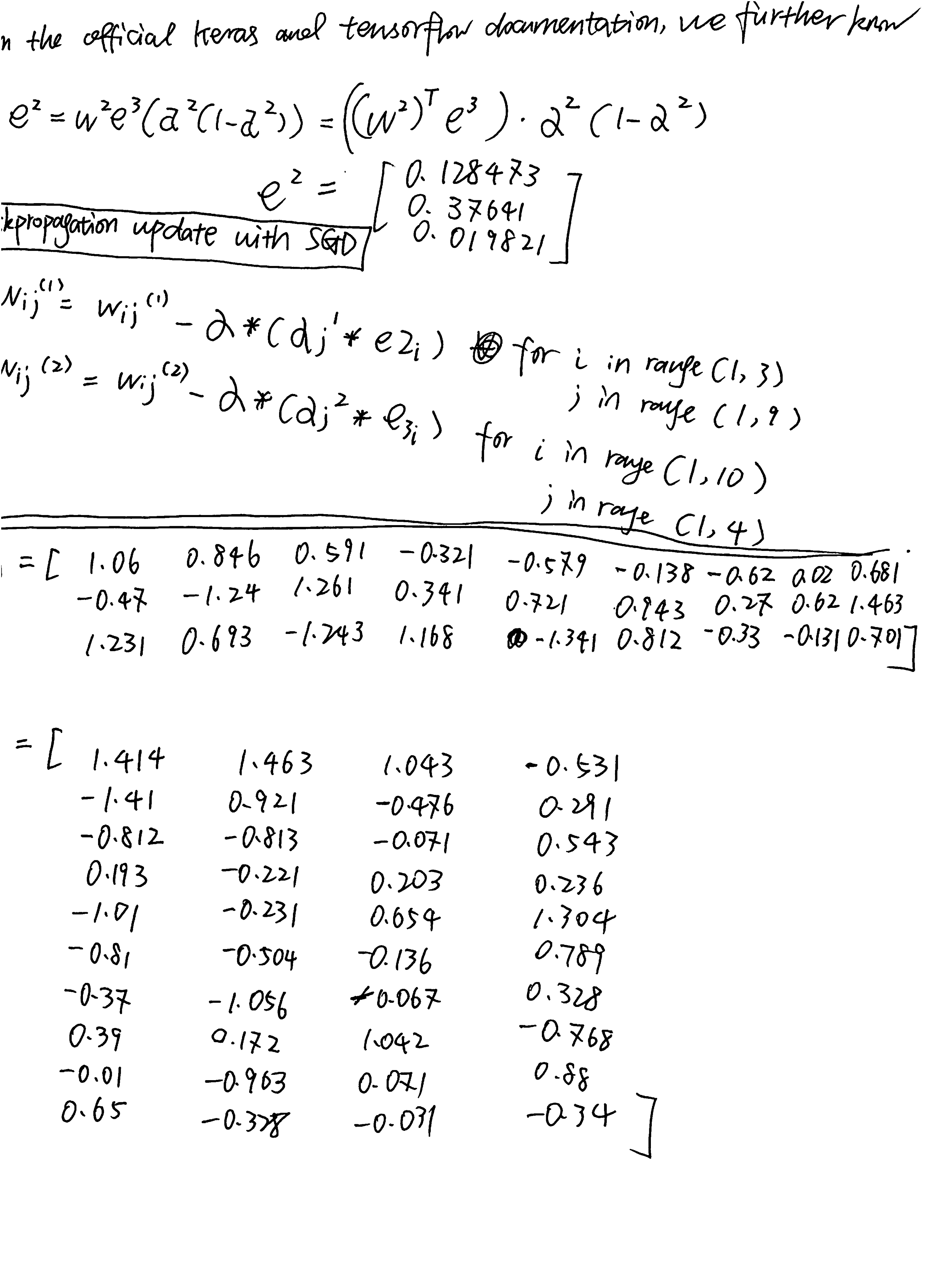
0.82709551x5 + 0.92576641x6 + 0.55193865x7 + -0.42336097x8

Problem 3

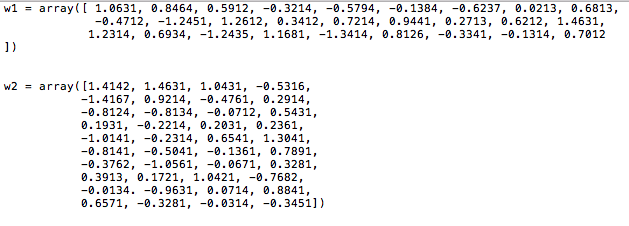
**Note: On the Last page, the first [] is W1 and second is W2. I also attached an annotated version of it in the folder. If you are confused with which one is W1 and which one is W2, please check out Problem2.pdf.**







You can see the machine result below and they are in agreement with each other. If the image is not clear for you, please check p3 result.txt.



Problem 4

Epoch: 360 Batch: 4 Learning Rate: 0.1 Optimizer: SGD

|  |  |  |  |
| --- | --- | --- | --- |
| Node \ Layer | 1 | 2 | 3 |
| 3 | 0.43733 | 0.50673 | 0.68801 |
| 6 | 0.39892 | 0.42341 | 0.68806 |
| 9 | 0.39229 | 0.43463 | 0.54582 |
| 12 | 0.36781 | 0.44811 | 0.55997 |

From the table, we can see that the optimal configuration is 1 Hidden layer with 12 Nodes per Hidden layer. The relationship can be generalized as the fact that increasing nodes per layer will reduce the error while increasing number of layers will escalate overfitting issue and thus lead to an increase in error.

Problem 5

It is classified as NUC using the optimal configuration with 1 hidden layer and 12 hidden nodes. I called model.predict\_classes(x) method and it returns 7, which is NUC.

Problem 6

If the error comes from Gaussian white noise, an easy way to do this is to use standard deviation to see if the data are far enough from each other. However, if two competing possibilities are equally high, this method does not work very well then. Another method might be sort the output and see if the highest one is high enough to distinguishes itself from the rest. As a result, the confidence level is the sum of the difference between the highest possibility and each of the rest of the estimation.